Design of a SECaaS Service in a Private Cloud

prof. Ing. Pavel Segeč, PhD.
Department of Information Networks
Faculty of Management Science And Informatics
University of Žilina
Žilina, Slovakia
pavel.segec@fri.uniza.sk

Mgr. Jana Uramová, PhD.
Department of Information Networks
Faculty of Management Science And Informatics
University of Žilina
Žilina, Slovakia
jana.uramova@fri.uniza.sk

Ing. Peter Tanuška
Department of Information Networks
Faculty of Management Science And Informatics
University of Žilina
Žilina, Slovakia
peter.tanuska@stud.uniza.sk

2025 International Conference on Emerging eLearning Technologies and Applications (ICETA) | 979-8-3315-7426-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICETA67772.2025.11280021

***Abstract*—Cloud computing offers flexibility but raises critical security challenges. This paper presents a methodology for implementing Security as a Service (SECaaS) in a private OpenStack cloud. Combining theoretical analysis with a proof-of-concept deployment, the study demonstrates benefits, risks, and applicability of SECaaS for academic and research environments.**

***Keywords*—Cloud Computing, SECaaS, OpenStack, Cybersecurity, Private Cloud, TAPaaS, VAaaS, IDSaaS,**

I. Introduction

Cloud computing (CC) is a rapidly evolving field of information technology, offering an expanding portfolio of services utilized across academic, commercial, and public sectors. While cloud computing provides numerous advantages, the security of cloud infrastructure, data, and services remains a critical challenge. Cyber threats can compromise the availability, confidentiality, and integrity of cloud-based assets, necessitating the adoption of novel protection strategies.

One such approach is Security as a Service (SECaaS), a modern paradigm for delivering security functionalities via cloud technologies. SECaaS enables customers to access centralized security services, with the provider assuming responsibility for their deployment, maintenance, and management. This model offers several benefits, including real-time monitoring, automation of security operations, efficient incident response, and statistical threat reporting. However, SECaaS also introduces risks — such as potential exposure of multiple clients in the event of an attack on the provider's infrastructure or when service separation is inadequately enforced.

The primary objective of this research was to develop a methodology for deploying and delivering selected SECaaS services within a private cloud environment based on the OpenStack platform. The study reflects the need for an accessible, modular, and flexible security solution, particularly in academic environments where experimental validation of new approaches is feasible.

The practical component of the research was conducted as a proof-of-concept implementation at the Department of Information Networks, Faculty of Management Science and Informatics, University of Žilina, Slovakia. The theoretical part focused on analyzing the current landscape of SECaaS, reviewing relevant cloud security standards, and assessing their applicability within the OpenStack ecosystem. The subsequent sections of this paper outline the proposed methodology, implementation details, and testing outcomes obtained from the pilot deployment.

II. Related Work

A. Network Intrusion Detection System as a Service in OpenStack

Network Intrusion Detection Systems (IDS) are considered essential tools for protecting cloud infrastructures against cyberattacks. A common approach is to deploy IDS instances at the entry points of virtual networks; however, this often leads to significant virtualization overhead and operational inefficiency, as a separate IDS instance must be maintained for each tenant. Recent research introduced the concept of Network Intrusion Detection System as a Service (NIDSaaS) in OpenStack, enabling dynamic deployment, updates, and termination of IDS services through orchestration mechanisms. The proposed architecture minimizes system load for cloud providers while maintaining high availability of IDS functions for tenants. The prototype integrates a client for service management, a plugin agent, and an IDS engine such as Snort, distributed across OpenStack controller, network, and compute nodes [1].

B. Zero-Trust Architecture

The Zero-Trust Architecture (ZTA) has become a foundational principle of modern cloud security, particularly due to the rise of remote work, BYOD (Bring Your Own Device) policies, and decentralized IT environments. Unlike traditional perimeter-based models, ZTA assumes that no user or device—whether inside or outside the network—should be automatically trusted. Instead, it requires continuous verification of user identity, device posture, and context, while rigorously enforcing the principle of least privilege [2].

Recent industry surveys confirm that organizations are rapidly adopting ZTA-based mechanisms to mitigate insider threats and lateral movement of attackers. A 2024 study highlights that most enterprises are transitioning toward identity-centric access control, integrating Multi-Factor Authentication (MFA), micro-segmentation, and real-time behavioral analytics into their security workflows [2].

According to Gartner, by 2025, over 60% of enterprises will adopt zero-trust principles as their primary cybersecurity strategy, moving away from perimeter defense toward zero-trust network access (ZTNA) [3]. Palo Alto Networks further emphasizes that ZTA implementation can significantly reduce the risk of unauthorized access, enhance visibility across cloud workloads, and ensure scalability in multi-cloud and hybrid environments [3][4].

In academic and enterprise deployments alike, ZTA is seen as a critical enabler for effective SECaaS delivery. By decoupling trust from network location, SECaaS providers

Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:40:33 UTC from IEEE Xplore. Restrictions apply.

can implement centralized and dynamic policy enforcement while protecting distributed cloud resources, edge devices, and remote users across diverse cloud domains.

Additionally, ZTA lays the groundwork for compliance-driven architectures, aligning with standards such as NIST SP 800-207 and supporting auditability and risk management in regulated environments.

C. Artificial Intelligence and Machine Learning

Artificial Intelligence (AI) and Machine Learning (ML) have emerged as transformative forces in cloud security, significantly enhancing capabilities in anomaly detection, predictive threat modeling, and automated incident response. Surveys indicate that more than 90% of organizations consider AI/ML critical to their cybersecurity strategy, with many planning increased investment in AI-driven tools within the next 12–24 months [5].

These technologies enable real-time detection of zero-day exploits, behavioral deviations, and advanced persistent threats (APTs) that traditional rule-based systems often miss. Studies show that well-trained AI models can outperform human analysts in tasks such as threat triage, alert correlation, and response orchestration, thereby accelerating Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR) [5].

This paradigm shift is further amplified by the global shortage of skilled cybersecurity professionals. The 2024 ISC² Cybersecurity Workforce Study identifies a workforce gap exceeding 4.8 million specialists worldwide, with critical shortages in fields such as secure DevOps, vulnerability assessment, and incident response engineering [6][7]. This shortage has accelerated the demand for AI-driven solutions that can autonomously analyze data, prioritize risks, and suggest remediation steps with minimal human input.

As a result, security platforms are increasingly adopting proactive, adaptive, and context-aware AI systems that evolve through continuous learning. These systems move beyond static signatures and predefined policies to leverage deep learning, natural language processing, and graph-based threat intelligence to detect complex attack patterns [8].

In the context of SECaaS, AI/ML integration enables service providers to offer scalable, resilient, and cost-effective protection mechanisms that adapt in real time to dynamic threat landscapes and customer-specific requirements.

D. Standards Related to Cloud Computing

Several ITU standards define the foundations of cloud computing (CC) and its security requirements.

Standard ITU-T Y.3500 [9] provides a vocabulary and overview of CC, describing its essential characteristics: network access, resource pooling, on-demand allocation, scalability, elasticity, and abstraction of physical resources into services.

Standard ITU-T Y.3501 specifies general requirements for CC services, including lifecycle management, interoperability, portability, legal compliance, and quality of service. For IaaS, it emphasizes resource abstraction, efficient provisioning, and monitoring. It also defines conditions for trusted services, such as data governance, privacy protection, cryptographic security, resilience, and SLA compliance.

Standard ITU-T Y.3502 [10] introduces the Cloud Computing Reference Architecture (CCRA), describing stakeholder roles, functional components, and design principles. It provides a reference framework for developing interoperable and standardized cloud services.

Standard ITU-T X.1602 [11] focuses on security requirements for SaaS environments, outlining the roles of cloud service providers, partners, and customers. It describes secure layering across IaaS, PaaS, and DaaS, with emphasis on protecting application development and operation while ensuring customers access services safely.

Together, these standards form the basis for designing secure, reliable, and interoperable SECaaS solutions.

E. Existing SECaaS Solutions

Security as a Service (SECaaS) delivers cloud-based security functions managed by external providers. Customers typically pay a subscription fee covering applications, data, and process protection.

The Cloud Security Alliance (CSA) defines SECaaS categories such as identity and access management, data loss prevention, encryption, vulnerability assessment, incident management, network security, application security, Business continuity and disaster recovery (BC/DR), traffic monitoring, and email protection. Many providers integrate multiple functions into comprehensive service packages.

F. OpenStack and SECaaS Services

SECaaS offerings differ in scope, delivery, and target users. Some providers specialize in one CSA-defined category, while others deliver integrated multi-function platforms.

OpenStack, launched in 2010 by NASA and Rackspace [12], is an open-source IaaS platform widely adopted in public and private clouds [13]. It supports resource management through web, CLI, and REST API interfaces and operates exclusively on Linux. Its strengths include scalability, security, reliability, and strong community support.

The Department of Information Networks at the Faculty of Management Science and Informatics, University of Žilina, operates a private OpenStack cloud across 12 servers interconnected via Cisco Nexus switching, secured by a Fortinet Fortigate Firewall.

OpenStack includes several security-relevant services:

- Security Groups: rule sets for controlling VM traffic.
- FWaaS: network traffic filtering via Neutron.
- TAPaaS: mirroring traffic for monitoring and security integration.
- VPNaaS: IPsec VPN support for secure tunneling.
- Tatu (SSHaaS): centralized SSH certificate management.

These capabilities provide a foundation for implementing SECaaS in academic and enterprise environments.

III. Methodology and Research Methods

The research aimed to explore the deployment of SECaaS services in a private OpenStack cloud and evaluate

979-8-3315-7426-0/25/$31.00 ©2025 IEEE
Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:40:33 UTC from IEEE Xplore. Restrictions apply.

their functionality through experimental validation. The process included analyzing existing SECaaS offerings and standards, selecting suitable tools, designing a methodology aligned with CC principles, deploying and testing services in a departmental OpenStack environment.

In OpenStack, security can be implemented via built-in components or dedicated VM-based services. To address risks from tenant misuse of shared resources, we implemented a vulnerability scanning service alongside IDS/IPS solutions. Services were deployed and tested in an isolated cloud environment using DevStack and Heat Orchestration Template (HOT) scripts. Remote access mechanisms were configured to support testing.

A. *Preparation of the Test Environment*

A dedicated VMware ESXi server (32 GB RAM, 16 CPU cores, 800 GB storage) provided the infrastructure. OpenStack (Victoria release) was installed on Ubuntu 20.04 using DevStack.

Since the server was internet-accessible, it was hardened with the UFW firewall, restricting access to internal IP address ranges and VPN clients.

DevStack scripts automated installation, creating default network settings and floating IP allocations.

B. *Cloud Installation, Networking and Testing.*

Proper configuration ensured connectivity for user instances. A public interface (br-ex) enabled outbound traffic and floating IP assignment, while ICMP and SSH rules were added to security groups.

Six devices were deployed via HOT, including an OpenStack router, five Linux VMs (including a Router VM), and a Jump-in VM for remote access. Router VM performed inter-network forwarding with IP forwarding enabled. Anti-spoofing features in Neutron required disabling port security or defining additional allowed IP/MAC pairs for NFV.

This environment established the foundation for deploying and evaluating SECaaS services within OpenStack.

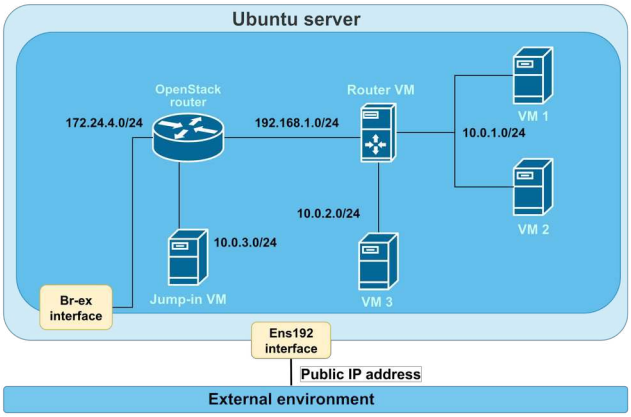


Figure 1 – Tested virtual infrastructure in the cloud

To manage cloud instances, a public SSH key must be defined at launch. In our setup, a key pair was generated and its public key distributed across all VMs, enabling remote login with the corresponding private key.

A dedicated Jump-in VM served as the entry point to the infrastructure, with a floating IP address assigned for external access. Floating IPs provide flexibility compared to fixed IPs, as they can be reassigned between instances. In our topology, the Jump-in VM received the floating IP 172.24.4.10, ensuring consistent connectivity.

Access to other VMs from the Jump-in VM requires the same private key used for initial login. Storing private keys on intermediate devices is insecure; therefore, we adopted SSH agent forwarding. This mechanism keeps keys locally on the client, while allowing remote systems to authenticate via forwarded credentials. Thus, private keys remain secure and are never transmitted across devices.

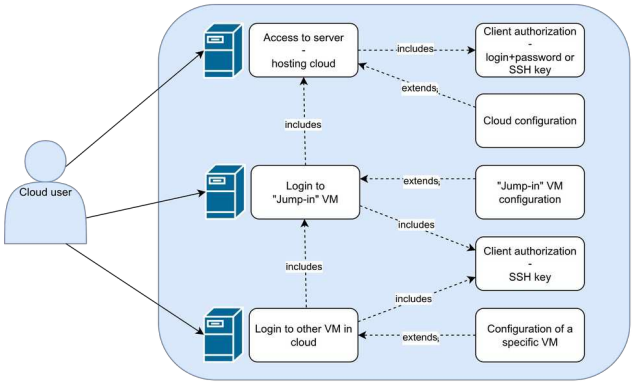


Figure 2 – Use-case diagram for cloud access

IV. RESULTS

A. *Implementation of Cloud IDS/IPS Service*

Two scenarios were evaluated. In the first scenario, an IDS is launched on the Jump-in VM, which receives mirrored IP flows from SECaaS clients in the cloud. Traffic mirroring is provided by the TAPaaS service in OpenStack, which allows for flexible redirection of selected tenant traffic to a monitoring instance. In the second scenario, a transparent IPS is deployed on the Router VM so that traffic of cloud instances traverses it. Upon identifying malicious activity, the IPS blocks the corresponding packets, thus protecting SECaaS clients from hostile sources. Both approaches showcase how passive and active security mechanisms can be implemented in multi-tenant cloud infrastructures without requiring changes on the client-side.

B. *Scenario 1 – Network IDS via TAPaaS*

This scenario focuses on implementing a network IDS using the Neutron TAPaaS service to mirror interface traffic in the cloud. The IDS runs on the Jump-in VM, which functions as a centralized inspection node. We define VM 1 and VM 3 as SECaaS service clients. The cloud is configured to mirror ingress traffic on their network interfaces. TAPaaS enables the administrator to specify directionality and select interfaces precisely, allowing for low-overhead, targeted traffic analysis. Finally, functionality is validated by launching attacks against VM 1 and VM 3 from VM 2, which acts as the attacker node. In this scenario, clients are not actively protected; only the occurrence of security incidents is detected and logged.

TAPaaS can be managed via the Neutron CLI or Horizon (with TAPaaS-dashboard). We used the CLI for its simplicity and suitability for scripting. A TAP service acts as the destination for mirrored traffic, while TAP flows specify interfaces and direction (IN, OUT, BOTH). For our scenario, only ingress traffic from VM1 and VM3 was mirrored to observe inbound threats. Functionality was initially verified

979-8-3315-7426-0/25/$31.00 ©2025 IEEE
Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:40:33 UTC from IEEE Xplore. Restrictions apply.

with a simple ping test between the VMs and confirmed via packet inspection at the IDS.

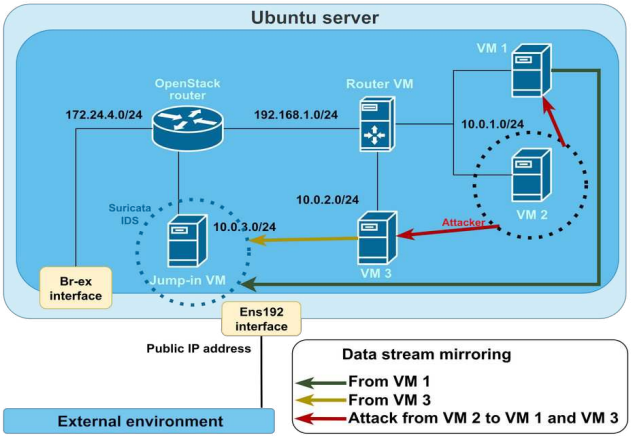


Figure 3 – IDS service implementation in the cloud

Suricata was installed using APT with the OISF repository. This ensured access to the latest detection engine with continuous rule updates. The capture interface was defined in /etc/default/suricata to bind Suricata to the correct monitoring interface. YAML-based configuration specified VM1 and VM3 as the HOME_NET, with all others marked as EXTERNAL_NET. This allowed Suricata to distinguish between internal assets and potential sources of attack. Custom rules were added to /var/lib/suricata/rules/my.rules, including detection of SSH login attempts from external hosts and DoS attacks on HTTP services, tailored to our test scenario.

After restarting Suricata, simulated attacks from VM2 were detected: unauthorized SSH attempts and a SYN-flood against VM1's HTTP service using hping3. Detection was immediate and alerts were generated in EVE JSON format. These logs recorded detailed threat information, such as source/destination IP addresses, protocol, ports, severity level, and timestamps. Alerts could be further processed using visualization tools or integrated into a SIEM for centralized analysis.

In IDS mode, Suricata does not block traffic but enables advanced attack detection and passive monitoring. This setup is particularly suitable for environments where inline filtering is not acceptable due to performance or compatibility concerns. It provides insights into potential intrusions without impacting availability or latency.

In production, IDS-based SECaaS could be deployed as a scalable service on dedicated VMs or containers, with administrators assigning TAP flows dynamically to subscribing clients. This model supports flexible tenant isolation and aligns well with Zero Trust architectures [3][4].

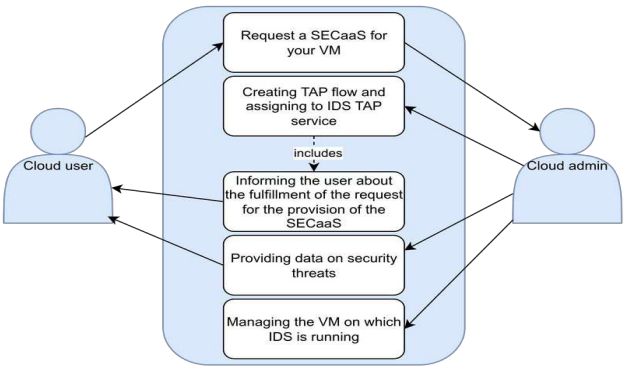


Figure 4 – Use-case diagram of the cloud IDS

C. *Scenario 2 – Transparent IPS*

The second scenario tested an active IPS deployed on the Router VM, which inspects and protects traffic of directly connected instances (VM1–VM3). This architecture ensures that any communication between clients or with external networks passes through the inspection point. Attacks were launched from the Jump-in VM to simulate threats originating within the same cloud infrastructure. Unlike the IDS scenario, Suricata in IPS mode actively blocks malicious traffic using Linux Netfilter Queue (NFQ), leveraging iptables for traffic redirection. This mode introduces a small latency overhead but provides immediate protection to tenants.

For signature management, we used suricata-update with the Emerging Threats Open ruleset enabled, which contains a wide array of community-maintained detection signatures. This ensures timely response to emerging threats and aligns with best practices for threat intelligence sharing.

The IPS setup mirrors the IDS configuration in most aspects but requires additional NFQ support in the Linux kernel. Traffic is redirected to Suricata with rules such as:

iptables -I FORWARD -j NFQUEUE

to inspect forwarded traffic. Optionally, the host itself can be protected with:

iptables -I INPUT -j NFQUEUE

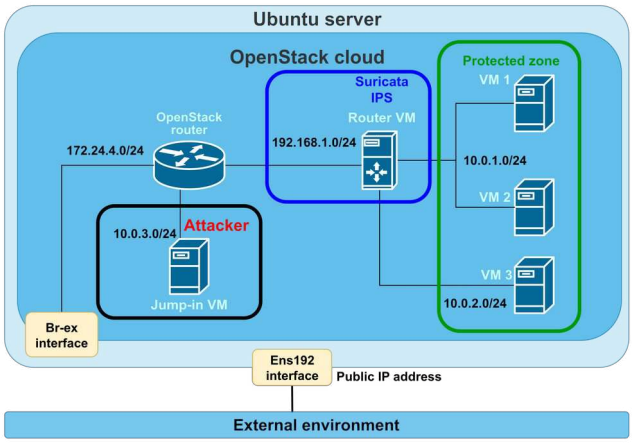


Figure 5 – IPS service implementation in the infrastructure

1) *Testing and Results*

A 5-minute SYN-flood DoS against VM1's HTTP port was executed with and without IPS. Without IPS, system load averaged 81%, indicating high CPU usage and reduced responsiveness. With IPS enabled, load dropped to 23%, a 58% reduction. This demonstrated effective mitigation. Of approximately 561,000 generated packets, around 496,000 reached the Router VM. Suricata identified and dropped ~491,000 packets (≈99%) as malicious.

Log files confirmed precise rule triggering and blocking actions. Each dropped packet was recorded with metadata, enabling post-incident analysis. This scenario validated that deploying a transparent IPS on a strategic VM enables effective SECaaS delivery for internal clients without modifying their own configurations.

Furthermore, the IPS scenario allows dynamic onboarding of new tenants. New client VMs can be attached directly to the protected network segment, while existing ones may require reconfiguration. The centralized protection

979-8-3315-7426-0/25/$31.00 ©2025 IEEE
Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:40:33 UTC from IEEE Xplore. Restrictions apply.

model is especially suitable for cloud providers offering security services to multiple customers.

This setup also supports adaptive policies, where traffic classification, threshold-based alerts, or even machine learning-based rulesets could be introduced in the future [5] [8]. While introducing minimal latency, the benefits of real-time protection outweigh the trade-offs in high-risk environments.

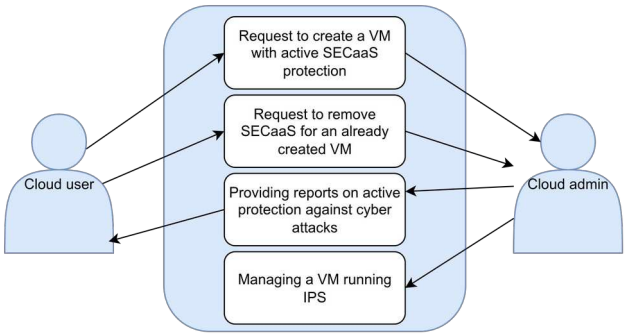


Figure 6 – Use-case diagram of the cloud IPS

D. *Implementation of a Vulnerability Assessment as a Service (VAssS)*

Work [14] introduced the concept of VAaaS, expanded in our experiment by implementing Greenbone GVM in an isolated OpenStack cloud to detect vulnerabilities on running instances and exposed services.

We selected OpenVAS, consolidated since 2019 into Greenbone Vulnerability Management (GVM), which provides detailed outputs and supports automation, scheduled scans, and custom configurations. GVM aligns with modern standards such as GDPR and is validated in research on large-scale network exposure [15].

The VA service ran on the Jump-in VM. Since the OpenStack host firewall blocked inbound access, we modified the hardening script: HTTPS traffic to the cloud server's public address on port 4040 was translated to the Jump-in VM's floating IP and port 443. Security was ensured using UFW, while advanced rules (port forwarding, NAT) were implemented in /etc/ufw/before.rules.

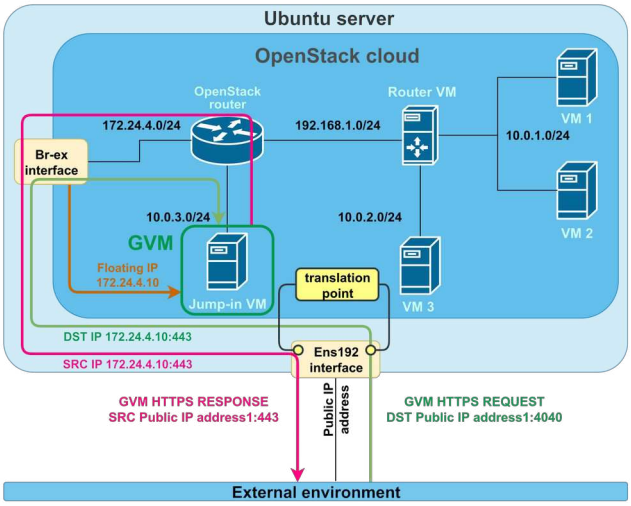


Figure 7 – Exposing GVM from the infrastructure to the Internet

Study [14] assessed an OpenStack cloud using Nessus, Metasploit, and OpenVAS on five VMs (Figure 8). Nessus reported 2 critical, 3 medium, and 1 low vulnerabilities externally; Metasploit found no external issues but listed open ports; OpenVAS matched Nessus and added a low-severity TCP timestamp risk. Internal scans gave detailed reports, with the Windows server showing 11 issues. Metasploit revealed most ports, Nessus offered mitigation guidance, and OpenVAS provided structured, detailed results as a free tool.

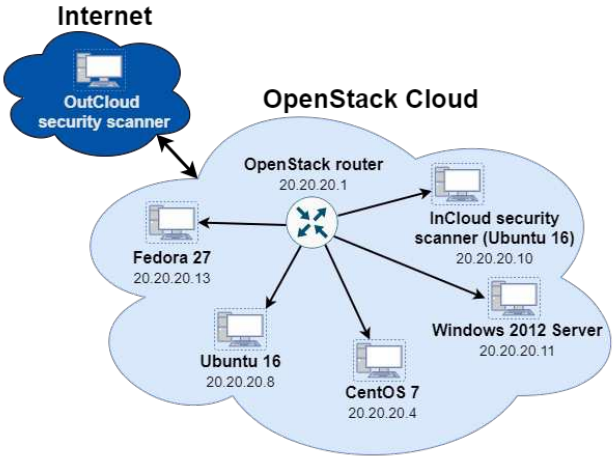


Figure 8 – Test infrastructure for cloud vulnerability assessment [14]

We deployed GVM to scan internal VMs. Services installed included Apache2 (VM 1), XRDP with GNOME (Router VM), and FreeRADIUS (VM 2). Scans, run with 70% Quality of Detection, lasted ~2.5 hours. Results correctly identified OS versions and services. The most severe issue was default credentials on the GSA web interface; TLS v1 use was also flagged. VNC console access lacked authentication, though mitigated by SSH key login. Medium risks were tied to etcd 3.3.12, and low risks to TCP timestamps.

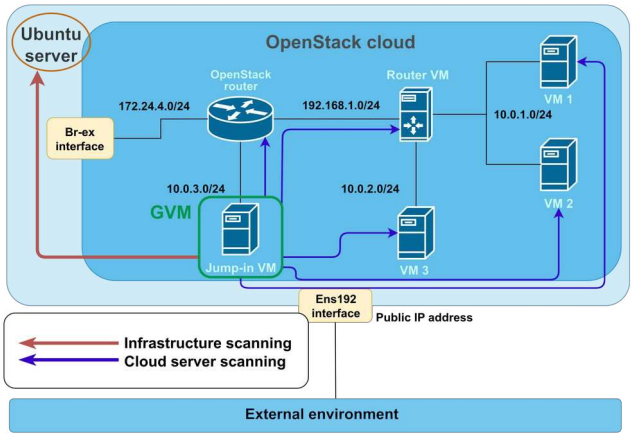


Figure 9 – Devices scanned by GVM

V. Discussion of Results

During the design and implementation phase, we successfully deployed three distinct Security as a Service (SECaaS) services and evaluated them within the experimental topology of a private OpenStack cloud, as illustrated in Figure 1.

The development process began with a review of relevant literature on security services in OpenStack-based cloud environments, primarily sourced from the IEEE Xplore Digital Library. This analysis enabled us to identify feasible approaches for delivering SECaaS in OpenStack infrastructures. Subsequently, the server provided by the

Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:40:33 UTC from IEEE Xplore. Restrictions apply.

Department of Information Networks was hardened using the UFW firewall, and a private OpenStack cloud was installed via DevStack scripts.

Before implementing the individual SECaaS services, we concentrated in our research on these services:

(i) Intrusion Detection and Prevention Systems (IDS/IPS) to detect and block malicious activity targeting cloud instances, and

(ii) vulnerability assessment for auditing running virtual machines.

Accordingly, three SECaaS scenarios were designed, implemented, and tested within the private cloud.

The first scenario involved deploying a centralized Network IDS using the TAPaaS service, allowing mirrored traffic from protected instances to be inspected. Although this setup successfully detected attack traffic, it did not block it — malicious packets continued to reach target virtual machines.

The second scenario introduced a Network IPS, which extended functionality by actively blocking detected malicious traffic. Both IDS and IPS services were implemented using the Suricata open-source engine.

The third scenario focused on providing a vulnerability assessment service. We deployed Greenbone Vulnerability Manager (GVM) on a dedicated Jump-in VM. The deployment proved more challenging due to the modular nature of the tool, which consists of several interconnected components. After multiple installation iterations, a stable and functional setup was achieved. Since the web interface was hosted inside the cloud, we configured iptables rules to translate addresses and ports, enabling external access and scan initiation.

VI. Conclusion

The primary objective of this research was to explore the feasibility of delivering Security as a Service (SECaaS) within a private OpenStack cloud. Due to the absence of standardized SECaaS implementation methodologies, we adopted a Software Development Life Cycle (SDLC) approach to decompose the task into sequential phases.

In the planning and analysis phases, we identified key cloud security challenges and reviewed relevant standards. Commercial SECaaS offerings were examined to guide the design of services aligned with our use cases.

As the solution was built on OpenStack, we analyzed its components, architecture, and built-in security features. Mechanisms were selected with a focus on educational and research applicability.

The test environment was deployed using DevStack on departmental infrastructure. A hardened server hosted the experimental cloud, including five virtual devices and one OpenStack router. External access was provided via a dedicated Jump-in VM with a floating IP.

Three SECaaS services were developed and validated: IDSaaS with TAPaaS, IPSaaS, and VAaaS. IDS and IPS functionalities were implemented using Suricata. The IDS detected threats from mirrored traffic, while the IPS actively blocked malicious packets. The third service, VAaaS, used GVM to generate detailed vulnerability reports based on severity and exposed services.

These results confirm the feasibility and practicality of deploying customizable SECaaS solutions in private academic cloud environments based on OpenStack.

In future work, we plan to publish results from multi-tenant experiments demonstrating policy isolation and enforcement in OpenStack-based SECaaS environments.

ACKNOWLEDGEMENT

Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00010.

REFERENCES

[1] Salanke, V. S., & Nagasundari, S. (2024). Leveraging OpenStack-based Private Cloud for Intrusion Detection Using Deep Learning Techniques. 1st International Conference on Intelligent Systems, IEEE. [Online]. Available: ResearchGate

[2] Weinberg, A., & Cohen, M. (2024). Zero trust implementation in the emerging technologies era: a survey. ResearchGate. [Online]. Available: https://www.researchgate.net/publication/379995483

[3] Gartner. (2024). Zero Trust Is an Essential Strategy to Reduce Risk. [Online]. Available: https://www.gartner.com/en/articles/zero-trust-strategy

[4] Palo Alto Networks. (2025). The State of Zero Trust Security 2025. [Online]. Available: https://www.paloaltonetworks.com/resources/whitepapers/state-of-zero-trust-2025

[5] Mamodiya, U., Pandey, S. K., & Neela, V. (2024). Secure Cloud Traffic Management using Trust-Aware Machine Learning to Identify Distributed Denial of Service Attacks. IEEE International Conference on Intelligent Systems. [Online]. Available: ResearchGate

[6] ISC². (2024). Cybersecurity Workforce Study 2024. [Online]. Available: https://www.isc2.org/research

[7] Goupil, A., et al. (2022). Cybersecurity Talent Gap: Academic and Workforce Challenges. In collaboration with ISC². [Online]. Available: https://www.isc2.org/News-and-Events/Press-Room/posts/2022-cybersecurity-skills-gap

[8] SANS Institute. (2024). AI and Its Growing Role in Cybersecurity. [Online]. Available: https://www.sans.org/white-papers/ai-in-cybersecurity/

[9] ISO/IEC. (2014). ISO/IEC 17788 - Y.3500 SERIES – Information technology – Cloud computing – Overview and vocabulary. ITU Series Y. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3500-201408-I/en

[10] ITU-T. (2014). ITU-T Y.3502 Cloud computing – Reference architecture. ITU Series Y. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3502/en

[11] ITU-T. (2016). ITU-T X.1602: Security requirements for software as a service application environments. ITU-T Series X. [Online]. Available: https://www.itu.int/rec/T-REC-X.1602-201603-I/en

[12] OpenStack Foundation. (2024). A Bit of OpenStack History. [Online]. Available: https://docs.openstack.org/project-team-guide/introduction.html

[13] OpenStack Foundation. (2025). Open Source Cloud Computing Infrastructure – OpenStack. [Online]. Available: https://www.openstack.org/

[14] E. Rigó, Z. Komáromi, B. Bencsáth, & G. Ládi. (2025). Attack Surface Mapping of Country-Wide Networks: A Technical Report. In Proc. IEEE 29th International Conference on Intelligence in Next Generation Networks (ICIN).

[15] W. Soussi, M. Christopoulou, & G. Gür. (2023). MERLINS – Moving Target Defense Enhanced with Deep-RL for NFV In-Depth Security. In Proc. IEEE SDN/NFV Conference.

979-8-3315-7426-0/25/$31.00 ©2025 IEEE Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:40:33 UTC from IEEE Xplore. Restrictions apply.