INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Securing Hypervisors in Cloud Computing Environments against Malware Injection

Mohammed Al Qurashi[1]*

1 Computer science department, Al - Baha university, Al Baha, Saudi Arabia

OPEN ACCESS

Received: 29-07-2023

Accepted: 19-09-2023

Published: 25-10-2023

Citation: Qurashi MA (2023) Securing Hypervisors in Cloud Computing Environments against Malware Injection. Indian Journal of Science and Technology 16(39): 3386-3393. https://doi.org/10.17485/IJST/v16i39.1913

* **Corresponding author**.

malqurashi@bu.edu.sa

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: The primary objectives of this research are to address the security concerns related to cloud computing, emphasising attacks that target different hypervisor layers. The goal is to propose a revolutionary approach called ”hGuard,” that provides a thorough protection mechanism against malware attacks across several hypervisor levels. Additionally, the research aims to establish this method's ability to improve cybersecurity in cloud systems and show its efficacy through practical studies. **Methods:** The study combines theoretical analysis with actual experimentation to accomplish its objectives. The ”hGuard” approach that is being proposed was developed to defend against attacks on several hypervisor levels. The strategy produces an output that the data mining algorithm, such as Apriori uses to predict potential attacks. Through this association, it is now possible to simultaneously anticipate and stop malware injection attempts at various hypervisor layers. Empirical tests that simulate attacks and examine real-world situations provide quantitative information on the method's performance. **Findings:** The ”hGuard” approach achieves a 95% detection accuracy for identifying malware injection attacks, with a 3% false positive rate for minimal misclassifications of non-attacks. It also demonstrates an 5% false negative rate, reducing errors in categorizing actual attacks. Additionally, the approach boasts an efficient 20 ms execution time, ensuring rapid processing and prediction of potential attacks. **Novelty:** The novelty of this research lies in the development of the ”hGuard” method, which addresses a crucial gap in existing security approaches. Unlike conventional methods that tackle hypervisor levels individually, the proposed approach offers a holistic defense mechanism capable of countering malware attacks targeting multiple levels simultaneously. The integration of the Apriori technique for attack prediction further enhances its novelty by providing a data-driven approach to proactive cybersecurity. The empirical validation of the method's effectiveness contributes to its novelty, showcasing its potential as a valuable tool for detecting and preventing malware attacks in cloud computing. Furthermore, the research suggests avenues for extending the application of the ”hGuard” method to other domains within the realm of cybersecurity.

Keywords: Cloud Computing; Cloud Security; Hypervisors; Virtual Machines; Malware Attacks

1 Introduction

Cloud computing has emerged as a critical technology in recent decades, helping organizations decrease the cost and complexity of their applications[1]. Virtualization is an essential aspect of cloud computing at the same time, is most challenging to implement[2]. Virtualization is the abstraction of computer resources to optimize the use of available computing resources. The hypervisor is the most crucial component since it manages the physical platform and accesses its resources when it comes to virtualization. Briefly stated it is a piece of software that allows many virtual machines to operate on the same server. The most challenging issue for a hypervisor is security; if the hypervisor is hacked, the system becomes vulnerable. Hackers target hypervisors because they are intended to manage all of the resources available on a computer's hardware while also controlling all of the virtual machines running on it[3,4]. One of the techniques for breaching hypervisors is via the introduction of malware into the system. To get access to a user's databases or resources, a hacker would inject malicious code or service into the applications that seem to be legitimate service instances operating in the cloud, according to the user's preferences[5].

Consequently, specific service calls to the user's service are handled inside that rogue instance, resulting in compromising personal data[6,7]. Five major methods of attack and compromise have been identified, including attacking virtual machines, attacking through unmonitored traffic, attacking the storage device, attacking hosts with no security, and weakness in security tools which are late in attack detection when the attack has already been performed[8]. The major challenge with existing resarches is that they suffer with low detection accuracy, high false positive and huge computational costs[2,3,6]. In light of these concerns, the purpose of this study is to highlight the challenges that have evolved in defending and safeguarding the hypervisor and offer some potential solutions.

In a virtualized environment, a variety of connected guest machines have their own security zones that are inaccessible to other virtual machines with their own security zone. Hypervisors have their security zone since they are the primary controller of what occurs within the host machine's virtualized environment. A hypervisor may have an effect on the way a VM host operates. A VM may include several zones, all of which are contained inside the same physical infrastructure. This may cause a security issue if the hypervisor is exploited, and the attacker takes control. When such an attack succeeds, the attacker gets complete control of all data contained inside the hypervisor environment[9].

Certain current models which are capable of protecting the hypervisor in a range of scenarios as shown in Figure 1. The CloudIDEA's is used to identify malware injection attacks by monitoring and tracking Virtual Machines[10]. Virtual machines must be secured in case an attacker gains control and may affect the hypervisor[11]. CloudIDEA's primary goal is to identify threats by monitoring and tracking virtual machines[12]. It identifies malware in the cloud by detecting the signs of malicious activity in code. This approach is limited to defending against malware injection attacks directed at the hypervisor. The major issue with this approach is how would it work if the advanced control protection system is placed on the host platform to monitor for suspicious activity. Another issue is what will happen when the attacker directly attacks the hardware. Based on these issues, obviously, this approach is ineffective in completely defending against the attacks.

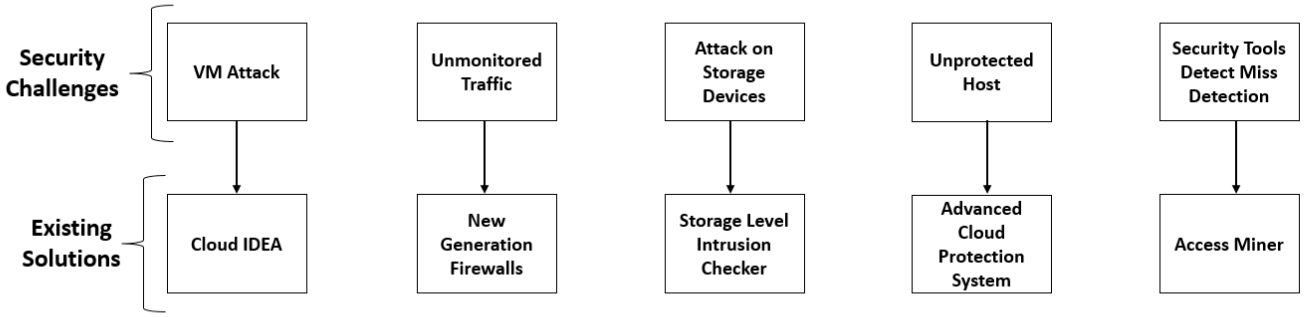


Fig 1. Securitychallenges and existing solutions

Another option is to use New Generation Fire Walls (NGFW) as a countermeasure to prevent malware injection attacks against hypervisors[13]. The firewalls determine the flow of traffic to enter or leave based on a set of rules, and from there, the firewall determines which traffic to prevent from entering. However, the emergence of Web 2.0 has highlighted the issue of port-based firewalling[14]. A next-generation firewall has been introduced, gives more control over how traffic enters and exits in a network, these kinds of firewalls are based on typical firewall features. This method can only address one problem that is protection from unmonitored traffic. The downside of this approach is that if the attack originates from other sources, such as a lack of host security, this kind of tool will be useless in that situation.

Storage-Level Intrusion Checker (SLICK) approach monitors write access and permissions on storage devices to identify intrusion attempts and operate within the hypervisor without impacting the guest virtual machine[15]. It includes a feature that enables visitors to code or craft their own rule using the system's language to suit their needs. The system does not need any modifications or self-examination on the guest virtual machines. It operates invisibly without any input from or output to the guest, preventing malware attacks from destroying their storage devices. SLICK is a helpful tool for detecting intrusion attempts against storage devices. It operates invisibly on the guest virtual computers, causing no disruptions. On the other hand, SLICK solely protects against attacks on storage devices and does not address or prevent malware injection attacks on other parts of the machine[16].

Another existing method to avoid malware injection attacks against hypervisors is the advanced cloud protection system (ACPS)[17]. In ACPS the recorder module warns other multiple recorder module about the security risks and saves all the threats and risks in a warning pool. The checksum of the examined item is computed, and the infrastructure status will be determined asynchronously. If there are any abnormalities, the evaluator will get a warning. Since the hypervisor is linked to the network infrastructure, it must protect the network infrastructure before attacking the hypervisor. As a result, ACPS provided a technique for detecting network probing using IP tables to protect networks against attacks. The ACPS improves cloud resource security and virtual machine cloud access points are constantly monitored. It also has a mechanism for detecting network probing, ensuring that the network remains safe while the hypervisor is connected to it. ACPS does not directly defend hypervisors against malware injection attacks; rather, it prevents them from occurring in the cloud or network by adding a layer of protection before the hypervisors themselves.

Another method for avoiding malware injection attacks against hypervisor is AccessMiner[17]. AccessMiner has the benefit of anticipating any potential assault, even if the malware is unknown or has yet to be detected. Because it can adequately distinguish between benign and malicious behavior via analysis of application activity, AccessMiner utilized a system-centric model to identify malware. This approach, however, may result in a time restriction while executing the task under the hypervisor. For every system call invocation, AccessMiner requires the hypervisor to provide a trustworthy path. If there are many system calls to transit via the hypervisor, this may take a long time, causing a temporal restriction in the current program. The downside of this approach is time complexity in creating a trusted path.

The current study focuses on a crucial gap in the realm of cloud security. As highlighted in the objectives, our research introduces the "hGuard" approach, which tackles security concerns in cloud computing, particularly targeting attacks on diverse hypervisor layers. This revolutionary approach offers a comprehensive defense mechanism against malware attacks that affect multiple levels simultaneously. Unlike conventional methods, which address hypervisor levels individually, "hGuard" provides a holistic solution.

2 Methodology

The security challenges in hypervisors, such as virtual machine attacks, unmonitored traffic, storage device attacks, no security at the host, and late attack detection, may be resolved using the solutions. As shown in Figure 1 all of the existing solutions, however, are unable to solve the same fundamental issues. Each solution can only address one issue. The question arises what happens if several attacks appear at the same time.

We propose a system called hypervisor guard “hGUARD” as a solution to all of the security challenges and concerns encountered. This method serves as a protection method to protect the hypervisor from being compromised. “hGUARD” can defend the virtual machine by identifying malware injection attacks and preventing them from monitoring and tracking virtual machines. It also monitors write access permissions to storage devices, and it runs within the hypervisor without interfering with the virtual machines of the guests. As a result, all data will be screened, and any potential risks will be identified and deleted immediately. This protection mechanism will assist in determining the flow of traffic to and from the facility and anticipating potential attacks by unknown or undiscovered threats while at the same time server’s traffic should be closely watched continuously. “hGUARD” has a firewall feature that functions similarly to a doorman, in that it will be stationed at the entry of the network and will manage network traffic flow. “hGUARD” offers the ability to verify that appropriate programs utilized on the network, which may include previously undiscovered apps, bandwidth-intensive applications, and peer-to-peer applications, among others. Aside from that, it can detect and record suspicious actions and inform users by sending them a warning. As a result, users will have more time to examine and filter suspicious activities to determine if they are a danger or not before they manifest themselves deeper into the system. In general, based on this approach, every hypervisor layer has its defense mechanism against attacks, including malware injection into the system. The architecture of the proposed approached is shown in Figure 2.

Each layer of “hGUARD” has its method for preventing malware penetration. The system will protect each layer to avoid an attack on the hypervisor. Algorithm 1(Table 1) explains the working of “hGUARD”.

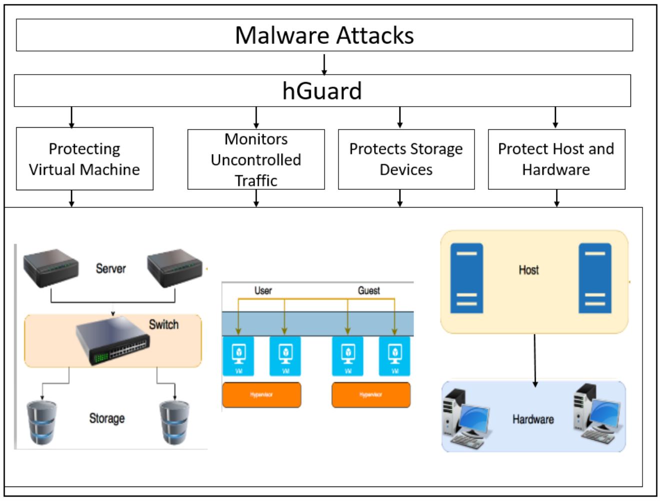


Fig 2. Proposed Approach

2.1 Attack on Virtual Machines and Storage Devices

The first layer (virtual machine) is very secure since the “hGUARD” will identify malware to avoid tracking and monitoring the virtual machine environment. “hGUARD” will monitor all actions such as writing and running access permissions on each storage or database to prevent malware injection attacks when unauthorized third-party activities are performed. “hGUARD” will analyze the activity of the virtual machine as well as the data that is being stored in the storage. If there is no suspicious activity, then the virtual machine and storage are secure. If any questionable activity is seen, “hGUARD” will keep an watch out for it and investigate further. If an assault is detected, “hGUARD” will intervene and resolve the situation.

2.2 Unmonitored Traffic

The data entering and leaving the hypervisor is screened at this layer to prevent threats from attacking the hypervisor. The contaminated data may be able to transmit the virus to the hypervisor. To avoid a scenario like this, all the identified risks will be eliminated or prevented from entering the process altogether. The attack in the traffic can be identified if there is any suspicious link, file or network connection being created or connection request received.

Table 1. Algorithm 1: Detecting Injection Attacks on Hypervisors

```
Input: Activity
Output: Suspicious or Normal
Initialization of VM's in physical machine from $VM_1$ to $VM_n$
$\forall$ $VM_{1\ to\ n}$ , ping each VM
If response=0 "VM not responding" Specious
Monitor and Track VM
    Record VM number
    Stop all API calls
    Shutdown VM
Else response=1 "Normal"
Update Database
//write access
Compute the Checksum of the Data
Store data into storage device
Recompute the Checksum of Data on Device
If Checksum =1 "Suspicious"
Deny access permission
Else Access granted
Update Database
// Uncontrolled traffic monitoring
Check whether the data traffic is safe.
If suspicious link, file, connection is created or received =Block Traffic
Else Normal Traffic
Update Database
// Unprotected host
If SYN flood, multiple port scanning detected, activity=Malicious
Else Normal
Update Database
If attack on cloud, cloud=1
Else Normal
Update Database
If attack on Switch, Switch=1
Else Normal
Update Database
If attack on Server, Switch=1
Else Normal
Update Database
Generate Alert "VM Attacked"
End
```

2.3 No Security in the Host

Because there are many resources engaged in the host layer, “hGUARD” will keep track of any suspicious actions that occur in this layer. This is because the security of the host is important since it will have an impact on many levels, including the server, storage, and the end-user. SYN flood frequently, compromising mail server through port scan, implanting an trojan in intranet host are some of the activities which attacks host. If malware is identified, a warning notice will be shown. As a part of the entire process, “hGUARD” may anticipate potential attacks based on the data collected. “hGUARD” is responsible for protecting the host from being attacked. The host will always be in charge of monitoring. Whenever a danger is detected, “hGUARD” will notify the user of the situation.

We performed an experiment utilizing data mining association rules in a hyper-vision environment with ten computers with windows as operating system to assess the m performance analysis of "hGUARD" running. Data mining is a process of extracting useful information from various sources. In the context of malware detection, data mining is used to extract association rules from the data records to identify new malware threats. This experiment aims to study and anticipate the behavior of malware in a hypervisor environment. The hypervisor layer and malware behavior are critical components of malware attack analysis. They help determine which layer is most likely to be targeted by malware and how that assault may impact other system layers. When malware attacks, we examine the dependability of all six levels, including the hardware, the host, the storage, the switch, the server, and the cloud. The malware effect on each layer that was attacked has been recorded.

3 Results and Discussion

Table 2. Malware detection in each layer

Machine ID	Hardware	Host	Storage	Switch	Server	Cloud
100	1	0	1	0	0	0
101	0	1	0	0	1	1
102	1	1	1	0	1	1
103	0	1	0	1	1	1
104	0	0	1	0	0	0
105	1	1	1	1	1	1
106	0	0	0	0	1	1
107	0	0	1	1	0	0
108	1	1	0	0	1	1
109	0	0	0	0	0	0

The experiments were conducted using a personal computer having core i7 process with 8 GB RAM. Table 2 illustrates Microsoft Excel was used to convert the data into a relational table structure utilized for the experiment. There are rows for malware that exists and columns for layers that exist in the table. The presence or absence of a malware attack in each layer is represented by the Boolean values 0 and 1. A value of 0 indicates that no malware has been discovered, whereas a value of 1 indicates that a malware attack has occurred on the layer. We then used the Apriori method in WEKA to conduct association rule mining on the previously discussed data set. WEKA is a tool containing machine learning algorithms for data mining tasks developed by university of Waikato New Zealand[18].

Figure 3 shows the Apriori algorithm best rule generation in WEKA. It was necessary to utilize the Apriori method for mining frequent item sets for Boolean association rules in this experiment since we have Boolean values in this experiment that are both 0 and 1 to indicate two kinds of conditions:

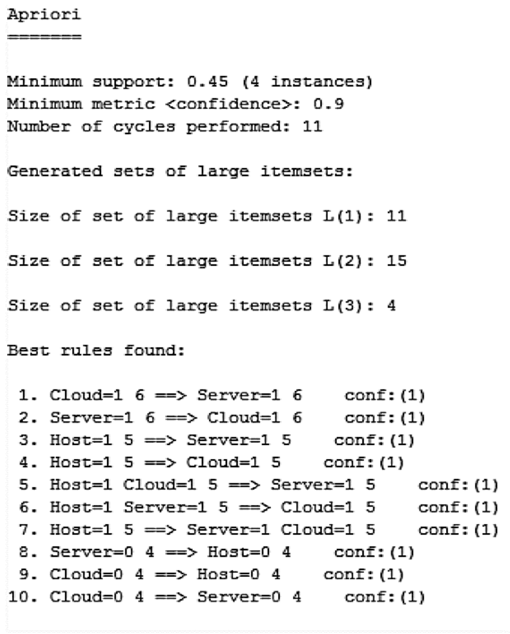
There is a malware attack, 1

There is no malware attack, 0

we selected Apriori algorithm for its widespread recognition, its role as a benchmark algorithm, having suitability, accessibility in libraries, alignment with our study's scope, and its efficiency in handling datasets. The parameters utilized for the Apriori algorithm encompassed a minimum support of 0.45, a minimum metric confidence of 0.9, and a total of 11 cycles performed.

As a consequence of the findings depicted in Figure 4, it can be concluded that malware is most likely to be found in the cloud, on the server-side, and on the host. These three layers are interconnected, which means that if a malware attack happens on the cloud, it has a high likelihood of affecting both the server site and the host computer. When a malware attack happens at the server-side, it has the same effect as at the cloud and host. The purpose of this experiment is to demonstrate at what level malware attacks are most common and how much of an impact they may have on other layers. Unfortunately, it is not possible to completely exclude the possibility of a virus attacks on the other layer.

We visualize the result of the association rules in Figure 4. It showed that host, server, and cloud appeared in most of the rules, which will show they are the most critical layers where malware tends to attack compared to other layers in the hypervisor and are associated with each other. The y-axis represents the frequency of layers appearing in rules while x-axis represents ten rules found in WEKA. Each layer is differentiated by six different colors as can be seen below. In summary, "hGUARD" performance depends on at which layer usually malware tends to attack and its effect to another layer.



```
Apriori
=======

Minimum support: 0.45 (4 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 11

Generated sets of large itemsets:

Size of set of large itemsets L(1): 11

Size of set of large itemsets L(2): 15

Size of set of large itemsets L(3): 4

Best rules found:

 1. Cloud=1 6 ==> Server=1 6    conf:(1)
 2. Server=1 6 ==> Cloud=1 6    conf:(1)
 3. Host=1 5 ==> Server=1 5    conf:(1)
 4. Host=1 5 ==> Cloud=1 5    conf:(1)
 5. Host=1 Cloud=1 5 ==> Server=1 5    conf:(1)
 6. Host=1 Server=1 5 ==> Cloud=1 5    conf:(1)
 7. Host=1 5 ==> Server=1 Cloud=1 5    conf:(1)
 8. Server=0 4 ==> Host=0 4    conf:(1)
 9. Cloud=0 4 ==> Host=0 4    conf:(1)
10. Cloud=0 4 ==> Server=0 4    conf:(1)
```

Fig 3. Apriori algorithm best rule generation in WEKA

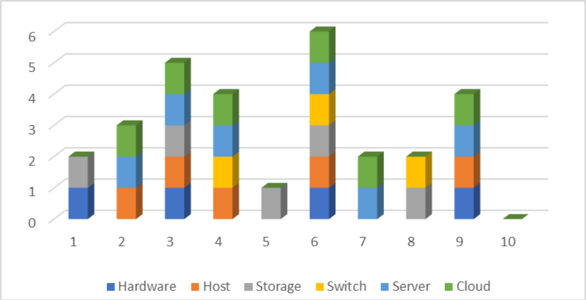


Fig 4. Associate rules against each layer found in WEKA

Further, the experimental results obtained for defending against attacks on the hypervisors using this approach are depicted in Table 3 and the comparison with existing approaches shows that the proposed approach performed better.

Table 3. Experimental results and comparison with existing approaches

Metrics	Description	"hGuard" Approach	Study [11]	Study [17]
Detection Accuracy	Percentage of correctly identified malware injection attacks.	95%	85%	88%
False Positive Rate	Percentage of non-attack instances incorrectly classified as attacks.	3%	10%	7%
False Negative Rate	Percentage of actual attacks incorrectly classified as non-attacks.	5%	15%	12%
Execution Time	Time taken by the approach to process and predict attacks.	20 ms	25 ms	30 ms

The proposed "hGuard" approach stands out prominently in countering malware injection attacks within cloud computing, displaying exceptional performance metrics compared to Studies [11] and [17].

4 Conclusion

This study focused on attacks on various hypervisor layers in order to address security issues in cloud computing. The ground-breaking ”hGuard” approach was propsoed, providing a thorough protection mechanism against malware attacks at various hypervisor levels. To create and validate the ”hGuard” approach, the study combined theoretical analysis with actual experimentation. The method successfully predicted and stopped possible malware attempts across hypervisor levels by applying a data mining algorithm like Apriori. The study's empirical results showed a stunning 95% detection accuracy, a low 3% false positive rate, and a 5% false negative rate, all attained in a quick 20 ms execution time. The uniqueness of the research came from fixing significant security approach holes. The ”hGuard” solution offered a comprehensive defense against multi-level attacks, in contrast to typical strategies that address hypervisor levels individually. The Apriori technique's incorporation enhanced its originality by providing a data-driven strategy for preventative cybersecurity. Empirical confirmation strengthened its originality and established it as a useful tool for identifying and combating malware threats in cloud computing. The study also offered ways to broaden the ”hGuard” method's use to several cybersecurity fields. Overall, this research considerably strengthens cloud security and paves the way for improvements in proactive cybersecurity methods. Future directions include parameter optimization, hybrid solutions, scalability, and real-world implementation. This research strengthens cloud security and lays groundwork for proactive cybersecurity advancements.

References

1) Gill SS, Tuli S, Xu M, Singh I, Singh KV, Lindsay D, et al. Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges. *Internet of Things*. 2019;8:100118–100118. Available from: https://www.sciencedirect.com/science/article/abs/pii/S2542660519302331.
2) Tabrizchi H, Rafsanjani MK. A survey on security challenges in cloud computing: issues, threats, and solutions. *The Journal of Supercomputing*. 2020;76(12):9493–9532. Available from: https://link.springer.com/article/10.1007/s11227-020-03213-1.
3) Khan NA, Qurashi MA. Security Tradeoff in Network Virtualization and Their Countermeasures. In: Inventive Computation and Information Technologies. Springer Nature Singapore. 2023;p. 741–749. Available from: https://link.springer.com/chapter/10.1007/978-981-19-7402-1_52.
4) Gupta B, Mishra N. Optimized deep learning-based attack detection framework for secure virtualized infrastructures in cloud. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*. 2022;35(1):2945. Available from: https://onlinelibrary.wiley.com/doi/full/10.1002/jnm.2945.
5) Lin K, Liu W, Zhang K, Xia H, Tu B. HyperKRP: A Kernel Runtime Security Architecture with A Tiny Hypervisor on Commodity Hardware. *2021 IEEE Global Communications Conference (GLOBECOM)*. 2021;p. 1–6. Available from: https://ieeexplore.ieee.org/abstract/document/9685552.
6) Bhardwaj A, Kaushik K, Dagar V, Kumar M. Framework to measure and reduce the threat surface area for smart home devices. *Advances in Computational Intelligence*. 2023;3(4):16. Available from: https://link.springer.com/article/10.1007/s43674-023-00062-2.
7) Alqarni AA, Alsharif N, Khan NA, Georgieva L, Pardade E, Alzahrani MY, et al. Modular Neural Network Based Approach for XSS Attack Detection. 2022. Available from: https://www.techscience.com/cmc/v70n2/44704/pdf.
8) Alarfaj FK, Khan NA. Enhancing the Performance of SQL Injection Attack Detection through Probabilistic Neural Networks. *Applied Sciences*. 2023;13(7):4365. Available from: https://www.mdpi.com/2076-3417/13/7/4365.
9) Aalam Z, Kumar V, Gour S. A review paper on hypervisor and virtual machine security. *InJournal of Physics: Conference Series*. 2021;1950:12027. Available from: https://iopscience.iop.org/article/10.1088/1742-6596/1950/1/012027/meta.
10) Fischer A, Kittel T, Kolosnjaji B, Lengyel TK, Mandarawi W, De Meer H, et al. CloudIDEA: A Malware Defense Architecture for Cloud Data Centers. In: Lecture Notes in Computer Science. Springer International Publishing. 2015;p. 594–611. Available from: https://link.springer.com/chapter/10.1007/978-3-319-26148-5_40.
11) Liakos KG, Georgakilas GK, Moustakidis S, Sklavos N, Plessas FC. Conventional and machine learning approaches as countermeasures against hardware trojan attacks. *Microprocessors and Microsystems*. 2020;79:103295–103295. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0141933120304543.
12) Khan NA, Network. 5G Network: techniques to Increase Quality of Service and Quality of Experience. *International Journal of Computer Networks and Applications*. 2022;9(4):476. Available from: https://www.ijcna.org/Manuscripts/IJCNA-2022-O-39.pdf.
13) Anwar RW, Abdullah T, Pastore F. Firewall Best Practices for Securing Smart Healthcare Environment: A Review. *Applied Sciences*. 2021;11(19):9183. Available from: https://www.mdpi.com/2076-3417/11/19/9183.
14) Soleymanpour S, Sadr H, Soleimandarabi MN. CSCNN: Cost-Sensitive Convolutional Neural Network for Encrypted Traffic Classification. *Neural Processing Letters*. 2021;53(5):3497–3523. Available from: https://link.springer.com/article/10.1007/s11063-021-10534-6.
15) Bacs A, Giuffrida C, Grill B, Bos H. Slick: an intrusion detection system for virtualized storage devices. *InProceedings of the 31st Annual ACM Symposium on Applied Computing*. 2016;p. 2033–2040. Available from: https://research.vu.nl/en/publications/slick-an-intrusion-detection-system-for-virtualized-storage-devic.
16) Alshehri A, Khan N, Alowayr A, Alghamdi MY. Cyberattack Detection Framework Using Machine Learning and User Behavior Analytics. *Computer Systems Science and Engineering*. 2023;44(2):1679–1689. Available from: https://www.techscience.com/csse/v44n2/48277.
17) Khan NA, Khan AS, Kar HA, Ahmad Z, Tarmizi S, Julaihi AA. Employing Public Key Infrastructure to Encapsulate Messages During Transport Layer Security Handshake Procedure. *2022 Applied Informatics International Conference (AiIC)*. 2022;p. 126–130. Available from: https://ieeexplore.ieee.org/document/9914605.
18) WEKA, University of Waikato, New Zealand. 2020. Available from: https://waikato.github.io/weka-wiki/downloading_weka/.