ASTES

*Advances in Science, Technology and Engineering Systems Journal*
*Vol. 8, No. 6, 107-119 (2023)*
www.astesj.com
*Special Issue on Innovation in Computing, Engineering Science & Technology*

**ASTES Journal**
**ISSN: 2415-6698**

# Enhancing Cloud Security: A Comprehensive Framework for Real-Time Detection, Analysis and Cyber Threat Intelligence Sharing

Fazalur Rehman[*,1], Safwan Hashmi[2]

[1]*Department of Cybersecurity, Air University, PAF Complex, E-9, Islamabad, Pakistan*

[2]*Department of Cybersecurity, Riphah International University, Evacuee Trust Complex, F-5, Islamabad, Pakistan*

ARTICLE INFO

*Article history:*
*Received: 30 September, 2023*
*Accepted: 05 December, 2023*
*Online: 30 December, 2023*

*Keywords:*
*Cloud Security*
*Virtual Machine Introspection*
*Cloud Computing*
*Incident Management*
*Cyber Threat Intelligence*
*Incident Response*
*Threat Sharing*
*Cloud Infrastructure Security*

ABSTRACT

*Cloud computing has emerged as a pivotal component of contemporary IT systems, affording organizations the agility and scalability required to meet the ever-changing demands of business. However, this technological evolution has introduced a new era of cybersecurity challenges, as attackers employ increasingly sophisticated strategies to breach cloud networks. Such breaches can have far-reaching consequences, including data loss, financial repercussions, reputational damage, and legal liabilities. In response to these challenges, developing a robust security framework is imperative for effectively safeguarding cloud infrastructure. This paper proposes a novel Hypervisor-based Virtual Machine Introspection (HVMI) for real-time detection and runtime forensic analysis of cyberattacks targeting cloud platforms. The framework proposed comprises several essential components, including a forensic application empowered by Virtual Machine Introspection (VMI) for real-time memory analysis, a centralized Cloud Forensic Tool (CFT) portal for streamlined incident management, and a data transmission and integration web service. Notably, this framework is founded upon a commitment to continuous optimization and enhancement. This iterative process is facilitated through a collaboration approach. It involves fine-tuning various aspects of the framework, such as adjusting settings for VMI, refining criteria for classifying incidents, and updating security controls. Enhancing the forensic application represents a proactive measure aimed at improving the efficiency and effectiveness of VMI and forensic analysis capabilities. The iterative refinement process integrates incident analysis, threat intelligence infusion, and collaborative efforts to adapt to emerging challenges. This dynamic approach fosters a flexible security posture capable of detecting, analyzing, and responding to emerging attacks within cloud platforms. In summary, the proposed framework embodies a comprehensive approach to cloud security, integrating advanced technology with continuous refinement to protect cloud infrastructure, mitigate risks, and navigate the ever-evolving cybersecurity threat landscape effectively.*

## 1 Introduction

Cloud computing has emerged as a fundamental element within modern computing systems, providing organizations with the capacity for on-demand resources and adaptable scalability. It is the on-demand delivery of IT resources. Organizations instead of buying or owning their physical data centers and servers depend on cloud service providers. Securing IT infrastructure is a major challenge nowadays. Within the domain of cloud computing, cybersecurity is a critical concern, given that data breaches have far-reaching impacts on organizations. This paper presents a novel technique and framework based on virtual machine introspection to detect and perform runtime forensic analysis of attacks on cloud platforms. This paper is an extension of the work originally presented at the IEEE 3rd International Conference on Artificial Intelligence (ICAI) 2023 [1].

With the expansion of cloud infrastructure adoption, there has been a corresponding amplification in the array of techniques and approaches employed by malicious entities to initiate attacks targeting these network environments [2]. Such attacks present a substantial threat to the confidentiality, integrity, and availability of cloud-based systems, and can have significant consequences for organizations depending on these systems for their operational functions [3]. These breaches can lead to the exposure of sensitive information, financial

[*]Corresponding Author: Fazalur Rehman, Air University, PAF Complex, E-9, Islamabad, Pakistan, 181219@students.au.edu.pk

losses, harm to an organization's reputation, and legal consequences [4]. To mitigate these risks, organizations need to implement effective cybersecurity measures to ensure the security of their cloud infrastructure [5].

Conventional methods employed for securing cloud infrastructures, such as virtual-level segregation, intrusion detection prevention systems (IDS/IPS), cloud access and security brokers (CASB), and endpoint detection & response, frequently prove inadequate in countering the progressively sophisticated techniques deployed by attackers targeting cloud networks [6]. Even when efforts are made to analyze or thwart these attacks, attackers often exhibit adaptability, which enables them to resist detection and mitigation. Furthermore, these protective measures often operate within virtualized environments shared across interconnected networks, rendering them susceptible to deceptive tactics, insider threats, and network-level attacks [7, 8].

In the event of a security breach, the proposed HVMI solution notifies the cloud service provider while concurrently initiating a forensic analysis to identify the root cause and extent of the breach. This real-time detection and analysis capability inherent in the HVMI tool represents a pivotal enhancement to cloud system security, offering substantial protection against the adverse consequences stemming from data breaches. Furthermore, by incorporating this solution within a web service framework, it attains cross-platform compatibility, irrespective of the underlying hardware and software infrastructure. In a bid to maximize the utility of this research, we have conceptualized a web portal where instances of attack patterns can be uploaded. This portal serves as a means to disseminate information to security organizations and cloud service providers globally, effectively notifying them of prevailing cloud-based attack trends and facilitating the formulation of defensive strategies against such threats. This research endeavor not only contributes to the collective knowledge in the field but also holds practical significance for industry stakeholders, security professionals, and cloud service providers. It offers a novel and real-time approach to detect and analyze attacks within cloud environments, with the overall objective of minimizing their adverse impact on the confidentiality, integrity, and availability of cloud systems.

### *1.1 Research Contributions*

1. The HVMI solution comprises a client application, specifically a forensic application, that operates on the cloud service provider's host. Its primary function is to identify and mitigate the impact of security breaches, making it a valuable resource for organizations seeking to shield themselves from the cyberattacks.

2. Within this framework, malicious artifacts are systematically gathered from virtual machines (VMs), yielding critical insights into the techniques and methods employed by attackers. These artifacts are subsequently transformed into a comprehensible, organized, and shareable format. HVMI adopts the Structured Threat Information eXpression (STIX) standard [9] to generate consistent threat details. These standardized threat details are invaluable to security organizations as they facilitate the development of defensive strategies against specific types of cyberattacks in the cloud, thereby enhancing the overall security posture of cloud systems.

3. The integration of a web service framework ensures cross-platform compatibility, rendering the HVMI solution independent of the underlying hardware and software. Additionally, the accompanying web portal serves as a dedicated platform for the dissemination of cyber threat intelligence. It caters to the needs of both security organizations and cloud service providers, facilitating the exchange of vital information to bolster cybersecurity efforts in cloud environments.

### *1.2 Organization of the paper*

The paper is organized as: section 2 provides literature review. Section 3 provides details of the proposed framework. Section 4 provides details of results, and attack simulations. Section 5 is conclusion & future directions.

## 2 Review of the Literature

This section offers a comprehensive summary of the existing research within the selected domain. It serves to elucidate the present state of knowledge regarding the subject matter, encapsulating the key findings and insights derived from prior studies. Furthermore, it identifies gaps in existing research or areas where current research falls short, signifying opportunities for further investigation and research.

Virtualization stands as a pivotal element within the IT industry, greatly augmenting management capabilities and unlocking the full potential of hardware infrastructure. In essence, Virtualization provides the capability to the impression of multiple physical systems of a single system. It provides a virtual version of the underlying hardware platform, storage media, network devices etc. Virtualization has numerous advantages over traditional physical system architectures, encompassing cost-efficiency, reduced hardware resource requirements, and optimal utilization of available hardware resources. It enables the utilization of hardware to its maximum potential, a feat unattainable within the confines of conventional infrastructure [10]. A hypervisor, also referred to as a virtual machine manager (VMM), is a software application designed to facilitate the creation and administration of virtual machines (VMs) on a physical host. These hypervisors introduce a layer of abstraction between the physical hardware and the virtual machines, enabling multiple VMs to efficiently utilize the resources of a single physical machine. Hypervisors are of paramount importance in cloud computing settings, as they assume a central role in the establishment and supervision of numerous virtual machines (VMs) on a single physical host. This functionality endows organizations with the flexibility to dynamically adjust their computational resources in tandem with fluctuations in demand, thereby optimizing resource allocation in the cloud environment. There are two primary categories of hypervisors:

**Type I (or native or bare metal) hypervisors:** These hypervisors operate directly on the host's hardware, creating a virtualization layer that interfaces between the hardware and the operating sys-

tem [11]. Prominent examples of Type 1 hypervisors encompass VMware ESXi, KVM, Microsoft Hyper-V, and Xen.

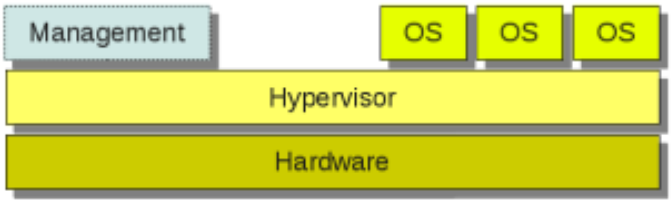


Figure 1: VMM Type I [12]

**Type II (or hosted) hypervisors:** These hypervisors run atop a host operating system, delivering virtualization capabilities for guest operating systems. Well-known instances of Type 2 hypervisors include VMware and VirtualBox.

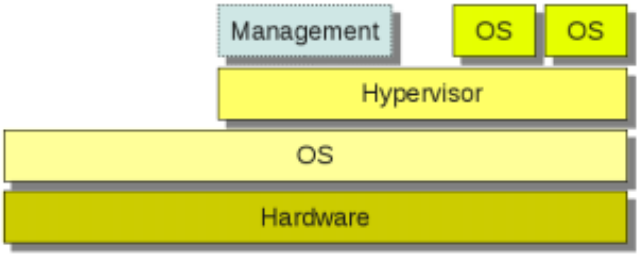


Figure 2: VMM Type II [12]

There has been a growing interest in the adoption of hypervisor-based introspection techniques for the identification and analysis of security breaches within cloud platforms. Notable contributions in this field include the work of Brian [13] have developed a toolbox known as Virtual Contemplation for Xen (VIX). This toolbox has demonstrated its capacity to conduct live examinations and gather volatile data from virtual machines. Their approach involves pausing the virtual machine, extracting the necessary data, and then resuming the virtual machine. In a similar vein, another toolset akin to VIX is XenAccess, created and maintained in [14]. XenAccess is designed to offer a library of functions for constructing a monitoring architecture. This includes employing virtual memory introspection to monitor applications and access the memory state of the virtual machine. This architecture has been leveraged to implement Virtual Machine Introspection (VMI), enabling forensic investigations within virtual machine environments.

In [15], the author introduced Memory Forensics Analysis (MFA). They achieved this by deploying an in-guest agent within the virtual machine (VM) to collect detailed information regarding the processes running on that VM. In a similar vein [16], the author proposed a comprehensive security framework that encompasses both a network-based intrusion detection system (NIDS) and a virtual machine introspection-based intrusion detection system (VMIIDS). The NIDS acts as the initial line of defense, monitoring network traffic before it reaches the virtualization layer. Meanwhile, the VMI-IDS operates at the hypervisor level, where it focuses on detecting intrusions occurring within the VMs themselves. This dual-layered approach enhances the security posture of cloud environments by addressing threats at different levels of the infrastructure.

In [17], the author introduced an anomaly-based detection system designed for the detection of malware at the hypervisor level. Nonetheless, the effectiveness of this approach is circumscribed, and it may not prove sufficiently robust against highly sophisticated and complex attacks. In [18], the author advocated for the utilization of hypervisor-based introspection to identify malware by analyzing information gathered from guest machines and network-level devices. Although this approach achieved a detection accuracy of approximately 90 percent under optimal conditions, its performance declined when confronted with more complicated and advanced attack vectors.

In [19], the author introduced an innovative hardware-based monitoring system named HyperMon, designed for the detection and surveillance of attacks within cloud platforms. HyperMon was implemented within the Xen hypervisor and used machine learning techniques to inspect low-level hardware data at the virtual machine manager (VMM) layer. This process was employed to construct statistical models for programs. In [20], the author put forward a Virtual Machine Introspection (VMI) approach, focusing on anomaly-based detection of keyloggers. This approach accomplished by tracking a spectrum of events, including memory reads and writes, interrupts, and network logs. Subsequently, the collected data underwent analysis employing an artificial immune system (AIS)-based intrusion detection system (IDS) to identify anomalies. However, it's noteworthy that this approach has limitations as it is primarily applicable to Linux-based systems and virtual machines.

In [21], the author proposed a security framework for cloud environments, centered around Virtual Machine Introspection (VMI). This framework is geared towards monitoring the active processes within virtual machines (VMs) by tracking system call traces. The method employs Nitro, a hardware-based tool, to capture these system call traces, which are subsequently relayed to a centralized analyzer. This analyzer utilizes a classification model to distinguish between legitimate and malicious processes. If a process is classified as malicious, an alert is generated. In alignment with the notion that security and monitoring should be implemented at the hypervisor level. In [22], the author proposed a malware analysis technique employing VMI. Their research, however, predominantly focuses on the domain of malware analysis, virtualization techniques, and specific malware families.

The existing solutions discussed in the literature review, while effective at detecting certain types of attacks like rootkits, keyloggers, or malware, exhibit limitations in their capacity to defend against emerging cybersecurity threats. These limitations arise due to their inability to comprehensively address the evolving tactics, techniques, and procedures employed by sophisticated attackers. Furthermore, many of these solutions lack a specific focus on the unique challenges posed by the cloud environment, thus constraining their effectiveness in safeguarding against cloud-based attacks. Traditional methods for analyzing virtual machine (VM) memory and conducting forensic investigations are becoming increasingly very complicated and resource-intensive, rendering them impractical for everyday use. There is a pressing need for an automated solution capable of handling the concurrent security of numerous

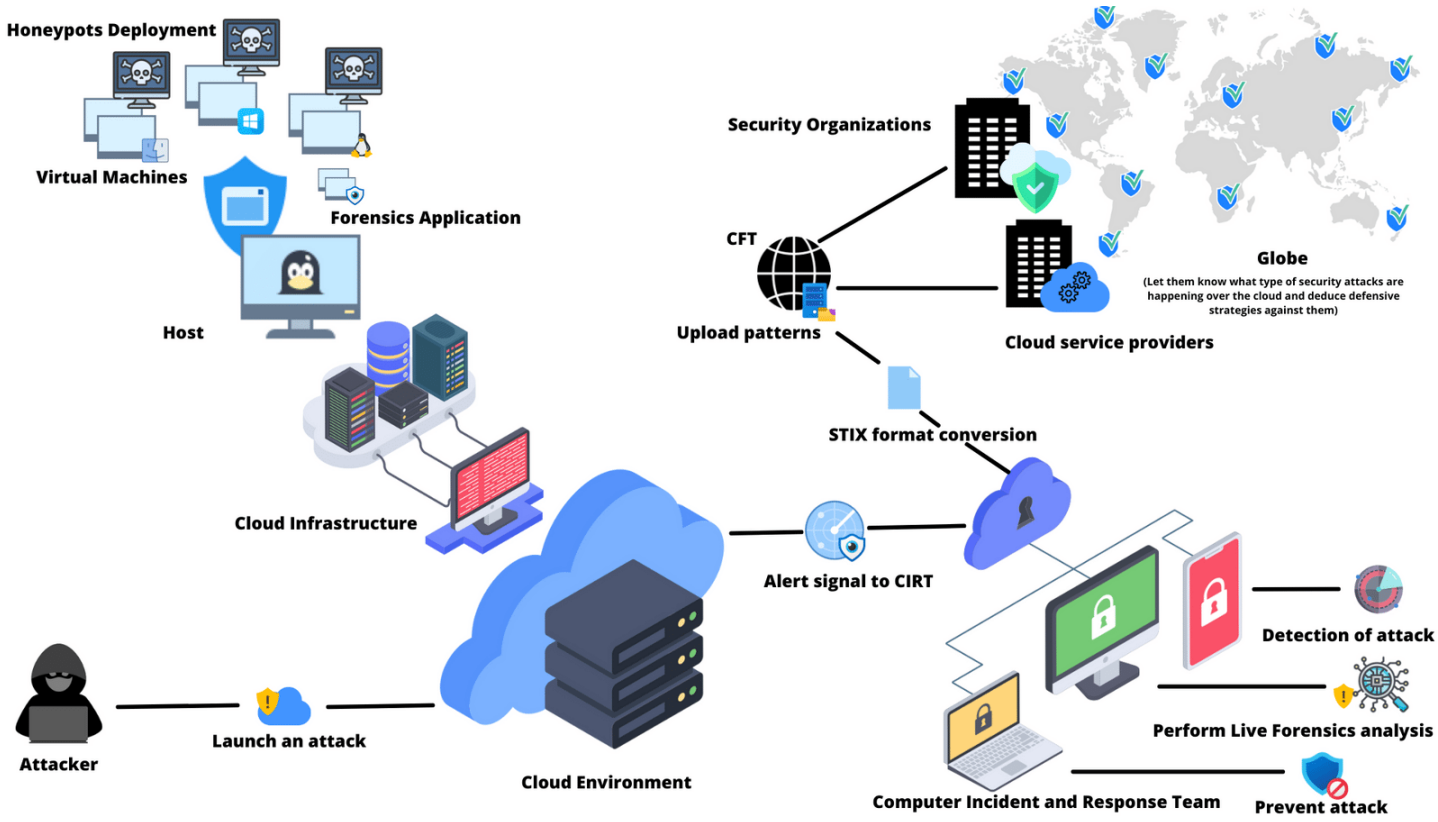


Figure 3: Provides a comprehensive overview of the proposed framework.

VMs, effectively guarding against a broad spectrum of emerging cyber threats, and possessing the scalability required to meet the demands of extensive cloud environments.

## 3 Proposed Framework: Design and Architecture

Our proposed framework for detecting and analyzing attacks on cloud platforms adopts a proactive monitoring approach encompassing both the host and virtual machine (VM) levels. Within this framework, a dedicated forensic application is deployed to inspect VM memory for valuable artifacts and to execute forensic analysis in response to potential attacks on VMs. In the event of a security breach, the system generates alerts, which are promptly disseminated to a Computer Incident Response Team (CIRT) portal. At the CIRT portal, an in-depth forensic analysis is initiated to identify the specific tactics, techniques, and procedures employed in the attack. The findings from this analysis are subsequently translated into a structured threat information expression (STIX) format, enabling their integration into the realm of cyber threat intelligence. Importantly, this framework emphasizes rapid incident reporting, ensuring that security incidents are communicated as soon as they are detected and thoroughly analyzed.

The CIRT possesses access to an extensive knowledge repository comprising analytical reports, procedural guidelines, service records, security logs, and applications installed within the targeted virtual machine (VM). This repository serves as a pivotal resource for identifying any indicators of malicious activity during forensic analysis. Furthermore, the forensic application integrated into the framework exhibits continuous operational verification by dispatching SYN (synchronize) messages to the user interface, thereby confirming its operational status and ensuring the continued activity of the VM. This application also generates diverse types of alerts that are duly documented within the activity log featured on the dashboard. The underlying objective of our proposed framework is to deliver a holistic approach for the detection and analysis of security breaches within cloud platforms. This entails facilitating the exchange of threat intelligence, enriching the knowledge base, and generating alerts, all while maintaining cross-platform compatibility—a facet often absent in traditional solutions. The fundamental aim is to mitigate the repercussions of security incidents and empower the formulation of effective defensive strategies [23].

Fig. 3 offers a comprehensive schematic representation of our proposed framework, providing a holistic view of its operational dynamics. The diagram illustrates the combination of (VMs) operating within the host machine, located in the upper left corner. Concurrently, it offers an illustrative depiction of an attack scenario orchestrated by an aggressor. On the right-hand side of the diagram, the visualized attack scenario seamlessly transitions into the stages of detection, forensic analysis, and the subsequent generation of attack patterns. Notably, these attack patterns are made accessible to the public domain, facilitating proactive measures for defense against forthcoming cyber threats. This visual representation encapsulates the essence of our proposed framework's operation clearly and concisely, offering a cohesive understanding of its functionali-

ties.

In the realm of cybersecurity, the acquisition and utilization of information hold paramount importance. The contemporary surge in cyberattacks has given rise to an encyclopedia of data related to these attacks, effectively constituting a comprehensive repository of information-gathering assaults. Safeguarding systems and IT infrastructure necessitates a two-fold approach. Firstly, a thorough understanding of system vulnerabilities and weaknesses is imperative. Simultaneously, it is crucial to possess insights into the tactics and techniques employed by malicious actors to conduct attacks. This dual comprehension forms the base for making informed decisions geared towards fortifying system defenses. Acknowledging the critical need for robust information exchange within the cybersecurity community, a standardized language has been devised to share threat indicators. This standardization provides flexible and effective dissemination of vital information among cybersecurity stakeholders. In tandem, it complements the proactive stance of gathering comprehensive knowledge about cybersecurity threats, thereby fortifying our capacity to defend against adversarial actions targeted at IT and digital infrastructure. Facilitating this information sharing and knowledge enrichment is done by the Structured Threat Information eXpression (STIX). STIX is an industry-standard programming initiative, developed from the collaborative efforts of multiple organizations, tailored to cater to the need of cyber threat intelligence. This standardized framework is designed to enable the most effective, flexible, and uniform exchange of information in the realm of cybersecurity. The data sharing encompassed within this framework covers threat indicators, incident reports, cyberattack campaigns, profiles of threat actors, courses of action, cyber observables, adversary Tactics, Techniques, and Procedures (TTPs), and exploited targets.

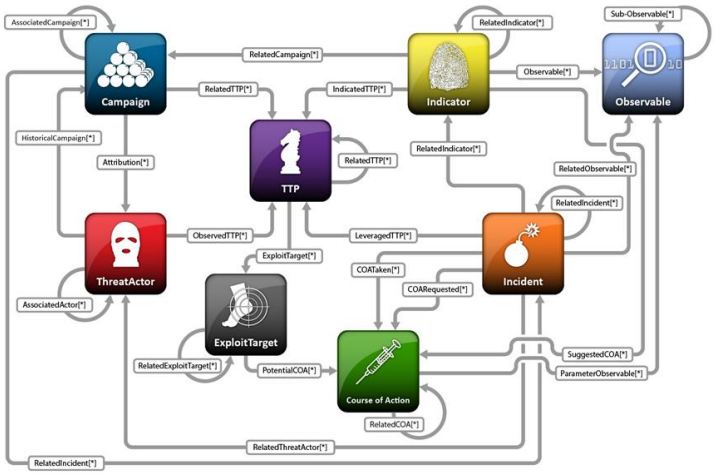


Figure 4: Figure 4 outlines STIX's architecture [24] for human-machine cyber threat understanding.

Fig. 4 provides a brief and basic overview of STIX's architecture consisting of cyber threat information which enables human and machine understandable format. Furthermore, our proactive approach extends to the development of a web portal that invites collaborative contributions from Computer Incident Response Teams (CIRTs), cloud service providers, and security organizations. This collaborative endeavor is predicated on the principle that sharing knowledge and expertise is instrumental in enhancing cybersecurity measures. In the aftermath of successfully mitigating an attack, the resulting attack patterns are transmuted into the STIX format and uploaded onto this web portal. This affirms the pivotal role played by STIX in the domain of cyber threat intelligence, as an indispensable tool for strengthening cybersecurity measures and fostering collective resilience against evolving threats.

Within this solution, we have strategically integrated the NIST 800-53 security control families, with particular emphasis on the following key elements: **Audit and Accountability (3.3):** This component is vital for maintaining comprehensive records of system activity to facilitate effective monitoring and analysis. **Assessment, Authorization, and Monitoring (3.4):** This aspect encompasses the critical phases of assessing, authorizing, and continually monitoring system security. **Vulnerability Monitoring and Scanning (3.16 RA-5):** The solution places significant focus on continuously monitoring and scanning for vulnerabilities, ensuring proactive threat mitigation. **Technical Surveillance Countermeasures Survey (3.16 RA-6):** This control addresses the imperative task of conducting technical surveillance countermeasures surveys to safeguard against potential threats. **Incident Response (3.8):** Incident response is a core facet, with particular emphasis on IR-4, which encompasses incident handling. This includes detection, analysis, containment, and eradication of security incidents. These NIST 800-53 control families are integral to the comprehensive complete solution, contributing to its robust security posture by addressing various aspects of security control, monitoring, and incident response, thereby fortifying the overall cybersecurity framework.

### *3.1 Deployment and Simulation*

The proposed framework for the detection and analysis of attacks on cloud platforms is adaptable to diverse cloud environments. Its flexibility is exemplified by the ability to deploy the forensic application both on the host and within virtual machines (VMs). Furthermore, the CIRT portal is designed to be accessible remotely, permitting authorized users to engage with it from a distance. To enhance the efficacy of the proposed framework, rigorous testing can be conducted through the execution of simulated attack scenarios encompassing various attack types and security contexts. The outcomes derived from these simulations serve as a means to comprehensively assess the framework's performance. Subsequently, any identified areas for potential enhancement or improvement can be systematically addressed, ensuring that the framework attains optimal effectiveness in real-world deployments. This iterative process of testing and refinement contributes to the continual evolution and robustness of the framework, aligning it with the dynamic landscape of cybersecurity in cloud environments.

To illustrate, the forensic application within the framework is amenable to configuration for the emulation of diverse attack types, encompassing scenarios involving malware, rootkits, and keyloggers. Subsequently, the CIRT can conduct an in-depth analysis of the data generated by these simulated attacks to assess the framework's efficacy in both detection and analysis. The outcomes of these simulations serve a dual purpose. Firstly, they enable the fine-tuning of the forensic application's configuration and the optimization of

the CIRT portal's functionality, enhancing their performance in the context of attack detection and analysis. Secondly, the simulation results can illuminate potential vulnerabilities or deficiencies within the framework, thereby providing critical insights for further enhancement. By utilizing the power of simulations, the proposed framework stands to undergo iterative refinement, resulting in an elevated level of effectiveness in the detection and analysis of attacks within cloud platforms. This iterative process, rooted in empirical testing and analysis, lends itself to the continuous improvement of the framework's capabilities, ensuring its relevance and robustness in an ever-evolving cybersecurity landscape within the cloud. To facilitate comprehension, an illustrative workflow is presented in Fig. 5, providing a simplified depiction of the sequential steps inherent to the outlined rational approach. This provides a clear and accessible representation of the procedural sequence.

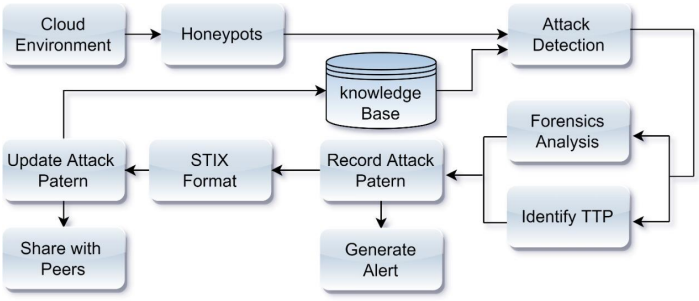


Figure 5: A visual representation of the attack detection workflow process.

The establishment of a cloud system entails the configuration of multiple virtual machines (VMs), each equipped with distinct honeypots designed to entice potential attackers into initiating attacks. Upon activation of these honeypots, they initiate a signaling process that notifies the Computer Incident Response Team via the dedicated CIRT portal. This orchestration allows the framework to conduct forensic analysis of the attack stealthily, rendering the attacker unaware of the ongoing detection, monitoring, and analysis. This stealthy approach is instrumental in uncovering the Tactics, Techniques, and Procedures (TTP) deployed by the attacker during the course of the attack. By surreptitiously observing and dissecting the attack, the framework can glean critical insights into the attacker's modus operandi, thereby enhancing our understanding of cyber threats and fortifying our defenses. Detailed information regarding the subcomponents, essential features, and functional domains is elaborated upon in the subsequent sections.

### 3.1.1 *Forensic application*

The core functionality of the forensic application encompasses the critical task of executing virtual machine introspection (VMI) and subsequently transmitting the acquired data to the web service [25]. It is important to highlight that the forensic application is designed to function flawlessly on both the host and the virtual machines (VMs), highlighting its critical function in enhancing cloud infrastructure security against persistent cyber threats. To ensure the distinct identity and traceability of each instance of the forensic application, a unique identification (ID) is assigned to every running instance, whether on the host or VMs. This rigorous identification system serves as a base for precise command execution during VMI operations. Notably, the framework's inherent cross-platform compatibility enables the seamless implementation of updates and patches for the forensic application, thereby ensuring its continual alignment with evolving security requirements. It is crucial to keep the forensic application hidden from possible attackers. It is safeguarded through an array of comprehensive countermeasures that are meticulously enforced to preserve the application's integrity. Furthermore, the forensic application, seamlessly integrated within the framework, maintains continuous operational verification. This is achieved through the systematic dispatch of SYN messages to the user interface, thereby affirming its operational status and ensuring the sustained functionality of the associated VM.

### 3.1.2 *Web services*

The web service plays a pivotal role within the architecture, serving as a vital intermediary between the forensic application and a centralized database repository. This repository houses an extensive range of data, which is subsequently made accessible through a dedicated cloud forensic tool (CFT) or portal. The primary function of the web service revolves around the seamless transmission of data from the forensic application to the database, a process executed in real-time upon data reception. The data collected by the forensic application is rich and multifaceted, encompassing crucial system information such as operating system details, processor specifications, core configurations, RAM allocations, storage characteristics (both hard drive and solid-state drive attributes), serial identification markers, geographical location data, active processes, system services, security logs, and a comprehensive list of installed applications. Before integration into the database, this collected data undergoes rigorous parsing procedures to ensure its uniformity and compatibility. Once parsed, the data is methodically inserted into the database, subsequently becoming readily accessible through the designated portal. The incorporation of the web service stands as a fundamental component within this framework, endowing the proposed solution with cross-platform compatibility. This strategic architecture ensures that the core infrastructure remains consistent, with adaptations made solely to the forensic application running on the host and VMs to align with the specific programming languages and hardware of the underlying systems. Consequently, the framework boasts versatility, capable of accommodating diverse operating environments ranging from UNIX, Linux, MAC, and IOS, to Android, among others. This cross-platform compatibility underscores the framework's adaptability and applicability across a wide spectrum of technology ecosystems.

### 3.1.3 *Cloud Forensic Tool/Portal (CFT)*

Cloud Forensic Tool (CFT) is a special portal designed to make it easier for Cloud Service Providers (CSPs) to interact with the framework. Through this site, CSPs can easily register both their host machines and virtual machines (VMs). Furthermore, the CFT operates as a centralized hub for receiving real-time security breach alerts, a critical function that underpins the framework's proactive stance in cybersecurity. Integral to the CFT's functionality is its role as the platform for orchestrating Virtual Machine Introspec-

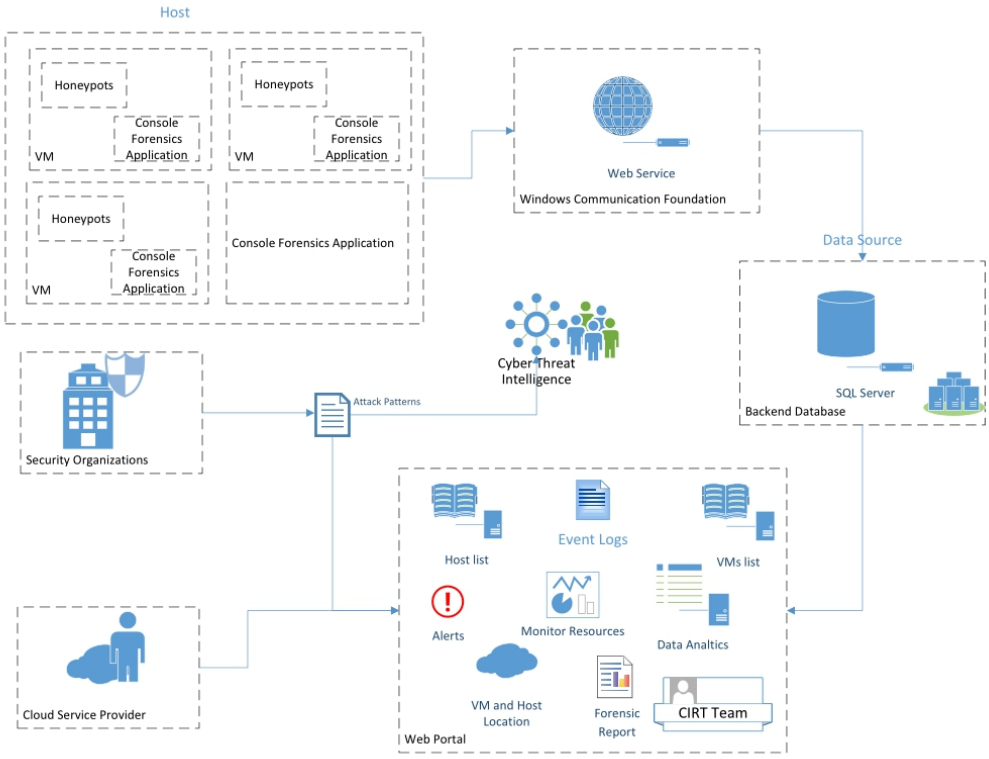


Figure 6: Provides a complete system architecture overview of the proposed framework.

tion (VMI). This capability empowers the framework to conduct deep and real-time examinations of VMs, enhancing its capacity to swiftly detect and analyze security breaches. Additionally, the CFT is the medium through which attack patterns are uploaded in a structured threat information expression (STIX) format after an attack has been effectively contained. This mechanism facilitates the dissemination of vital cyber threat intelligence, thereby bolstering the collective knowledge base and enhancing preparedness against future threats. Fig 7 offers a glimpse into the CFT's user interface, providing an insightful single-screen overview encompassing all integral components of the framework. This visual representation serves as an invaluable tool, affording users a consolidated perspective of the framework's operations, enhancing user experience, and streamlining administrative functions.

Figure 7: A dashboard screen snapshot offering a comprehensive overview of CFT components.

The CFT dashboard screen serves as a pivotal control center within the framework, offering a comprehensive array of critical information and functionalities. It serves as a dynamic repository of indispensable data, including reported cybersecurity incidents, real-time threat alerts, updates regarding newly onboarded Cloud Service Provider (CSP) members, resource utilization metrics pertaining to host machines, and geospatial location data pinpointing the whereabouts of both host machines and their hosted virtual machines (VMs). Notably, CSPs are endowed with the versatile capability to access and extract data in various formats, including the option to download data in CSV files or generate hardcopy printouts. This flexibility extends to encompass data retrieval from host machines, VMs, as well as comprehensive logs and process records.

In terms of incident management and response, both CSPs and administrators wield the authority to initiate alert transmissions to the Computer Incident Response Team (CIRT). This role complements the foundational commitment of the framework to prompt and efficient incident response and plays a crucial role in launching forensic investigations. Furthermore, the dashboard's functionality extends to the systematic dispatch of SYN (synchronize) messages from VMs to validate the operational status of the forensic application and affirm the active state of the VM. Together with the various alert types found in the activity log, these messages help to strengthen the dashboard's real-time monitoring and notification features.

Optimizing the framework is a crucial aspect of maintaining a robust security posture in cloud environments. The optimization process involves identifying vulnerabilities or weaknesses in the framework's configuration. This can be achieved through rigorous testing, incident simulations, and real-world incident analysis. The primary goals of optimizing the framework are to enhance its resilience against emerging cyber threats, improve its efficiency in detecting and analyzing attacks, and minimize false positives. During the optimization process, cloud service providers (CSPs) and administrators collaborate to fine-tune various components of the framework. This may include adjusting settings for Virtual Machine Introspection (VMI), refining incident classification criteria, and updating security controls. By continually optimizing the framework, organizations can adapt to evolving threat landscapes and ensure that their cloud infrastructure remains secure and resilient. Enhancing the forensic application is a proactive measure aimed at improving the effectiveness of virtual machine introspection (VMI) and forensic analysis capabilities. Enhancements to the forensic application may encompass several aspects for example performance improvement, feature updates, security enhancement, and cross-platform compatibility. Regular updates and enhancements to the framework help to stay ahead of emerging cyber attacks.

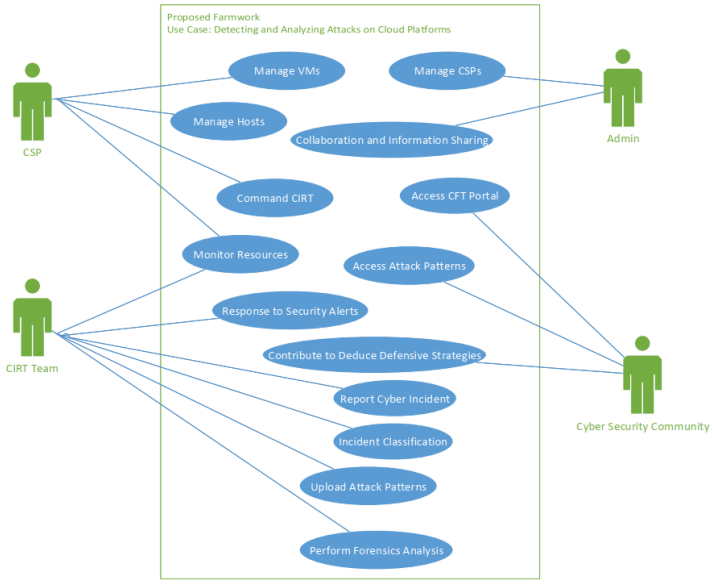


Figure 8: Provides a concise overview of use case modeling.

Fig 8 provides a very brief and precise use case modeling encompassing the proposed framework, and Fig 6 depicts the system architecture implemented within the proposed framework.

# 4 Results

The outcomes derived from the utilization of our proposed Hypervisor-based Virtual Machine Introspection (HVMI) tool exhibited promising results in detecting and analyzing attacks on cloud platforms. Rigorous testing was conducted through a series of simulations to comprehensively evaluate HVMI's efficacy in detecting a spectrum of attack types and dissecting the complexity of their tactics, techniques, and procedures (TTPs).

In the initial simulation, our evaluation focused on assessing HVMI's capability to identify a straightforward brute-force attack directed toward a virtual machine. The findings from this simulation revealed that HVMI exhibited prompt and effective detection of the attack, transmitting timely notifications to the CSP mere seconds after the attack's commencement. Furthermore, HVMI furnished a detailed and comprehensive analysis of the attack's Tactics, Techniques, and Procedures (TTPs). This encompassed an in-depth examination of the attack's nature, its originating source, and the specific system that was the subject of the attack.

Within the scope of the second simulation, our assessment aimed to evaluate HVMI's proficiency in detecting and analyzing a complicated and highly sophisticated malware attack targeted at a virtual machine. The findings from this simulation affirmed HVMI's adeptness in not only promptly identifying the attack but also in conducting a comprehensive analysis of the attack's intricacies. This analysis culminated in the generation of an exhaustive report detailing the Tactics, Techniques, and Procedures (TTPs) employed by the attack. Furthermore, the report delved into the potential ramifications of the attack, particularly in terms of its potential impact on the confidentiality, integrity, and availability of the system. To facilitate a holistic understanding of the entire process, we have included a detailed case study within this context.

## *4.1 Case Study: Launching a reverse TCP attack*

Conducting the case study necessitated the initial setup of a Kali Linux machine, designated as the attacker entity. This machine underwent meticulous configuration to enable the initiation of a reverse TCP attack against a virtual machine (VM) hosted within the cloud infrastructure, ultimately aiming to illicitly establish reverse shell access. For the execution of this attack, the utility was employed to craft a payload tailored to facilitate the invasion into the VM. A pivotal component of this payload was an executable (Exe) file, meticulously constructed to be delivered to the target VM. Upon execution of this payload, a chain of events was set into motion. It prompted the VM to initiate the opening of a reverse shell, thus enabling the attacker to gain remote access through reverse TCP connectivity. Within the payload's configuration, crucial details such as the IP address and port on which the attacker's machine would be listening were meticulously specified. This case study served as a practical exercise to comprehensively assess HVMI's capabilities in detecting and analyzing an incursion of this nature, ultimately yielding valuable insights into the attack's methodologies and potential impact on the targeted system's security and integrity.

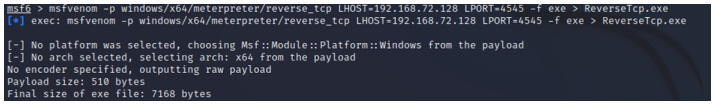


Figure 9: A Snapshot of an Exe payload generation for targeted VM.

Upon configuring the requisite parameters, the exploit was executed. This initiation prompted the attacker's machine to transition into a listening state, awaiting the incoming reverse TCP connection

from the target virtual machine (VM). This allowed an attacker to gain unauthorized access to the victim machine's shell, thereby affording an array of capabilities for potentially malicious activities. Within the scope of this gained access, activities encompassed the potential for privilege escalations and the execution of DLL injection, thereby underscoring the severity and breadth of the security breach. Fig 10 shows a snapshot of the reverse shell from the attack simulation.

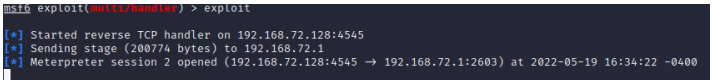


Figure 10: A snapshot of the accessed reverse TCP shell in the attack simulation.

Following the successful execution of the attack, our subsequent actions encompassed the installation of remote access software with the intent of securing persistent access while simultaneously deleting all traces within the event logs to obfuscate our activities and minimize detection. However, as soon as this attack occurred, our forensic application sprang into action. It promptly transmitted a series of alerts to the designated web portal and initiated the critical forensic analysis procedure. Fig 11 provides a comprehensive insight into the detailed real-time alerts generated as part of this incident response mechanism, emphasizing the real-time nature of our detection and analysis capabilities.

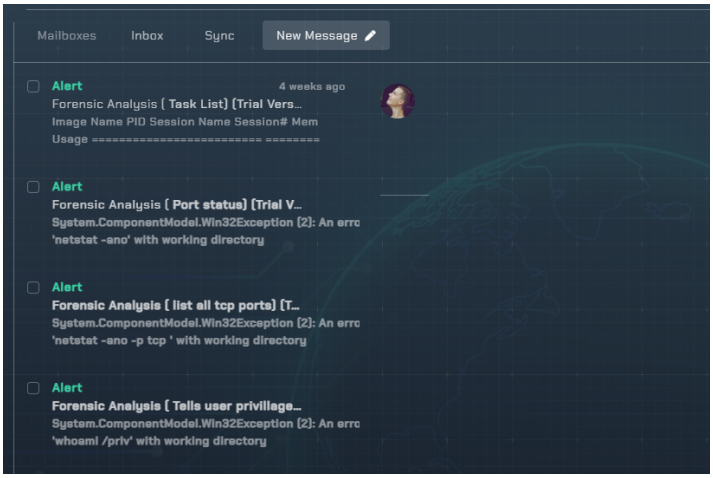


Figure 11: A snapshot of real-time alert prompts displayed on the portal.

This point underscores the pivotal role played by HVMI in resolving these security challenges. The proactive security and monitoring functionalities embodied within HVMI serve as a potent deterrent against such malicious incursions.

Before embarking on the identification of anomalous entities within VM's memory, several key inquiries come to the forefront. These include: Process Legitimacy: Are there any indications of suspicious processes deviating from their expected execution paths? Is the suspicious process running in tandem with its legitimate parent, or does it originate from an unrelated source? Temporal Aspects: What is the temporal profile of the process in question? When did it commence and conclude its execution? What are the implications of its execution timeline, and do they align with the expected behavior? Behavioral Profiling: What behavioral advantages can be attributed to the observed process? Does its behavior conform to the anticipated outcomes, or does it exhibit aberrant characteristics that warrant scrutiny? Process Name Variance: Does the process name exhibit any anomalies? Is it masquerading as a valid Windows process, thus evading detection? Comprehensive Analysis: Beyond process name scrutiny, an exhaustive analysis extends to threads, mutexes, Dynamic Link Libraries (DLLs), process-to-file mappings, Resident Memory (RAM) sections, and any associated sockets and open ports attributed to the process. These attributes are critical in discerning the origin of connections and the entities initiating them, as well as their associated dependencies. Our proposed solution (HVMI) empowers cybersecurity professionals with a potent arsenal of capabilities, enabling a proactive stance in countering, identifying, and comprehensively analyzing security threats.

| HandleCount | Name | Priority | ProcessId | ThreadCount | WorkingSe |
|---|---|---|---|---|---|
| 0 | System Idle Process | 0 | 0 | 8 | 8192 |
| 4139 | System | 8 | 4 | 115 | 57344 |
| 0 | Registry | 8 | 124 | 4 | 79228928 |
| 60 | smss.exe | 11 | 352 | 2 | 573440 |
| 645 | csrss.exe | 13 | 452 | 11 | 2678784 |
| 161 | wininit.exe | 13 | 540 | 3 | 1728512 |
| 169 | csrss.exe | 13 | 548 | 10 | 2392064 |
| 214 | winlogon.exe | 13 | 640 | 3 | 4018176 |
| 644 | services.exe | 9 | 684 | 9 | 8208384 |
| 4304 | lsass.exe | 9 | 704 | 8 | 19484672 |
| 1033 | svchost.exe | 8 | 824 | 11 | 15077376 |
| 33 | fontdrvhost.exe | 8 | 852 | 5 | 1613824 |
| 33 | fontdrvhost.exe | 8 | 860 | 5 | 1523712 |
| 1232 | svchost.exe | 8 | 936 | 10 | 20365312 |
| 1911 | svchost.exe | 8 | 988 | 5 | 9269248 |
| 2881 | svchost.exe | 8 | 500 | 127 | 33820672 |
| 503 | LogonUI.exe | 13 | 440 | 9 | 45416448 |
| 752 | dwm.exe | 13 | 552 | 14 | 21630976 |
| 211 | svchost.exe | 8 | 984 | 1 | 2277376 |
| 340 | svchost.exe | 8 | 1080 | 1 | 4386816 |
| 611 | svchost.exe | 8 | 1152 | 6 | 30240768 |
| 158 | svchost.exe | 8 | 1272 | 3 | 3686400 |
| 138 | svchost.exe | 8 | 1300 | 2 | 3866624 |
| 213 | svchost.exe | 8 | 1364 | 2 | 7499776 |
| 179 | svchost.exe | 8 | 1372 | 3 | 2637824 |

Figure 12: A snapshot of real-time processes running on VM.

The forensic application, as part of its operational protocol, diligently transmitted an array of pertinent information regarding the processes actively running within the targeted virtual machine (VM). This comprehensive data set encompassed critical attributes such as process priority, unique process identification (PID), thread counts, and resource consumption metrics, among other relevant parameters. For more clarity and reference, Fig 12 provides a visual representation of the enumerated processes actively executing within the targeted VM, thereby affording a comprehensive overview of the system's operational state.

| | | | | |
|---|---|---|---|---|
| sethc.exe | 12800 | | 2 | 720 K |
| TabTip.exe | 13360 | | 2 | 9,588 K |
| TabTip32.exe | 14272 | | 2 | 1,376 K |
| svchost.exe | 12740 | Services | 0 | 5,996 K |
| SecurityHealthService.exe | 12572 | Services | 0 | 9,932 K |
| msedge.exe | 2816 | | 2 | 86,100 K |
| msedge.exe | 6444 | | 2 | 9,116 K |
| msedge.exe | 8892 | | 2 | 31,396 K |
| msedge.exe | 5396 | | 2 | 35,784 K |
| msedge.exe | 6552 | | 2 | 19,728 K |
| msedge.exe | 12692 | | 2 | 72,888 K |
| TabTip.exe | 11060 | Console | 1 | 13,500 K |
| msedge.exe | 9140 | | 2 | 134,240 K |
| MusNotifyIcon.exe | 11844 | | 2 | 3,420 K |
| w3wp.exe | 8252 | Services | 0 | 322,612 K |
| Ssms.exe | 6084 | | 2 | 306,216 K |
| WmiPrvSE.exe | 11760 | Services | 0 | 20,896 K |
| sppsvc.exe | 11236 | Services | 0 | 12,688 K |
| devenv.exe | 11052 | | 2 | 1,257,808 K |
| ReverseTcp.exe | 11056 | Services | 2 | 9,924 K |
| PerfWatson2.exe | 8160 | | 2 | 77,848 K |
| Microsoft.ServiceHub.Cont | 10704 | | 2 | 73,008 K |
| ServiceHub.VSDetouredHost | 14648 | | 2 | 101,360 K |
| ServiceHub.IdentityHost.e | 14548 | | 2 | 71,900 K |
| ServiceHub.ThreadedWaitDi | 15692 | | 2 | 80,312 K |
| ServiceHub.RoslynCodeAnal | 700 | | 2 | 434,764 K |
| ServiceHub.Host.CLR.x86.e | 9676 | | 2 | 73,828 K |
| ServiceHub.SettingsHost.e | 14584 | | 2 | 76,404 K |
| ServiceHub.Host.CLR.x64.e | 12860 | | 2 | 827,164 K |
| ServiceHub.Host.CLR.exe | 6148 | | 2 | 117,212 K |
| ServiceHub.TestWindowStor | 14172 | | 2 | 80,300 K |
| Microsoft.Alm.Shared.Remo | 15284 | | 2 | 70,808 K |
| Microsoft.VisualStudio.We | 12324 | | 2 | 90,512 K |
| ServiceHub.DataWarehouseH | 5580 | | 2 | 105,408 K |
| WebViewHost.exe | 13668 | | 2 | 55,956 K |
| conhost.exe | 2940 | | 2 | 11,576 K |
| msedgewebview2.exe | 10308 | | 2 | 103,552 K |
| msedgewebview2.exe | 12492 | | 2 | 8,880 K |
| msedgewebview2.exe | 7936 | | 2 | 45,892 K |
| msedgewebview2.exe | 9396 | | 2 | 29,668 K |
| msedgewebview2.exe | 4420 | | 2 | 19,160 K |
| msedgewebview2.exe | 4344 | | 2 | 68,864 K |
| msedgewebview2.exe | 1088 | | 2 | 52,956 K |
| ServiceHub.Host.CLR.x86.e | 8560 | | 2 | 86,092 K |
| node.exe | 10212 | | 2 | 43,752 K |
| node.exe | 4576 | | 2 | 44,092 K |
| conhost.exe | 5488 | | 2 | 11,744 K |
| conhost.exe | 6980 | | 2 | 11,752 K |
| node.exe | 5384 | | 2 | 38,636 K |
| VsDebugConsole.exe | 8460 | | 2 | 6,200 K |
| conhost.exe | 11936 | | 2 | 22,984 K |

Figure 13: A snapshot of real-time identification of ”ReverseTcp.exe” by the forensic application on the VM.

Attackers employ diverse methodologies for delivering malicious payloads to victim machines. Given our primary emphasis on defensive measures, we have confined our analysis to two prominent attack vectors. The first scenario involves an attacker uploading a payload to a server hosting a service accessible via a specific port. Alternatively, attackers may disseminate the payload by incorporating it into a website or server. In such instances, unsuspecting users who attempt to access the compromised server or website inadvertently trigger the download of the payload. Additionally, attackers may resort to deception by disguising the payload as a Trojan horse. In response to any of these scenarios, the Computer Incident Response Team (CIRT) wields the capability to issue commands to the forensic application, thereby facilitating the retrieval of a comprehensive list of installed applications within the target virtual machine (VM). This scrutiny serves as a crucial component in the quest to identify any suspicious software installations or potential security breaches. Fig 13 provides an inclusive catalog encompassing active services and applications hosted within the VM. This exhaustive compilation of applications is readily accessible through the Computer Incident Response Team (CIRT) portal, thereby offering a streamlined reference point for cybersecurity professionals and facilitating in-depth investigative analysis. Of noteworthy significance within this list is the presence of the ”Reverse TCP Exe” application. This particular application bears significance as it was generated by the attacker with the explicit intent of establishing remote access to the virtual machine, thereby highlighting a critical security concern warranting further investigation and remediation.

Fig 14 provides a brief overview of the activity log within the CIRT portal. This log serves a dual purpose: first, it offers operational verification of both the Virtual Machine (VM) and the forensic applications throughout the simulation, ensuring their continuous activity and functionality. Secondly, it highlights the availability of the forensic analysis report on the portal, presenting a real-time and chronological account of activities and alerts generated during the simulation.


Figure 14: A snapshot of the activity log within the CIRT portal.

The HVMI tool demonstrated its capability to access and retrieve a comprehensive analysis report pertaining to the targeted virtual machine (VM). This encompassed a thorough examination of the VM's operational facets, encompassing active processes, running services, security logs, and the catalog of installed applications. Furthermore, the Cloud Service Provider (CSP) was equipped with the remote accessibility to monitor processes and network connections, allowing for the identification of the processes responsible for initiating and sustaining these connections. At a broader analytical spectrum, the HVMI tool extends two primary reporting avenues, specifically the Forensic Report and the Incident Report. The Forensic Report serves as a document that presents the results of forensic investigation. Fig 15 provides a snapshot of the interface, showcasing the automated forensic reports that have been systematically generated and made accessible through the portal.


Figure 15: A snapshot of available forensic reports on the portal.

Fig 16 provides a visual representation of the logs that have been systematically generated and recorded throughout the entire operational process. These logs are pivotal in facilitating comprehensive documentation of the actions, events, and activities undertaken within the cloud platform, rendering them invaluable for analytical and investigative purposes within the academic and technical domains.

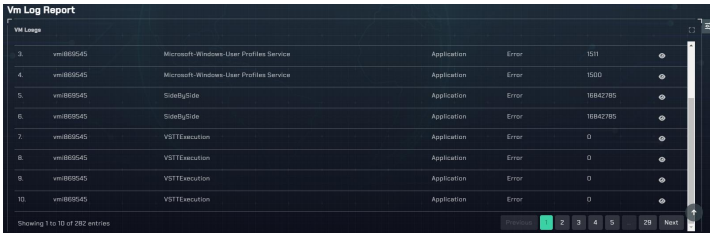


Figure 16: A visualization of VM logs generated during the entire process.

The Computer Incident Response Team (CIRT) portal offers the capability to export event logs in digital format, specifically in Comma-Separated Values (CSV) format for electronic use, or in hardcopy format for physical documentation purposes. This functionality allows for the efficient dissemination and preservation of critical event data for further analysis and reference in the context of incident response and cybersecurity management.

Within the context of cloud service providers (CSPs), the establishment of a robust cyber threat intelligence capability is of paramount importance. A pivotal facet of such a capability lies in the practice of sharing critical information with trusted partners, peers, and relevant stakeholders. This collaborative approach to cyber threat intelligence serves as a vital means of streamlining and comprehending the vast and very complicated landscape of cybersecurity data that CSPs encounter. By engaging in cyber threat intelligence and data sharing endeavors, organizations can effectively distill and prioritize the overwhelming volume of complicated cybersecurity data they encounter. This process involves gaining insight into the vulnerabilities and flaws within their information systems, as well as an understanding of the complicated tactics, techniques, and procedures (TTPs) employed by cyber adversaries. Additionally, it encompasses the identification and analysis of attack patterns. For the purpose of global accessibility and ease of comprehension, the results of cyber incidents, intelligence data, and attack patterns are systematically generated in the standardized Structured Threat Information Expression (STIX) format. This format facilitates the seamless exchange of cyber threat intelligence and enhances the collaborative efforts aimed at fortifying information systems against cyberattacks. Fig 17 provides a sample code snippet that serves as an illustrative for comprehending the visualization of Structured Threat Information Expression (STIX).

```
<stix:Threat_Actors>
    <stix:Threat_Actor id="trojan:threatactor-9b371afe-ddfd-4954-abaf-8abb357ac78e" xsi:type="ta:ThreatActorType" version="1.2">
        <threat-actor:Title>Trojan Horse backdoor</threat-actor:Title>
        <threat-actor:Type>
            <stixCommon:Value>APT</stixCommon:Value>
        </threat-actor:Type>
    </stix:Threat_Actor>
    <threat-actor:Observed_TTPs>
        <threat-actor:Observed_TTP>
            <stixCommon:Relationship>Use</stixCommon:Relationship>
            <stixCommon:TTP idref="trojan:ttp-649b70a0-0158-4cc6-a8d5-cf8a441dc290"/>
        </threat-actor:Observed_TTP>
    </threat-actor:Observed_TTPs>
</stix:Threat_Actors>
```

Figure 17: A sample code for visualization of patterns and syntax within STIX code.

STIX serves as a standardized language designed to convey comprehensive information concerning cyber threats. This encompassing data includes indicators of compromise (IOCs), which are distinctive attributes or artifacts linked to a cyberattack, such as specific IP addresses, domain names, file hashes, and other pertinent information instrumental in identifying or tracing an attack's origins. The strategic collection and analysis of IOCs empower organizations to pinpoint ongoing or past attacks, subsequently enabling them to initiate pertinent countermeasures and proactively deter future assaults. STIX plays a pivotal role in streamlining the aggregation, organization, and dissemination of IOCs, ensuring consistency and structure in the process. Fig 18 provides a visual representation of the relationship between the extracted IOCs during the attack simulation.

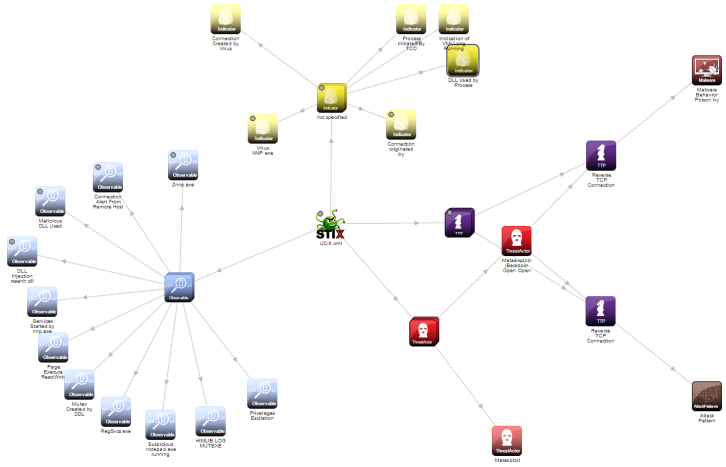


Figure 18: Provides a visual representation of the relation between the extracted IOCs.

In the cloud environment organizations frequently operate within shared infrastructure and resource environments, which inherently pose challenges in terms of tracking and detecting cyberattacks. The integration of STIX and IOCs into their cybersecurity framework equips organizations with a standardized mechanism to enhance their ability to recognize and respond to attacks more efficiently. This is achieved by facilitating the systematic collection and sharing of critical threat-related information and indicators of compromise, ultimately enhancing their cyber defense capabilities.

As an illustration, consider an organization that strategically incorporates STIX and IOCs into its cloud security framework. In this context, the organization leverages various sources of threat intelligence feeds or other pertinent information repositories to proactively identify potential indicators of compromise (IOCs). These IOCs might encompass entities like IP addresses or domains that are linked to previously documented malicious activities. Subsequently, the organization uses this invaluable information to configure its suite of security tools and policies. These configurations serve to either block or trigger alerts in response to these identified IOCs, thereby fortifying their capacity to avert or detect cyberattacks at an early stage, consequently reducing the potential for substantial damage.

Upon the completion of the comprehensive forensic analysis, the prevention of the cyber attack, and the documentation of attack patterns, the subsequent crucial step involves the formal reporting of the incident. An incident report serves as comprehensive documentation encompassing everything about the cyber attack incident that has occurred. These details encompass temporal information such as the precise timing of the incident, location, and a thorough narrative outlining the nature of the occurrence. Additionally, the report encapsulates actions taken in direct response to the incident.

Fig 19 provides a snapshot of the cyber incident report available on the portal.

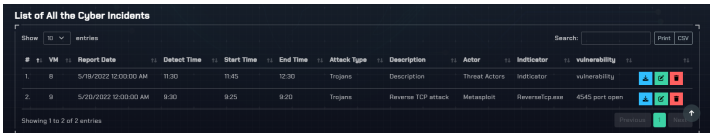


Figure 19: A snapshot of a comprehensive cyber incident report accessible via the portal.

The cyber incident report serves as a repository of critical insights into the security breach, exploring various facets and aspects of the incident from different angles. This comprehensive report encompasses pivotal details such as the attack vectors employed, the specific vulnerability that was exploited, the identity of the threat actor or actors involved, the classification of the attack type, and a detailed narrative about the attack's modus operandi. Moreover, the report incorporates a comprehensive analysis of attack pattern vectors and indicators, facilitating in understanding of the incident's intricacies. Additionally, it features a graphical STIX format, providing a visual overview of the incident's characteristics. Notably, the report also offers insights into recommended patches and fixes for the vulnerabilities that were exploited to compromise the system, thus contributing to the post-incident remediation process.

In summary, the outcomes of our simulation exercises demonstrate the efficacy of HVMI in detecting and analyzing cyberattacks within cloud infrastructures. HVMI emerges as a potent instrument, providing cloud service providers and cybersecurity practitioners with a robust means to safeguard cloud environments and mitigate the repercussions of security breaches. These results underscore the significance of HVMI as a proactive and potent tool in the arsenal against cloud-based cyber threats, offering valuable insights and capabilities to enhance the security posture of cloud systems.

## 5 Conclusion and Future Work

In conclusion, the ever-expanding realm of cloud computing necessitates a robust and adaptive security framework to combat the growing sophistication of cyber threats. The proposed HVMI solution, comprised of core components like the VMI-enabled forensic application, CFT portal, and data transmission web service, offers a comprehensive and promising strategy for detecting and analyzing attacks in real-time. Its commitment to iterative optimization and enhancement, coupled with proactive measures such as fortifying the forensic application, ensures that organizations can effectively safeguard their cloud infrastructure. By continually refining their security posture through HVMI, organizations can stay resilient against emerging threats and minimize the impact on the confidentiality, integrity, and availability of their cloud systems in this dynamic cybersecurity landscape. Nonetheless, here are several areas for future research that hold potential for enhancing HVMI's efficacy.

To ensure the effectiveness of HVMI in safeguarding large-scale cloud systems, optimizing its scalability is paramount. This endeavor entails devising techniques capable of efficiently processing substantial data volumes while fine-tuning HVMI's performance under heavy loads. The scalability enhancements will enable HVMI to protect expansive cloud infrastructures effectively, even as they evolve and expand. Future research can delve into advanced techniques for detecting and analyzing cyberattacks within cloud environments. Notably, machine learning-based approaches can be explored to elevate HVMI's capacity to identify and respond to increasingly sophisticated threats. By integrating machine learning models for anomaly detection, the framework can discern novel attack patterns and adapt dynamically to emerging threats.

Moreover, future work could explore intriguing intersections with emerging technologies such as blockchain and virtual reality. Integrating blockchain technology into HVMI has the potential to enhance the security and traceability of its forensic analyses. This innovation would instill greater confidence in the integrity of HVMI's findings, as they would be anchored in an immutable blockchain ledger. Meanwhile, integrating virtual reality could revolutionize forensic analysis, creating immersive and interactive experiences that empower investigators to explore attack scenarios in unprecedented depth. Lastly, synergizing HVMI with cloud tools represents a compelling roadmap for further research. This integration could automate security processes, providing rapid responses to emerging threats. Additionally, it would facilitate the swift deployment of security measures across cloud systems, bolstering their overall security posture. These directions mark a new exciting frontier in the ongoing effort to strengthen cloud security against ever-evolving and persistent cyber threats.

**Resources** For source code and further references, see the GitHub repository.

## References

[1] F. Rehman, Z. Muhammad, S. Asif, H. Rahman, "The next generation of cloud security through hypervisor-based virtual machine introspection," in 2023 3rd International Conference on Artificial Intelligence (ICAI), 116–121, 2023, doi:10.1109/ICAI58407.2023.10136655.

[2] N. S. Shaikh, A. Yasin, R. Fatima, "Ontologies as Building Blocks of Cloud Security," International Journal of Information Technology and Computer Science (IJITCS), **14**(3), 52–61, 2022.

[3] J. Shahid, Z. Muhammad, Z. Iqbal, A. S. Almadhor, A. R. Javed, "Cellular automata trust-based energy drainage attack detection and prevention in Wireless Sensor Networks," Computer Communications, **191**, 360–367, 2022.

[4] M. Fatima, H. Abbas, T. Yaqoob, N. Shafqat, Z. Ahmad, R. Zeeshan, Z. Muhammad, T. Rana, S. Mussiraliyeva, "A survey on common criteria (CC) evaluating schemes for security assessment of IT products," PeerJ Computer Science, **7**, e701, 2021.

[5] S. Asif, M. Ambreen, Z. Muhammad, H. ur Rahman, S. Iqbal, "Cloud Computing in Healthcare-Investigation of Threats, Vulnerabilities, Future Challenges and Counter Measure," LC International Journal of STEM (ISSN: 2708-7123), **3**(1), 63–74, 2022.

[6] W. R. Simpson, K. E. Foltz, "Network Segmentation and Zero Trust Architectures," in Lecture Notes in Engineering and Computer Science, Proceedings of the World Congress on Engineering (WCE), 201–206, 2021.

[7] P. Purnaye, V. Kulkarni, "A comprehensive study of cloud forensics," Archives of Computational Methods in Engineering, **29**(1), 33–46, 2022.

[8] Z. Muhammad, F. Amjad, Z. Iqbal, A. R. Javed, T. R. Gadekallu, "Circumventing Google Play vetting policies: a stealthy cyberattack that uses incremental updates to breach privacy," Journal of Ambient Intelligence and Humanized Computing, 1–10, 2023.

[9] "Structured threat information expression (STIX™) 1.x archive website," .

[10] D. Barrett, G. Kipper, "2 - Server Virtualization," in D. Barrett, G. Kipper, editors, Virtualization and Forensics, 25–36, Syngress, Boston, 2010, doi: https://doi.org/10.1016/B978-1-59749-557-8.00002-3.

[11] Z. Aalam, V. Kumar, S. Gour, "A review paper on hypervisor and virtual machine security," in Journal of Physics: Conference Series, volume 1950, 012027, IOP Publishing, 2021.

[12] N. R. Nasab, "Security functions for virtual machines via introspection," 2012.

[13] B. Hay, K. Nance, "Forensics examination of volatile system data using virtual introspection," ACM SIGOPS Operating Systems Review, **42**(3), 74–82, 2008.

[14] B. D. Payne, D. d. A. Martim, W. Lee, "Secure and flexible monitoring of virtual machines," in Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007), 385–397, IEEE, 2007.

[15] M. A. Kumara, C. Jaidhar, "Leveraging virtual machine introspection with memory forensics to detect and characterize unknown malware using machine learning techniques at hypervisor," Digital Investigation, **23**, 99–123, 2017.

[16] P. Mishra, E. S. Pilli, V. Varadharajan, U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," Journal of Network and Computer Applications, **77**, 18–47, 2017.

[17] M. R. Watson, A. K. Marnerides, A. Mauthe, D. Hutchison, et al., "Malware detection in cloud computing infrastructures," IEEE Transactions on Dependable and Secure Computing, **13**(2), 192–205, 2015.

[18] A. K. Marnerides, P. Spachos, P. Chatzimisios, A. U. Mauthe, "Malware detection in the cloud under ensemble empirical mode decomposition," in 2015 international conference on computing, networking and communications (iCNC), 82–88, IEEE, 2015.

[19] H. Zhou, H. Ba, Y. Wang, T. Hong, "On the Detection of Malicious Behaviors against Introspection Using Hardware Architectural Events," IEICE TRANSACTIONS on Information and Systems, **103**(1), 177–180, 2020.

[20] H. Huseynov, K. Kourai, T. Saadawi, O. Igbe, "Virtual Machine Introspection for Anomaly-Based Keylogger Detection," in 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), 1–6, IEEE, 2020.

[21] B. Borisaniya, D. Patel, "Towards virtual machine introspection based security framework for cloud," Sādhanā, **44**(2), 1–15, 2019.

[22] S. Paakkola, "Assessing performance overhead of Virtual Machine Introspection and its suitability for malware analysis," 2020.

[23] Z. Muhammad, Z. Anwar, B. Saleem, J. Shahid, "Emerging Cybersecurity and Privacy Threats to Electric Vehicles and Their Impact on Human and Environmental Sustainability," Energies, **16**(3), 1113, 2023.

[24] S. Barnum, "Standardizing cyber threat intelligence information with the ... - stix," 2014.

[25] J. Shahid, Z. Muhammad, Z. Iqbal, M. S. Khan, Y. Amer, W. Si, "SAT: Integrated Multi-agent Blackbox Security Assessment Tool using Machine Learning," in 2022 2nd International Conference on Artificial Intelligence (ICAI), 105–111, IEEE, 2022.