Enhancing Cloud Security: A Hypervisor-Level Framework for Malware Prevention and Detection

AnamikaAgarwal[1], SatyaBhushanVerma[2], Bineet Kumar Gupta[3] , Shobhit Mani Tiwari [4] and Sunil Kumar Patel[5]

[1] Shri Ramswaroop Memorial University, Barabanki, India 225003agrawal.anamika18@gmail.com
[2] Department of Computer Science and Engineering,Shri Ramswaroop Memorial University, Barabanki, India 225003 satyabverma1@gmail.com
[3] Shri Ramswaroop Memorial University, Barabanki, India 225003bkguptacs@gmail.com
[4] Lucknow University , Lucknow, India 226001tiwari_shobhit@lkouniv.ac.in
[5]Department of Computer Science & Engineering, Manipal University Jaipur, Jaipur, Rajasthan, India – 303007 sunilpatel.bsb@gmail.com

Abstract: The introduction outlines the significance of data storage and management in businesses and highlights the cost-effectiveness of data outsourcing, particularly in the context of cloud computing. It emphasizes the benefits and challenges of cloud computing, with a focus on security concerns related to malware threats. The proposed malware prevention and detection (MP&D) framework aims to address these challenges by providing a comprehensive set of processes and tools to secure virtual machines in cloud computing environments. The framework's objectives include large-scale system management, identifying various attacks, early detection, rapid and accurate detection, scalability, and resistance to compromise. The section also mentions the prevalence of encrypted and modern malware, which poses additional challenges. The background procedures and tools for the MP&D framework are briefly outlined, along with a high-level algorithmic overview of its key components. The expected results of implementing the framework are discussed, highlighting improvements in security posture, reduced malware infections, early threat detection, faster incident response, and compliance benefits. The conclusion underscores the importance of robust security measures in cloud computing, particularly at the hypervisor level, and acknowledges the evolving nature of malware threats, indicating the need for ongoing research and development in this field.

Keywords: Virtualization, Hypervisor, malware, Security, ransomware, Threat Detection, Data encryption, encrypted traffic**.**

I. Introduction

In today's digital age, data storage and maintenance stand as the cornerstone of modern businesses. The exponential growth of data generated necessitates efficient storage solutions, but the traditional methods often come with significant costs[1]. Hardware procurement, maintenance, and ongoing management pose financial challenges for organizations striving to stay competitive in dynamic markets. However, the emergence of cloud computing has revolutionized the landscape, offering cost-effective alternatives that enhance accessibility, scalability, and disaster recovery capabilities[2]. Outsourcing data management to specialized cloud providers has become increasingly popular among businesses seeking to optimize their operations. Cloud computing, through its distributed network infrastructure and virtualized resources, offers a myriad of benefits[3]. By leveraging economies of scale, cloud providers can offer storage and processing solutions at a fraction of the cost compared to on-premises infrastructure. Moreover, the pay-as-you-go model eliminates the need for large upfront investments, providing scalability and flexibility tailored to the evolving needs of businesses[4]. Cloud computing transcends traditional storage limitations, enabling seamless access to data from anywhere in the world. This accessibility fuels innovation, empowering organizations to leverage data analytics, machine learning, and artificial intelligence to extract valuable insights and drive informed decision-making[5]. Furthermore, the cloud's high availability and disaster protection mechanisms ensure business continuity, mitigating the risks associated with hardware failures or natural disasters. Despite its numerous advantages, cloud computing is not without its challenges. Security remains a paramount concern, particularly in the context of virtualized environments[6]. The inherent shared nature of cloud infrastructure introduces vulnerabilities that malicious actors may exploit to compromise data confidentiality, integrity, and availability. The National Vulnerability Database (NVD) highlights numerous vulnerabilities associated with virtualization implementations, including hypervisors and virtual machines. The proliferation of malware targeting cloud services further exacerbates security concerns. Over the past decade, the AV-TEST Institute has recorded a staggering increase in the number of dangerous malware variants, posing a significant threat to cloud-based systems. Figure 2 illustrates the alarming evolution of malware, emphasizing the pressing need for robust security measures to safeguard cloud environments[7]. Protecting cloud computing environments from malware and other system-level threats requires a multi-faceted approach. Traditional malware detection methods, while effective to a certain extent, struggle to cope with the diverse and dynamic nature of cloud infrastructures. Challenges such as efficient attack detection, scalability, and maintaining resistance to compromise necessitate innovative solutions tailored to the unique characteristics of cloud environments. In

conclusion, while cloud computing offers compelling advantages in terms of cost-efficiency and scalability, security concerns cannot be overlooked[8]. Businesses must strike a delicate balance between harnessing the benefits of cloud technology and mitigating potential risks. Proactive measures, including robust security protocols, threat intelligence, and continuous monitoring, are imperative to safeguarding data integrity and ensuring business continuity in an increasingly interconnected digital ecosystem[9]. By embracing a holistic approach to cloud security, organizations can unlock the full potential of cloud computing while safeguarding against emerging threats and vulnerabilities.

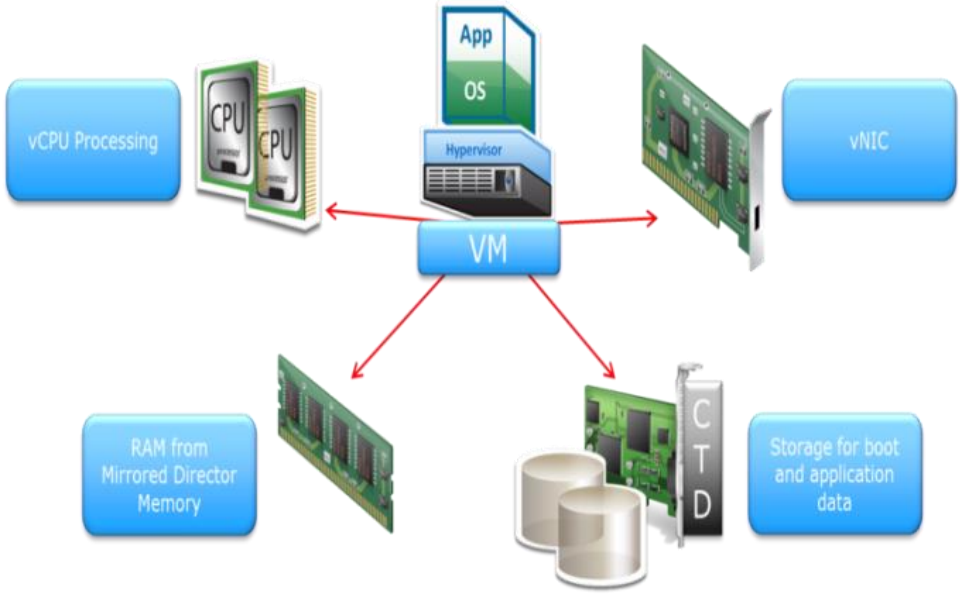

Figure 1. Virtualization using Hypervisor

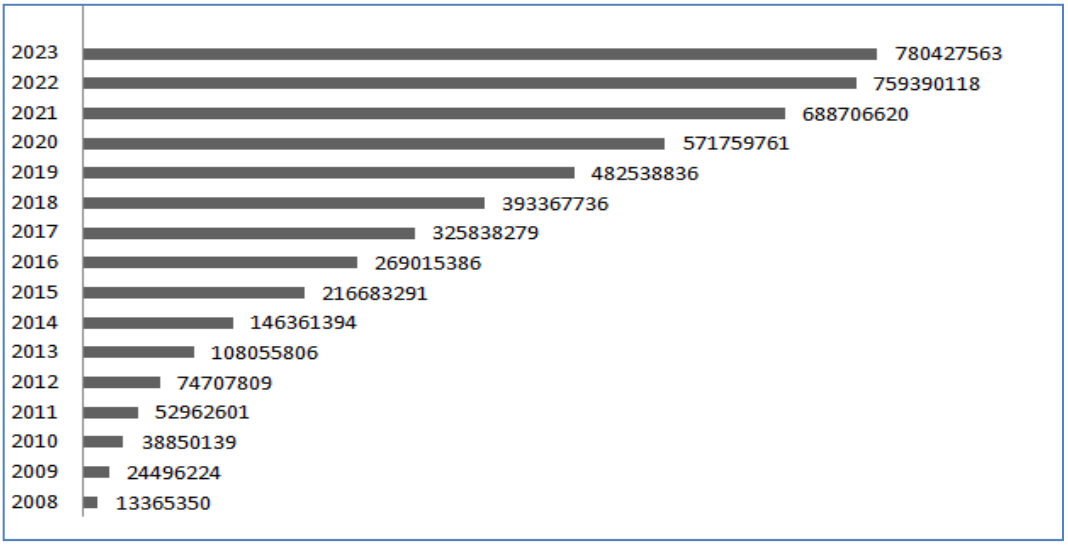

Figure 2. Total amount of malware per year [10]

A. Objective Of MP&D

The proposed MP&D framework aims to secure virtual machines by finding malicious executables in cloud computing's high-risk virtual machines while meeting the following security requirements:

1) Large-scale computer system management

Single-virtual machine security solutions cannot protect the cloud computing environment's virtual environment. It is necessary to take into account the security of the underlying VMs because they have various operating system-specific security needs.

2) Identifying various attacks

At the virtual machine layer, signature-based security solutions may identify known malware[10]. However, in the cloud, each virtual machine has individual security needs. As a result, malware's signature database needs to be updated on a regular basis. The anomaly detection system is capable of finding unidentified malware.

However, it struggles with a high number of false alarms. Therefore, malware detection software should be able to identify both common attacks and new ones. It should also correlate malware alarms from many virtual machines to find any distributed or coordinated attacks.

3) Early malware detection

The suggested framework should be able to detect malware early on without actually running it. This aids in thwarting potential assaults.

4) Rapid detection and great precision

The suggested framework ought to be quick enough to accurately and quickly detect malware given the large-scale computing systems and dynamic cloud behavior. To increase detection accuracy, it is necessary to discover the best features and use workable machine learning approaches.

5) Excellent scalability Multiple

VMs are being watched at once by the hypervisor-based security mechanisms. A new VM may be dynamically added to or withdrawn from each server for load balancing and the dynamic deployment of VMs. VMs can also be swiftly rolled back to their original state. In this situation, the suggested architecture must be scalable to handle the dynamic load associated with the shifting actions of VMs.

6) Compromise resistance

The security framework should be able to fend off compromise. Therefore, to make it more difficult for an attacker to compromise the security framework, it should be segregated from a VM or operated with higher privileges.

7) 60% of all malware assaults used encrypted traffic to communicate

Malware assaults are frequently sent by threat actors over SSL/TLS traffic that is encrypted. The success rates for the malware packages under consideration are higher as a result of encrypted channels' increased difficulty in identification and mitigation. But according to WatchGuard [23], this type of assault accounted for 60.1% of all malware discovered in Q1 2022, down from 91.1% in Q2 2021.

8) Advance/Modern Malware

Modern malware is capable of changing its appearance to avoid detection by signature-based systems and is more adaptable than before. Attackers are able to regularly modify a malware file's binary structure so that it seems different to antivirus software by using techniques they refer to as "packing and crypting," or P&C. The malicious executable still performs the exact same action, but because it appears to be a new file, AV programmed might not detect it because it was previously known to exist. Signature-based antivirus simply cannot keep up with the hundreds of millions of new malware variants that are identified every year.

II. Review of Existing Literature

The cloud malware detection system continuously monitors and analyzes VM service and resources for potential threats that could compromise cloud security. Several efforts have been made to detect malware in cloud computing:

Angelos et al. [12] proposed a hypervisor-based malware detection method using ensemble empirical mode decomposition (E-EMD), but it's designed for single VMs and may not be suitable for large-scale systems. Fattori et al. [13] introduced Access Miner, a system-centric behavioral malware detector that looks at program behavior and operating system interactions in real-time, but it generates many false alarms and struggles with sophisticated viruses. Watson et al. [14] suggested a strategy for identifying cloud anomalies using a one-class support vector machine (SVM) on the hypervisor, but it focuses on a single VM. Mishra et al. [15] proposed a system call analysis method called "Malicious System Call Sequence Detection (MSCSD)" for identifying malware, but findings on sophisticated malware are yet to be published. Xie and Wang [16] introduced a malware detection method that examines DLL files and their running environment in guest VMs, but it doesn't provide early virus detection. Ajay and Jaidhar [17] presented an automated internal-external malware detection system but face security and executable processing issues. Jia et al. [18] suggested FindEvasion, a cloud-based approach for identifying environment-sensitive malware but it's time-consuming and challenging for large-scale systems. Mishra et al. [19] proposed a VMI-based evasion detection (VAED) system but it can't detect advanced malware and needs evaluation against recent datasets. Xu et al. [20] suggested a hardware-assisted malware detection system that classifies harmful behavior based on virtual memory access patterns but generates false positives. Joseph and Mukesh [21] recommended VM memory snapshot-based malware detection methods but require improved detection accuracy. Patil et al. [22] proposed an agent-based malware

detection (AMD) framework but cannot detect encrypted malware like ransomware. In terms of data security, access control, and other challenges, cloud computing represents a new paradigm for computers [14].

Over the past 10 years, a lot of survey research have been published regarding the security issues with cloud computing. Additionally, it is clear that most of the reviewed papers provided play a significant role in cloud security difficulties, and these noteworthy assessed works have been very beneficial and complete research in this area. Kavin et al.[35] By offering safe cloud storage and retrieval, it is possible to efficiently guarantee secure data transfer. In terms of security level and performance, the trial findings also demonstrated the effectiveness of the suggested strengthened security architecture. Tabrizchi et al.[36] a study of current security frameworks has been suggested with the goal of decreasing vulnerabilities to thwart potential assaults. To lessen risk and susceptibility and boost confidence in a constantly linked world, the article gave a number of written rules, procedures, and processes that define the safe management approach in the cloud environment. In order to protect high-risk VMs from malware at an early point of the VMlife cycle, Patil et al. [37] have presented an in-VM-assisted lightweight agent-based malware detection system. It utilises both signature-based detection at the VM layer and anomaly detection at the hypervisor layer, allowing it to identify both known and unidentified malware. As an added bonus, it only provides profiles of malware that is unknown to the hypervisor layer for anomaly detection, hence minimising communication cost. In addition to demonstrating a way for resolving security issues in a multi-tenant environment, Kumar et al. [38] also highlighted a variety of data security obstacles in cloud computing. Actually, the entire focus of this essay is on issues with data security and methods for maintaining privacy and data security.

Ransomware attack vectors used for financial gain and threat actors hunting for new attack surfaces were analysed by Naik1 et al. [39]. This was done by disseminating an endless amount of polymorphic ransomware samples while the offenders eluded detection and remained anonymous. Additionally, it is examined whether the authors of ransomware threat actors share a common writing style or other traits that might be used to help identify them. Grouping several ransomware samples based on little to no information about them allows researchers to investigate and identify their features and signatures, which is the first step in trying to identify the attack's source. In order to organise ransomware samples, this research advises employing two fuzzy techniques: fuzzy hashing and fuzzy c-means (FCM) clustering. Unlike other clustering techniques, FCM does not require additional transformational steps to gather object distance in order to arrange items into comparable groups. The similarity scores generated by a fuzzy hashing method may be used directly instead. It reduces the computational overheads by employing fuzzy similarity scores gathered during the first triage of whether the sample is known or unknown ransomware. The performance of the suggested fuzzy technique is contrasted against k-means clustering and the two fuzzy hashing methods SSDEEP as well as SDHASH, which are assessed based on their FCM clustering results, in order to understand how the similarity score effects the clustering outcomes. R. Wilson and I. Iftimie [40] presented that Ransomware is the sort of cybercrime that is expanding the quickest and is a significant security risk for businesses. Data on a hacked computer is encrypted by conventional ransomware, preventing users from accessing it until a ransom is paid. Double-extortion ransomware operations, in which files are first seized and then encrypted, can still generate income by extorting the firm or, if a higher ransom is not paid, by selling any sensitive material on the dark web.

Enterprises cannot be shielded from this new species of data theft and extortion firm by firewalls, antivirus software, and regular backups. To find and destroy this danger, intelligence analysts and proactive cyber threat hunters are required. Singh ,et al.[41] suggests a software-defined perimeter (SDP) as a way to safeguard IaaS, providing a logical boundary with authentication and authorization to restrict access to services. Results show that SDP can withstand DoS attacks and still permit legitimate user traffic even while the attack is ongoing. PanJun Sun[42] discussed Cloud computing security and privacy protection, with research progress on privacy security challenges and a strategy for protecting against risks. Rakesh Kumar,and Rinkaj Goyal [43] This survey offers an integrationist end-to-end mapping of cloud security requirements, detected threats, known vulnerabilities, and recommended solutions, as well as a uniform taxonomy for security requirements, threats, vulnerabilities, and solutions.

These malware detection approaches need to be evaluated with recent malware in mind, considering factors like secure component placement, memory snapshot-based detection, early analysis, accuracy, false alerts, and encrypted malware detection in cloud computing.

III. Proposed malware prevention and detection (MP&D) framework

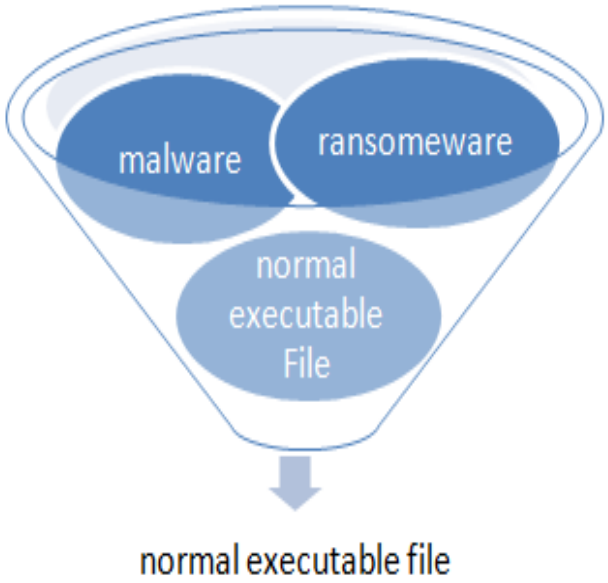

Figure 3. MP&D acting as a Funnel

A. Background procedures and tools

Implementing a complete malware prevention and detection (MP&D) algorithm involves a complex set of processes and techniques. The MP&D framework acts as a funnel that filters all types of executable files and stops all types of malwares, as shown in Figure 3. The process flow of the proposed MP&D framework in the virtual environment of the cloud is shown in the flowchart. While the specifics can vary depending on the context and requirements, here is a high-level outline of an algorithm that encompasses key MP&D components as given in Fig.4 elaboration of key points are given below :

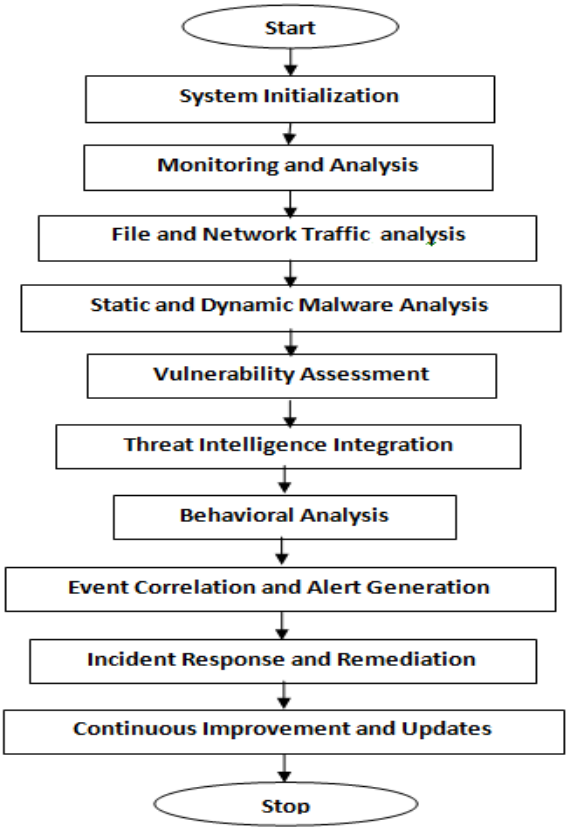

Figure 4. Flowchart that encompasses Malware Prevention and Detection (MP&D) algorithm

Step 1) System Initialization

- Set up the necessary data structures, configurations, and resources for MP&D.
- Load malware signatures and indicators of compromise (IOCs) into a database or memory for reference.

Step2) Monitoring and Analysis

- Continuously monitor system activities, network traffic, and user behavior for anomalies or suspicious patterns.
- Capture system logs, network packets, and relevant events for analysis.

Monitoring system activities, network traffic, and user behavior for anomalies or suspicious patterns at the hypervisor level in a cloud environment requires specialized tools and techniques. Here's an outline of an algorithm to achieve continuous monitoring at the hypervisor level in the cloud:

(a) Access Hypervisor API: Gain access to the hypervisor API provided by the cloud platform or virtualization technology being used. Obtain necessary authentication credentials or API keys to interact with the hypervisor.

(b) Retrieve System Activity Logs: Utilize the hypervisor API to retrieve system activity logs and events from virtual machines (VMs) running on the hypervisor. Collect information such as VM start/stop events, resource usage, and administrative actions.

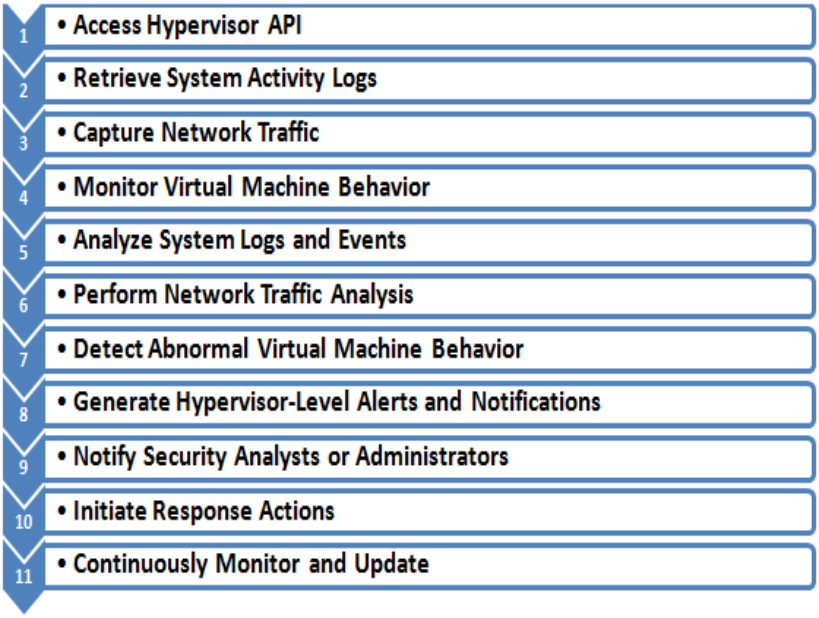

Figure 5. Outline to achieve continuous monitoring at the hypervisor level in the cloud.

(c) Capture Network Traffic: Configure the hypervisor to capture network traffic at the virtual switch or network interface level. Retrieve packet-level data or network flow information passing through the hypervisor.
(d) Monitor Virtual Machine Behavior: Continuously monitor the behavior of VMs hosted on the hypervisor. Collect data on processes, file access, network connections, and other relevant activities within each VM.
(e)Analyze System Logs and Events: Apply log analysis techniques to the system activity logs obtained from VMs. Employ statistical analysis, machine learning, or rule-based methods to identify anomalies or suspicious patterns. Look for events indicating potential security incidents or abnormal behavior within VMs.
(f)Perform Network Traffic Analysis: Analyze captured network traffic data at the hypervisor level. Employ intrusion detection/prevention systems (IDS/IPS), network behavioral analysis, or signature-based detection techniques. Identify patterns indicative of malicious activities, such as network scans, malware communications, or data exfiltration.
(g) Detect Abnormal Virtual Machine Behavior: Compare the behavior of individual VMs against established baselines or normal patterns. Identify deviations, such as unusual resource usage, unauthorized access attempts, or suspicious process activities. Utilize anomaly detection algorithms or behavior analytics to identify potentially compromised VMs.
(h) Generate Hypervisor-Level Alerts and Notifications: Generate alerts or notifications at the hypervisor level when anomalies or suspicious patterns are detected. Prioritize alerts based on severity levels or predefined rules. Include relevant information, such as VM identifiers, timestamps, or affected resources, in the alerts.
(i) Notify Security Analysts or Administrators: Deliver alerts or notifications to security analysts or system administrators responsible for managing the cloud environment. Use appropriate communication channels, such as email, SMS, or dedicated monitoring consoles.Ensure timely delivery and escalation of alerts based on predefined procedures and incident response plans.
(j) Initiate Response Actions: Take appropriate response actions based on the severity and nature of the detected anomalies or suspicious patterns. Isolate affected VMs, suspend suspicious network connections, or initiate incident response procedures as required. Collect additional evidence or perform forensic analysis at the VM or hypervisor level, if necessary.
(k) Continuously Monitor and Update: Repeat the monitoring process at the hypervisor level on an ongoing basis. Update monitoring techniques, rules, or algorithms based on evolving threats and changes in the cloud environment. Regularly review and refine the monitoring strategy to enhance the detection and prevention of anomalies or malicious activities.
Implementing the algorithm would require knowledge of the specific hypervisor API, virtualization technology, and cloud platform being used. Additionally, it's important to consider any limitations or constraints imposed by the cloud provider regarding hypervisor-level monitoring capabilities.

Step3) File and Network Traffic Analysis:
- Analyze incoming and outgoing files, email attachments, and network traffic for potential malware.

• Employ a combination of signature-based detection, heuristics, behavior analysis, and machine learning techniques.

Step4) Static and Dynamic Malware Analysis:
• Perform static analysis on files and executables, examining their characteristics, metadata, and structure.
• Execute suspicious files in controlled environments (e.g., sandboxes) to observe behavior and identify malicious actions.

Step 5) Vulnerability Assessment:
• Conduct regular vulnerability assessments to identify system weaknesses and potential entry points for malware attacks.
• Scan systems, applications, and network infrastructure for known vulnerabilities and misconfigurations.

An algorithm for conducting vulnerability assessment at the hypervisor level in a cloud environment:
(a) Access Hypervisor Management Interface: Gain access to the management interface or API provided by the cloud platform to interact with the hypervisor. Obtain necessary authentication credentials or API keys to establish a connection.
(b) Retrieve Hypervisor Configuration: Retrieve the configuration details of the hypervisor, including its version, patch level, and applied security updates. Obtain information about the virtualization technology being used (e.g., VMware, Hyper-V, KVM).
(c) Identify Common Vulnerabilities and Exposures (CVEs): Access publicly available vulnerability databases, such as the National Vulnerability Database (NVD) or Common Vulnerabilities and Exposures (CVE) database. Identify known vulnerabilities associated with the specific hypervisor version and virtualization technology.
(d) Perform Vulnerability Scanning: Utilize a vulnerability scanning tool capable of scanning the hypervisor and its components. Configure the scanner to target the hypervisor & IP address or management interface. Run the vulnerability scanner to identify potential vulnerabilities, misconfigurations , or weaknesses.
(e) Analyze Vulnerability Scan Results: Analyze the vulnerability scan results to identify critical, high, and medium-severity vulnerabilities. Prioritize vulnerabilities based on their potential impact and exploitability. Review the scan results for misconfigurations, insecure default settings, or outdated software versions.
(f) Cross-reference Vulnerabilities with Security Advisories: Cross-reference the identified vulnerabilities with security advisories or patches provided by the hypervisor vendor. Determine if patches or mitigations are available for the discovered vulnerabilities. Take note of any specific steps or recommendations provided by the vendor for addressing the vulnerabilities.
(g) Assess Hypervisor Hardening Configuration: Evaluate the hypervisor's security hardening configuration against established best practices and security guidelines. Ensure that security controls, such as access controls, resource isolation, and encryption, are properly configured and enforced. Identify any weaknesses or gaps in the hypervisor's security posture.
(h) Generate Vulnerability Assessment Report: Consolidate the findings from the vulnerability scan, security advisories, and hardening assessment into a comprehensive report. Include details of identified vulnerabilities, their severity, recommended remediation actions, and associated risks. Provide recommendations for patching, configuration changes, or additional security measures to mitigate the identified vulnerabilities.
(i) Prioritize and Remediate Vulnerabilities: Prioritize the identified vulnerabilities based on their severity, potential impact, and exploitability. Develop a remediation plan that includes patching, applying security updates, or implementing configuration changes. Coordinate with the cloud provider or relevant stakeholders to schedule and implement the necessary remediation actions.
(j) Perform Regular Vulnerability Assessments: Establish a regular schedule for conducting vulnerability assessments at the hypervisor level. Periodically repeat the vulnerability scanning process to identify new vulnerabilities, patch gaps, or misconfigurations. Stay up to date with the latest security advisories and patches released by the hypervisor vendor.
It's important to note that the specific tools, APIs, and techniques used in the vulnerability assessment process may vary based on the hypervisor technology and cloud platform being utilized. Be sure to refer to the documentation and guidelines provided by the hypervisor vendor and cloud provider for accurate and up-to-date information.

Step 6) Threat Intelligence Integration:
• Integrate with threat intelligence sources to receive real-time updates on emerging malware threats, IOCs, and attack patterns.
• Match observed behaviors and indicators against threat intelligence data to identify known malware.

Integrating threat intelligence at the hypervisor level in a cloud environment involves leveraging external threat intelligence sources to enhance the security posture of the hypervisor and protect the virtualized infrastructure. Here's an algorithm for integrating threat intelligence at the hypervisor level in the cloud:

(a) Access Hypervisor Management Interface: Gain access to the management interface or API provided by the cloud platform to interact with the hypervisor. Obtain necessary authentication credentials or API keys to establish a connection.

(b) Retrieve Hypervisor Configuration: Retrieve the configuration details of the hypervisor, including its version, patch level, and applied security updates. Ensure that the hypervisor is capable of integrating with external threat intelligence sources.

(c) Identify Relevant Threat Intelligence Sources: Identify reputable and reliable threat intelligence sources that provide up-to-date information on emerging threats, malware, and indicators of compromise (IoCs). Consider sources such as commercial threat intelligence feeds, open-source threat intelligence projects, government agencies, or industry-specific threat intelligence providers.

(d) Establish Integration with Threat Intelligence Sources: Configure the hypervisor to establish integration with the identified threat intelligence sources. Set up appropriate mechanisms to receive threat intelligence data, such as APIs, data feeds, or automated feeds.

(e) Retrieve and Process Threat Intelligence Data: Regularly retrieve threat intelligence data from the integrated sources. Receive indicators of compromise (IoCs), threat signatures, malicious IP addresses, domain names, file hashes, or other relevant information.

(f) Match Threat Intelligence Data with Hypervisor Environment: Compare the retrieved threat intelligence data with the hypervisor environment and virtualized infrastructure. Look for matches or overlaps between the threat intelligence data and the hypervisor's configuration, network traffic, or virtual machine activities.

(g) Generate Alerts and Notifications: Generate alerts or notifications when matches or overlaps are found between the threat intelligence data and the hypervisor environment. Prioritize alerts based on the severity of the threats and the potential impact on the cloud infrastructure. Include relevant information, such as IoCs, threat descriptions, and affected resources, in the alerts.

(g) Notify Security Analysts or Administrators: Deliver alerts or notifications to security analysts or system administrators responsible for managing the cloud environment. Utilize appropriate communication channels, such as email, SMS, or dedicated monitoring consoles. Ensure timely delivery and escalation of alerts based on predefined procedures and incident response plans.

(h) Take Response Actions: Initiate appropriate response actions based on the severity and nature of the identified threats. Isolate affected virtual machines, block network connections, or initiate incident response procedures as required. Apply security patches, updates, or configuration changes to mitigate the impact of the threats.

(i) Continuously Update Threat Intelligence Data: Regularly update the integrated threat intelligence data to ensure it reflects the latest threats and vulnerabilities. Establish a mechanism for automated or periodic retrieval of new threat intelligence updates. Stay informed about emerging threats and adjust the integration with threat intelligence sources accordingly.

It's important to note that the specific integration mechanisms and the choice of threat intelligence sources may vary depending on the hypervisor technology and cloud platform being used. Consult the documentation and guidelines provided by the hypervisor vendor and cloud provider for accurate and up-to-date information on integrating threat intelligence at the hypervisor level in your specific cloud environment.

Step 7) Behavioral Analysis:

- Analyze system behavior, process activities, network communications, and user actions to detect anomalous or malicious behavior.
- Utilize machine learning algorithms, anomaly detection, and statistical models to identify deviations from normal patterns.

Implementing behavioral analysis at the hypervisor level in a cloud environment involves monitoring and analyzing the behavior of virtual machines (VMs) and the hypervisor itself to detect abnormal or malicious activities. Here's an algorithm for conducting behavioral analysis at the hypervisor level in the cloud:

(a) Access Hypervisor Management Interface: Gain access to the management interface or API provided by the cloud platform to interact with the hypervisor. Obtain necessary authentication credentials or API keys to establish a connection.

(b) Retrieve Hypervisor Configuration: Retrieve the configuration details of the hypervisor, including its version, patch level, and applied security updates. Ensure that the hypervisor is capable of supporting behavioral analysis and monitoring.

(c) Define Baseline Behavior Profiles: Establish baseline behavior profiles for the normal activities of VMs and the hypervisor. Capture and analyze the typical behavior patterns, resource utilization, network traffic, and

process activities of VMs.
(d) Monitor VM and Hypervisor Activities: Continuously monitor the activities of VMs and the hypervisor at runtime. Collect information such as process execution, file access, network connections, and resource usage.
(e) Analyze Behavior Patterns: Apply behavior analysis techniques to the collected activity data. Utilize statistical analysis, machine learning, or rule-based methods to detect deviations from normal behavior.
(f) Identify Anomalies or Suspicious Patterns: Compare the observed behavior patterns against the established baseline profiles. Identify anomalies, outliers, or patterns that deviate significantly from normal behavior.
(g) Generate Alerts and Notifications: Generate alerts or notifications when anomalies or suspicious patterns are detected. Prioritize alerts based on the severity and potential impact of the identified behavior deviations. Include relevant information, such as VM identifiers, timestamps, or affected resources, in the alerts.
(h) Notify Security Analysts or Administrators: Deliver alerts or notifications to security analysts or system administrators responsible for managing the cloud environment. Utilize appropriate communication channels, such as email, SMS, or dedicated monitoring consoles. Ensure timely delivery and escalation of alerts based on predefined procedures and incident response plans.
(i) Perform Root Cause Analysis: Conduct further investigation and root cause analysis for detected anomalies or suspicious patterns. Collect additional contextual information, such as log files, system events, or network captures, for detailed analysis. Identify the underlying causes of the behavior deviations and assess their potential impact.
(j) Initiate Response Actions: Based on the severity and nature of the detected behavior deviations, initiate appropriate response actions. Isolate affected VMs, suspend suspicious network connections, or trigger incident response procedures as required. Apply security patches, updates, or configuration changes to mitigate the impact of identified malicious activities.
(k) Continuously Update Behavior Profiles: Regularly update the baseline behavior profiles to adapt to changes in the cloud environment and evolving threats. Analyze new normal behavior patterns and adjust the profiles accordingly. Stay informed about emerging attack techniques and adjust the behavior analysis techniques as necessary.
It's important to note that the specific techniques and tools used for behavioral analysis may vary based on the hypervisor technology and cloud platform being utilized. Consult the documentation and guidelines provided by the hypervisor vendor and cloud provider for accurate and up-to-date information on implementing behavioral analysis at the hypervisor level in your specific cloud environment.

Step 8) Event Correlation and Alert Generation:
- Correlate and analyze collected data from various sources, such as logs, network events, and malware detections.
- Generate alerts or notifications for potential malware incidents based on predefined rules or anomaly thresholds.

Step 9) Incident Response and Remediation:
- Provide appropriate response actions for detected malware incidents, including isolation, containment, and eradication.
- Automate or guide incident response processes, such as quarantining infected files, blocking network connections, and initiating remediation actions.

Step 10) Continuous Improvement and Updates:
- Regularly update malware signatures, IOCs, and detection algorithms based on new threat intelligence and research findings.
- Monitor system effectiveness, evaluate false positives/negatives, and refine detection and prevention mechanisms.

IV. Pseudocode For MP&DF

Pseudocode for the above flowchart is given below:

```
// Step 1: System Initialization
function systemInitialization()
{
  // Perform system initialization tasks, such as loading configuration files, setting up loggers, etc.
loadConfigurations();
initializeLoggers();
// Initialize necessary data structures and variables for other steps
initializeDataStructures();
```

```
}
// Step 2: Monitoring and Analysis
function monitoringAndAnalysis()
{
// Start monitoring network and system activities
startMonitoring();
// Analyze the monitored data for suspicious activities or anomalies
analyzeData();
}
// Step 3: File and Network Traffic Analysis
function fileAndNetworkTrafficAnalysis()
{
// Capture and analyze file traffic
captureFileTraffic();
analyzeFileTraffic();
// Capture and analyze network traffic
captureNetworkTraffic();
analyzeNetworkTraffic();
}
// Step 4: Static and Dynamic Malware Analysis
function staticAndDynamicMalwareAnalysis()
{
// Perform static analysis on suspicious files
performStaticAnalysis();
// If files show malicious behavior, proceed with dynamic analysis
 if (isMaliciousBehaviorDetected())
{
performDynamicAnalysis();
}
}
// Step 5: Vulnerability Assessment
function vulnerabilityAssessment()
{
// Scan systems and applications for vulnerabilities
scanForVulnerabilities();
    // Analyze the results and prioritize vulnerabilities based on severity
analyzeVulnerabilities();
}
// Step 6: Threat Intelligence Integration
function threatIntelligenceIntegration()
{
// Retrieve and integrate threat intelligence data
retrieveThreatIntelligenceData();
integrateThreatIntelligence();
}
// Step 7: Behavioral Analysis
function behavioralAnalysis()
{
 // Conduct behavioral analysis on system and user activities
conductBehavioralAnalysis();
}
// Step 8: Event Correlation and Alert Generation
function eventCorrelationAndAlertGeneration()
{
// Correlate events from different sources to identify potential threats
correlateEvents();
// Generate alerts for potential security incidents
generateAlerts();
}
// Step 9: Incident Response and Remediation
```

```
function incidentResponseAndRemediation()
{
 // Initiate incident response procedures for confirmed security incidents
initiateIncidentResponse();
// Implement necessary remediation actions to contain and resolve the incidents
implementRemediationActions();
}
// Step 10: Continuous Improvement and Updates
function continuousImprovementAndUpdates()
{
// Review and analyze the effectiveness of the security measures
reviewEffectiveness();
// Identify areas for improvement and plan necessary updates
planUpdates();
}
// Main function to execute all the steps in order
function main()
{
systemInitialization();
monitoringAndAnalysis();
fileAndNetworkTrafficAnalysis();
staticAndDynamicMalwareAnalysis();
vulnerabilityAssessment();
threatIntelligenceIntegration();
behavioralAnalysis();
eventCorrelationAndAlertGeneration();
incidentResponseAndRemediation();
continuousImprovementAndUpdates();
}
// Call the main function to start the security process
main();
```

It's important to note that the actual implementation of each step may involve utilising various technologies, frameworks, and programming languages, depending on the specific requirements and available resources. Additionally, algorithms can be augmented with additional features, such as cloud security, user education, and advanced threat hunting, to enhance MP&D capabilities.

V. Expected result of Mp&D

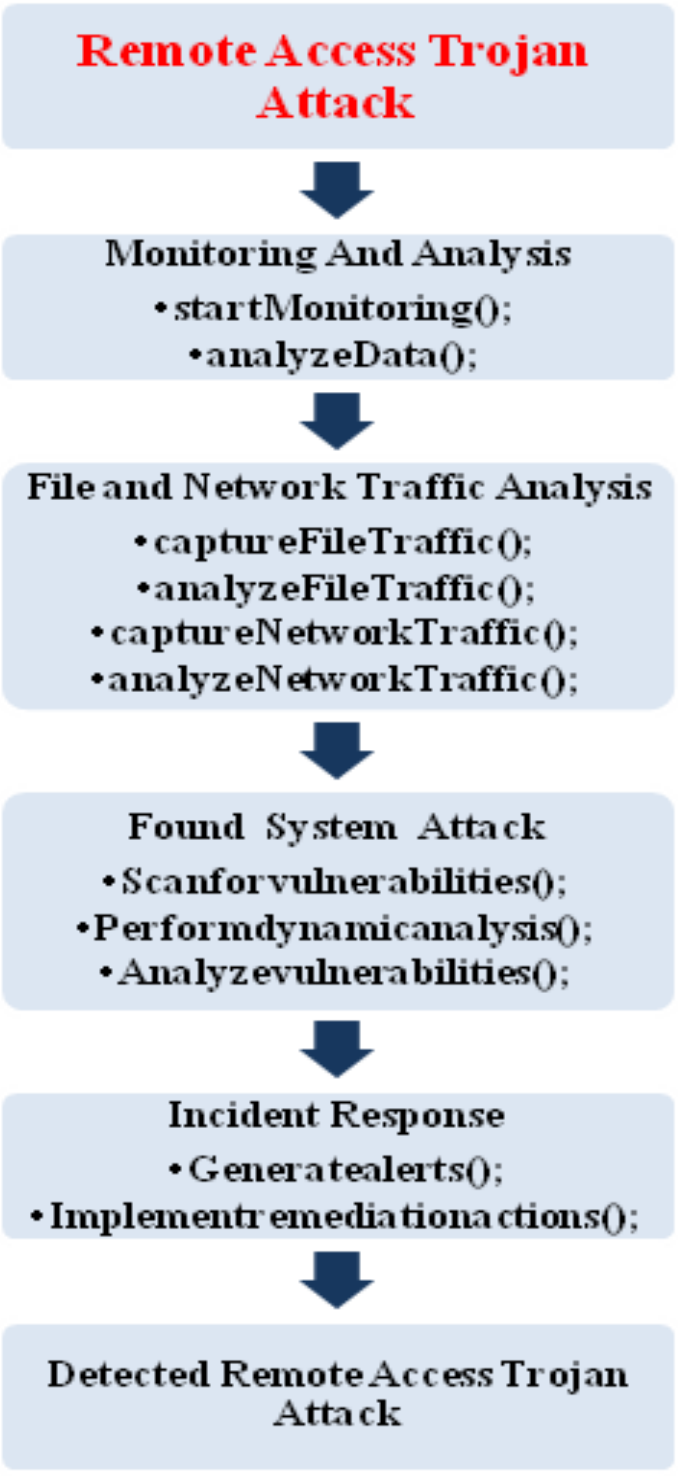

Figure 6.Action taken while Remote Action Trozen attack

MP&D frameworks often include real-time monitoring and analysis capabilities that enable the early detection of malware. By continuously monitoring system activities, network traffic, and user behavior, the framework can identify suspicious patterns, anomalies, or known indicators of compromise (IoCs) associated with malware activities. With a well-implemented MP&D framework, security incidents can be detected and responded to promptly. Automated alerting mechanisms and predefined incident response procedures allow security teams to take immediate action, minimizing the impact of malware infections and reducing recovery time. In the context of a ransomware attack (Figure 7), the monitoring and analysis process is vital. It involves capturing and analyzing file traffic through functions like CaptureFileTraffic() and AnalyzeFileTraffic().When a system attack is detected, the security process shifts towards vulnerability assessment using ScanForVulnerabilities()

and dynamic malware analysis with PerformDynamicAnalysis(). Once vulnerabilities are analyzed, the system can take action. In the event of an actual ransomware attack, incident response comes into play. It includes generating alerts through GenerateAlerts() and implementing remediation actions with ImplementRemediationActions() to mitigate the damage. The ultimate goal is to prevent the system from falling victim to the ransomware attack by quickly identifying, responding, and recovering from the threat.In the context of a Remote Access Trojan (RAT) attack (Figure 6), comprehensive monitoring and analysis are crucial. This process begins by initiating monitoring with startMonitoring() and analyzing system data. File and network traffic analysis, involving captureFileTraffic(), analyzeFileTraffic(), captureNetworkTraffic(), and analyzeNetworkTraffic(), helps identify potential RAT activity. When a system attack is confirmed, vulnerability scanning via ScanForVulnerabilities() and dynamic analysis using PerformDynamicAnalysis() are essential steps. Following this, the incident response phase kicks in, with GenerateAlerts() and ImplementRemediationActions() at the forefront. The main objective is to swiftly respond to and mitigate the effects of the detected RAT attack, securing the system against unauthorized remote access.

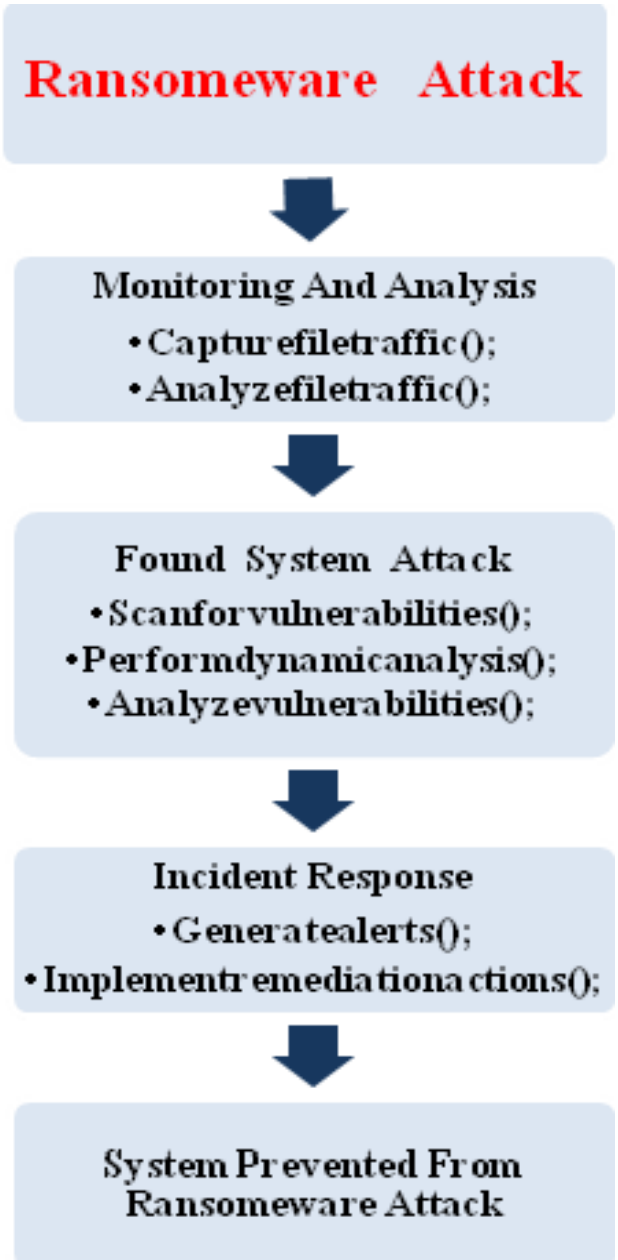

Figure 7. Action taken while Ransomeware attack

It's important to note that the effectiveness of an MP&D framework depends on its proper implementation, regular updates, and the integration of multiple layers of security controls. Additionally, maintaining awareness

of emerging threats and staying up-to-date with the latest security technologies and best practices is crucial to ensuring ongoing protection against evolving malware threats.

VI. Conclusion

In conclusion, the proposed malware prevention and detection (MP&D) framework presents a comprehensive approach to safeguarding systems against malware threats in cloud environments. The framework acts as a funnel, filtering executable files and halting various types of malware through systematic processes and techniques. It encompasses steps such as system initialization, continuous monitoring and analysis, file and network traffic analysis, static and dynamic malware analysis, vulnerability assessment, threat intelligence integration, behavioral analysis, event correlation, incident response, and continuous improvement.

Each step is meticulously designed to detect and respond to potential malware incidents effectively. For instance, continuous monitoring and analysis enable real-time detection of suspicious activities, while vulnerability assessment identifies system weaknesses vulnerable to exploitation. Integration with threat intelligence sources enhances the framework's ability to identify known malware threats. Additionally, behavioral analysis helps detect anomalous behavior indicative of potential malware infections.

In the event of confirmed malware incidents, the framework swiftly initiates incident response procedures, generating alerts and implementing remediation actions to contain and mitigate the impact of the threats. Whether facing ransomware attacks or remote access Trojan (RAT) infiltrations, the framework provides tailored responses to mitigate the damage and secure the system against unauthorized access.

Furthermore, the effectiveness of the MP&D framework hinges on its proper implementation, regular updates, and integration with multiple layers of security controls. Additionally, staying abreast of emerging threats and adopting the latest security technologies and best practices is paramount to ensuring continued protection against evolving malware threats.

The proposed MP&D framework serves as a robust defense mechanism against the ever-evolving landscape of malware threats, offering organizations a proactive approach to safeguarding their cloud environments and mitigating potential risks effectively. By implementing this framework, organizations can enhance their cybersecurity posture and bolster their resilience against sophisticated malware attacks.

References

[1] Mell, P., & Grance, T. (2009), The NIST Definition of Cloud Computing, from NIST Information Technology Laboratory, http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf, retrieved on may 2011.
[2] Ross A. Lumley (2010), Cyber Security and Privacy in Cloud Computing: Multidisciplinary Research Problems in Business, the George Washington University, Report GW-CSPRI-2010-4, December, pp. 1-10.
[3] L. Arockiam, S. Monikandan, G.Parthasarathy (2011), Cloud Computing: A Survey, International Journal of Internet Computing, ISSN No: 2231 – 6965, Volume-1, Issue-2, pp. 26-33. L.
[4] Arockiam et al Security Framework to Ensure the Confidentiality of Outsourced Data in Public Cloud Storage 1270 |International Journal of Current Engineering and Technology, Vol.4, No.3 (June 2014)
[5] Modi, C., Acha, K.: Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. J. Supercomput. 73(3), 1192–1234 (2017)
[6] National vulnerability database—search and statistics. https://nvd.nist.gov/vuln/data-feeds/ (2017)
[7] Li, S.-H.,Yen,D.C., Chen, S.-C.,Chen, P.S., Lu,W.-H.,Cho, C.-C.:Effects of virtualization on information security. Comput. Stand.Interfaces 42, 1–8 (2015)
[8] Malware statistics & trends report: AV-TEST. AV. (n.d.). https://www.av-test.org/en/statistics/malware/
[9] Patil, R., Modi, C.: Designing an efficient framework for vulnerability assessment and patching (VAP) in virtual environment of cloud computing. J. Supercomput. 75(5), 2862–2889 (2018)
[10] Tahir Alyas, Khalid Alissa, Mohammed Alqahtani, TauqeerFaiz, Suleiman Ali Alsaif, Nadia Tabassum, Hafiz Hasan Naqvi. (2022). Multi-Cloud Integration Security Framework Using Honeypots. Mobile Information Systems. 2022. 2600712. 1-13. https://doi.org/10.1155/2022/2600712
[11] Yunlong, F., & Jie, L. (2024). Incentive approaches for cloud computing: challenges and solutions. Journal of Engineering and Applied Science (Online), 71(1), 51-. https://doi.org/10.1186/s44147-024-00389-8
[12] Marnerides, A.K., Spachos, P., Chatzimisios, P., Mauthe, A.U.:Malware detection in the cloud under ensemble empirical mode decomposition, In: 2015 International Conference on Computing, Networking and Communications (ICNC), pp. 82–88 (2015)
[13] Fattori, A., Lanzi, A., Balzarotti, D., Kirda, E.: Hypervisor-based malware protection with accessminer. Comput. Secur. 52, 33–50 (2015)
[14] Watson, M.R.,Marnerides, A.K.,Mauthe, A., Hutchison, D., et al.: Malware detection in cloud computing infrastructures. IEEE Trans.DependableSecur. Comput. 13(2), 192–205 (2016)
[15] Mishra, P., Pilli, E.S., Varadharajan, V., Tupakula, U.: Securing virtual machines from anomalies using program-behavior analysis in cloud environment. In: 18th International Conference on Data Science and

Systems (DSS), pp 991–998, IEEE (2016)
[16] Xie, X., Wang, W.: Lightweight examination of dll environments in virtual machines to detect malware. In: 4th ACM International Workshop on Security in Cloud Computing, pp. 10–16, ACM (2016)
[17] Kumara, M.A., Jaidhar, C.: Leveraging virtual machine introspection with memory forensics to detect and characterize unknown malware using machine learning techniques at hypervisor. Digit. Investig. 23, 99–123 (2017)
[18] Jia, X., Zhou, G., Huang, Q., Zhang,W., Tian, D.: Findevasion: an effective environment-sensitive malware detection system for the cloud. In: International Conference on Digital Forensics and Cyber Crime, pp. 3–17, Springer (2017)
[19] Mishra, P., Pilli, E.S., Varadharajan, V., Tupakula, U.: Vaed: Vmi assisted evasion detection approach for infrastructure as a service cloud. Concurr. Comput. Pract. Exp. 29(12), e4133 (2017)
[20] Xu, Z., Ray, S., Subramanyan, P., Malik, S.: Malware detection using machine learning based analysis of virtual memory access patterns. In: Proceedings of the conference on design, automation & test in Europe, pp. 169–174, European Design and Automation Association (2017)
[21] Joseph, L., Mukesh, R.: Detection of malware attacks on virtual machines for a self-heal approach in cloud computing using vm snapshots. J. Commun. Softw. Syst. 14(3), 249–257 (2018)
[22] Patil, R., Dudeja, H. & Modi, C. Designing in-VM-assisted lightweight agent-based malware detection framework for securing virtual machines in cloud computing. Int. J. Inf. Secur. 19, 147–162 (2020). https://doi.org/10.1007/s10207-019-00447-w
[23] "Advanced Malware." WatchGuard Technologies, 29 Nov. 2016, www.watchguard.com/wgrd-solutions/security-threats/advanced-malware.
[24] Menéndez, Héctor D., et al. "Getting Ahead of the Arms Race: Hothousing the Coevolution of VirusTotal With a Packer." Entropy, vol. 23, no. 4, MDPI AG, Mar. 2021, p. 395. Crossref, https://doi.org/10.3390/e23040395.
[25] Erocarrera. "GitHub - Erocarrera/Pefile: Pefile Is a Python Module to Read and Work With PE (Portable Executable) Files." GitHub, github.com/erocarrera/pefile.git.
[26] Ransomeware Affected File Extension .https://www.fileextensions.org/search/extensions/search/Ransomware/is_true_search/1/sortBy/extension/order/asc/page/1
[27] Michael Miller (2010), Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, QueKnowledge and Grids, pp. 105-112.
[28] Minqi Zhou et al. (2008), Security and Privacy in Cloud Computing: A Survey, In Proceedings of Sixth International Conference on Semantics, Publications, First Printing, Augutst, pp. 149-150.
[29] Damiani E., De Capitani di Vimercati S., Foresti S., Jajodia S., Paraboschi S., and Samarati P. (2005), Metadata management in outsourced encrypted databases, Springer-Verlag, Lecture Notes in Computer Science, In Proceedings of the 2nd VLDB Workshop on Secure Data Management, Trondheim, Norway, September, pp. 1-17.
[30] Fatima Trindade Neves, Fernando Cruz Marta, Ana Maria Ramalho Correia, Miguel de Castro Neto (2011), The Adoption of Cloud Computing by SMEs: Identifying and Coping with External Factors, 11ª Conferência da Associação Portuguesa de Sistemas de Informação, October, pp. 1-11.
[31] Atiq ur Rehman, M.Hussain (2011), Efficient Cloud Data Confidentiality for DaaS, International Journal of Advanced Science and Technology, Vol. 35, October, pp. 1-10.
[32] Richard Chow, Philippe Golle, Markus Jakobsson, Ryusuke Masuoka, Jesus Molina ,Elaine Shi, Jessica Staddon (2009), Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control, In Proceedings of the ACM workshop on Cloud computing security, November, pp. 85-90.
[33] Jyun-Yao Huang, I-En Liao (2012), A Searchable Encryption Scheme for Outsourcing Cloud Storage, COMNETSAT '12, IEEE, pp. 142-146.
[34] Kamara S. and Papamanthou P. (2006), Parallel and Dynamic Searchable Symmetric Encryption, In Proceedings of the 13th ACM conference on Computer and communications security, ACM, NY, USA, October, pp. 79 – 88.
[35] Prabhu Kavin, B., Ganapathy, S., Kanimozhi, U. and Kannan, A., 2020. An Enhanced Security Framework for Secured Data Storage and Communications in Cloud Using ECC, Access Control and LDSA. Wireless Personal Communications, 115(2), pp.1107-1135.
[36] Tabrizchi, H. and Kuchaki Rafsanjani, M., 2020. A survey on security challenges in cloud computing: issues, threats, and solutions. The Journal of Supercomputing, 76(12), pp.9493-9532.
[37] Patil, R., Dudeja, H. and Modi, C., 2019. Designing in-VM-assisted lightweight agent-based malware detection framework for securing virtual machines in cloud computing. International Journal of Information Security, 19(2), pp.147-162.
[38] Kumar PR, Raj PH, Jelciana P (2018) Exploring data security issues and solutions in cloud computing.Proc Comput Sci 125:691–697

[39] N. Naik, P. Jenkins, N. Savage and L. Yang, "Cyberthreat Hunting - Part 2: Tracking Ransomware Threat Actors using Fuzzy Hashing and Fuzzy C-Means Clustering," 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 2019, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2019.8858825.
[40] R. Wilson and I. Iftimie, "Emerging ransomeware threats: An anticipatory ethical anaylsis," 2021 IEEE International Symposium on Technology and Society (ISTAS), Waterloo, ON, Canada, 2021, pp. 1-1, doi: 10.1109/ISTAS52410.2021.9629211.
[41] Singh, J., Refaey, A., & Koilpillai, J. (2020). Adoption of the software-defined perimeter (sdp) architecture for infrastructure as a service. Canadian Journal of Electrical and Computer Engineering, 43(4), 357-363.
[42] Sun, PanJun. "Security and privacy protection in cloud computing: Discussions and challenges." Journal of Network and Computer Applications 160 (2020): 102642.
[43] Kumar, Rakesh, and Rinkaj Goyal. "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey." Computer Science Review 33 (2019): 1-48.
[44] Agarwal Anamika., Verma Satya., Ransomware Anatomy, Impact, and Mitigation Strategies. The journal of Machine Intelligence Research Lab (MIR lab), (2013)
[45] Singh Suyogita., Verma Satya., Federated Learning for GDPR. The journal of Machine Intelligence Research Lab (MIR lab), (2013).
[46] Satya Bhushan Verma., Abhay Kumar Yadav., Detection of Hard Exudates in Retinopathy Images ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal Regular Issue, Vol. 8 N. 4, 41-48 eISSN: 2255-2863 (2019)
[47] Satya B Verma., Shashi B V., Data Transmission in BPEL (Business Process Execution Language), ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal Regular Issue, Vol. 9 N. 3 105-117 eISSN: 2255-2863 (2020).
[48] SB Verma., Brijesh P., and BK Gupta., Containerization and its Architectures: A Study, ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, Vol. 11 N. 4 , 395-409, eISSN: 2255-2863, (2022).
[49] Verma, S. B., & Saravanan., C. Performance analysis of various fusion methods in multimodal biometric. In 2018 International Conference on Computational and Characterization Techniques in Engineering & Sciences (CCTES) (pp. 5–8). IEEE. (2018).
[50] Verma, S.B., Yadav, A.K. Hard Exudates Detection: A Review., Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing, vol 1286. Springer, Singapore (2021).
[51] Huawei Technologies Co., Ltd. (2023). Cloud Computing Technology (1st ed. 2023.). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-3026-3
[52] 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
[53] Ahmad, Waqas, AamirRasool, Abdul RehmanJaved, Thar Baker, ZuneraJalil. (2022). Cyber Security in IoT-Based Cloud Computing: A Comprehensive Survey. Electronics 11(1): 16. https://doi.org/10.3390/electronics11010016