ESP-IJACT
ESP International Journal of Advancements in Computational Technology
ISSN: 2583-8628 / Volume 2 Issue 1 March 2024 / Page No: 190-201
Paper Id: IJACT-V2I1P121 / Doi: 10.56472/25838628/IJACT-V2I1P121

Original Article

Dynamic Threat Modeling For Internet-Facing Applications in Cloud Ecosystems

Chaitanya Vootkuri
Distinguished Cloud Security Architect, USA.

Received Date: 26 February 2024 Revised Date: 08 March 2024 Accepted Date: 29 March 2024

Abstract: *The more a cloud services provider relies on Internet-facing applications, the more important the security becomes. As security challenges are evolving, dynamic threat modelling is an emerging critical methodology. Dynamic threat modelling is different from static approaches because it combines real-time data and lets the real-time landscape and architecture change constantly. This paper explores principles and techniques for Internet-facing applications in cloud environments through dynamic threat modelling. It is the identification and prioritisation of potential attack vectors, using cloud-native tools for continuous monitoring, and applying machine learning to detect and predict threats. We also talk about integrating security automation frameworks like Infrastructure, such as Code (IaC) scanning, container security, and runtime protection. Risk mitigation emphasis is placed on Distributed Denial Of Service (DDoS), unauthorised access, and data breach risks faced in Internet-facing systems. This paper combines theoretical insights and practical implementations to give security professionals and cloud architects actionable guidance. These results show how dynamic threat modelling can provide significant resilience to cloud-based applications while maintaining operational agility.*

Keywords: *Dynamic Threat Modelling, Internet-Facing Applications, Cloud Security, Threat Intelligence, Infrastructure As Code (IAC).*

I. INTRODUCTION

A. The Rise of Internet-Facing Applications in Cloud Ecosystems

Adopting cloud ecosystems has revolutionised how organisations can build, deploy, and scale Internet-facing applications. These applications are provided as part of public network access, allowing customer conversations and operational efficiencies to flow seamlessly. [1-3] However, since they are also exposed to the Internet, they are a potential cyberattack target. As with most cloud environments, scalability and reliability come at the expense of security, caused by shared infrastructure, complex architecture, and dynamic workload.

B. Evolving Threat Landscape

The threat landscape of Internet-facing applications is changing all the time. Attackers use advanced techniques, including automatically executed botnets, zero-day exploits, and social engineering, to gain an initial foothold in systems. Since the dynamic and unpredictable risks of the modern threat landscape cannot be addressed with traditional static threat modelling approaches, we demonstrate a novel method focused on detecting memory exploits in the Python programming language. However, they lack the nimbleness to adapt in real-time to such threats, which leaves great security holes.

C. The Need for Dynamic Threat Modeling

Dynamic threat modelling proves to be a proactive approach toward protecting Internet-facing applications. Dynamic threat modelling differs from traditional methods for identifying and mitigating threats because it has real-time data, threat intelligence, and automated analysis, to name a few, that constantly detect and mitigate vulnerabilities. It uses cloud-native services like elastic scaling, security monitoring and automated response systems to keep in front of the attackers.

II. RELATED WORK

A. Dynamic Threat Modeling in IoT Environments

Dynamic threat modelling has been increasingly utilised in IoT distribution, where systems are highly dynamic. In 2024, Salayma et al. [4] presented a pioneering threat modelling methodology for the IoT environment. Their research shows the weaknesses of static attack graph approaches that do not incorporate the dynamic element of IoT, which frequently sees devices joining and leaving the network. In order to tackle this, the authors suggested the exploitation of a dynamic attack graph built on graph database management tools such as Neo4j. Real-time updates of the system's topology can be provided to these graphs so that security analysts can visualise and truly understand how threats propagate through the system. One important aspect of this work is that it takes an approach that maintains operational integrity even if some parts of the network are compromised. While this principle applies to cloud ecosystems that have interconnectivity between services and applications, the same vulnerabilities arise. The results of this research introduce an imperative need for dynamic modelling techniques that can adjust to the dynamically changing state of systems in a cloud environment.

B. Cloud Application Threat Modeling

A structured methodology for threat modelling is provided for cloud-based applications. His work focuses on developing a detailed process for identifying potential attack vectors, system vulnerability analysis, and subsequent implementation of the appropriate security controls. Carter's methodology is based on one cornerstone: creating a threat model diagram that visually connects application components, data flows, and trust boundaries. It allows security teams and stakeholders to better understand the weak points from the chart itself and, thus, how to prioritise mitigation strategies. [5-8] Carter also talks about how threat modelling gets integrated into the software development lifecycle (SDLC) and used to ensure risks are identified early in the design phase. This work provides actionable guidance and derives practical templates for those entities seeking to enhance their security position in cloud ecosystems.

C. Threat Modeling in Cloud Platforms

This blog post explores threat modelling across major cloud platforms like AWS, Azure, and Google Cloud Platform (GCP) and why it is becoming complex with all the native cloud technologies introduced. In addition, serverless computing, containerised workloads, and microservices architecture increase the attack surface, which requires specialised attention to those security configurations. There is a shared responsibility model of cloud services, where the cloud providers and customers all have their responsibility. It advocates the idea of threat modelling integration in DevOps workflows to create organisations where identified risks can be identified during the development and deployment phases. By adopting this proactive approach to compliance with best practices, vulnerabilities can be mitigated so they can't be exploited. However, the insights from this discussion are especially important for related application areas of dynamic threat modelling because they emphasise the importance of continuous assessment and real-time adaptation to rapidly changing security environments on cloud ecosystems.

III. SYSTEM ARCHITECTURE AND PROBLEM CONTEXT

The dynamic threat modelling system architecture proposed here aims to overcome the special problems of cloud ecosystems presenting Internet-facing applications. [9-12] By interacting with legitimate end users and malicious actors, these applications expose them to many different attack vectors. In order to mitigate these risks effectively, the system develops components capable of real-time threat identification, logging, risk assessment, and mitigation strategy development. The architecture uses cloud-native tools and services to maintain scalability and agility to cover ever-changing security threats continually. The main target of potential attacks is encapsulated by the Internet Facing Application module, which forms the foundation of the architecture. In particular, this covers web applications which consist of frontend interface components, API gateway, and backend logic. The first set consists of components that process user interactions and perform business logic; these components interface with the cloud ecosystem for storage and computing services, networking, and other services. The interconnected nature of the system emphasises the need for robust monitoring and on-the-fly security interventions that can detect and stop malicious activity.

The architecture uses a Dynamic Threat Modeling Framework to identify, log and overcome potential threats. The framework continuously analyses API traffic and system logs using a threat identification engine and a monitoring and logging system. A risk assessment module then evaluates the threats and feeds actionable insight to a mitigation strategy engine to create a proactive response. The system allows organisations to respond dynamically to threats and provision applications in a secured and operationally secure manner hosted in the cloud, integrating these components. The architecture shows how the cloud ecosystem matches the cloud security shared responsibility model. Underlying Infrastructure and updates are managed by cloud service providers, but application owners must do everything from the application to the data level to add additional security workloads. The essence of this interaction is collaboration between cloud providers and users, using their efforts to keep the environment safe. The dynamic threat model system architecture is shown in the figure, which shows interactions between users, Internet-facing applications, cloud ecosystems, and the dynamic threat modelling framework. It shows data flow, the attack vectors and the system's monitoring mitigation process.

- Users and Threat Sources: We will depict two categories of users: legitimate end users and malicious actors. As end users use the web application from the front side, malicious actors attack it through some attack vector in the API or another (such as unauthorized access).

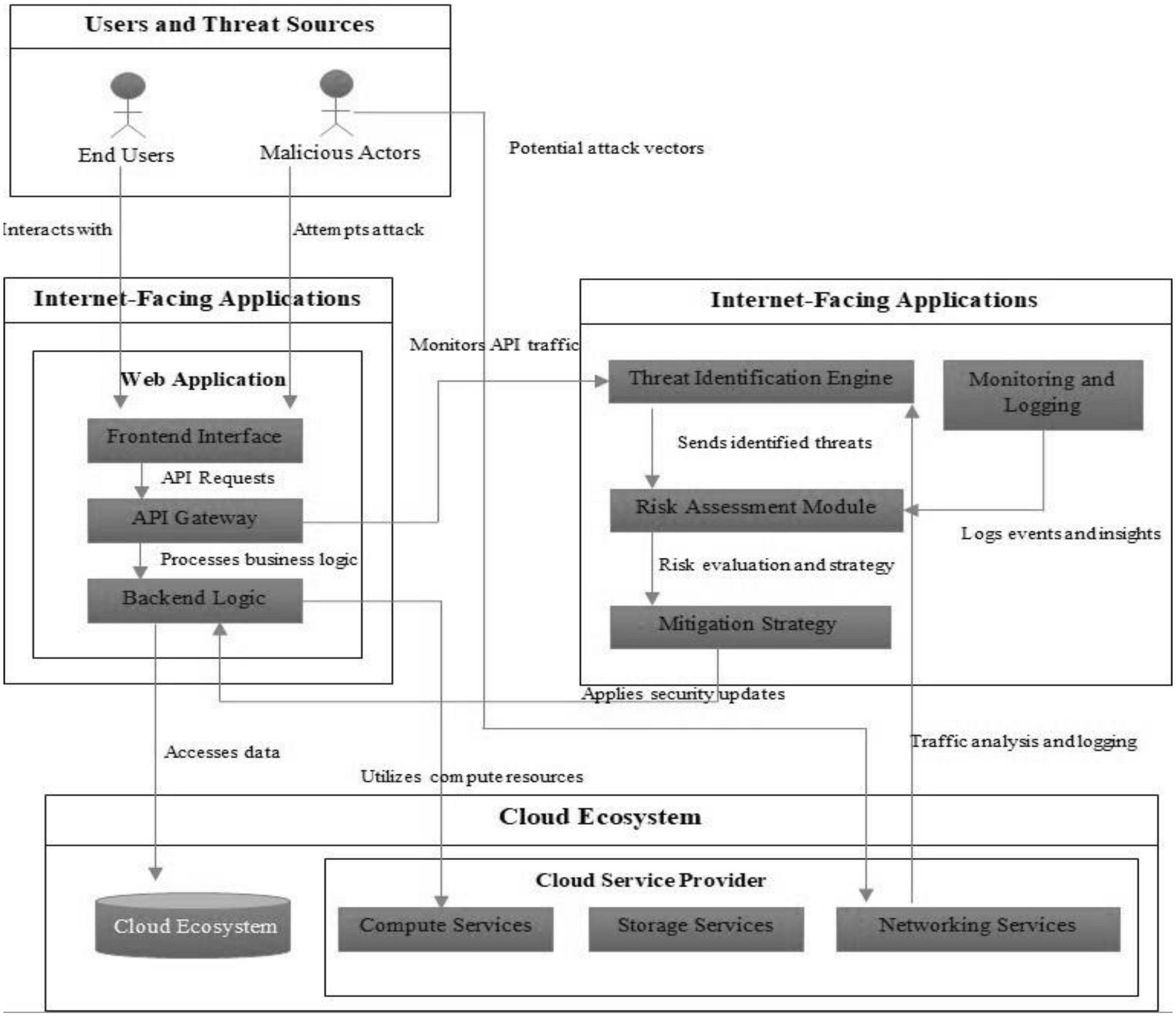

Figure 1: Dynamic Threat Modeling System Architecture

- Internet-Facing Applications: The web app is a frontend, an API gateway, and backend logic. They handle user requests, process business logic and talk to the cloud ecosystem. In particular, the API gateway is a critical component in which API traffic is observed for attack vectors, which makes it a key focus of dynamic threat modelling.
- Cloud Ecosystem and Service Provider: The cloud ecosystem has the foundational Infrastructure computing, storage, and networking services. Through this ecosystem, application data is accessed and processed while the cloud service provider applies security updates. The second collaborative environment showed that securing all layers of the system has become an essential aspect.
- Dynamic Threat Modeling Framework: This architecture is based on a central component named the threat identification engine, which feeds into a monitoring and logging system, risk assessment module, and mitigation strategy engine. It helps keep a traffic check, identify threats, evaluate risks, and implement mitigating steps. These processes provide insights into the risk to the overall security posture of the application.

A. Overview of Internet-Facing Applications

Internet-facing applications are Internet-accessible software systems designed to enable seamless interaction between users and some services. These include web applications, mobile applications, and interfaces of APIs and online platforms where users can interact with business processes. In most cases, we find that these applications involve a combination of numerous components, for example, frontend interfaces, backend services, and APIs, that share information to provide services and functionality. Given their exposure to the Internet, these applications are vulnerable to a huge variety of threats, such as brute force attacks, injection attacks and DDoS campaigns. An architecture that often integrates third-party systems, cloud services, and microservices brings great functionality but also means that they have a bigger attack surface. Further, data exchange by APIs for data exchange introduces potential entry points for attackers. These applications need to be scaled up rapidly, and they should be continuously deployed; thus, there is a need for dynamic and real-time threat detection mechanisms to secure and reliably use these applications.

B. Unique Challenges in Cloud Ecosystems

Although cloud ecosystems provide many scalability benefits, cost efficiency, and flexibility, they also have unique security issues. However, one of the big downsides is that because the model is shared responsibility, cloud service providers are responsible for securing the underlying infrastructure, while the application and data have to be secured by the users. Cloud misconfiguration, namely open storage buckets or overly lax access controls, is the cause of data breaches and unauthorised access more often than you'd think. Cloud services are dynamic and elastic, adding another layer of complexity. Virtual Machines, containers, and serverless functions can be easily created and destroyed so quickly that keeping the security configurations consistent is very hard. Additionally, the generous use of multi-cloud or hybrid cloud environments makes security monitoring and threat mitigation more difficult, thanks to variances in standards and tools across platforms. In addition, the addition of sophisticated technologies, such as microservices and container orchestration frameworks (e.g., Kubernetes), expand the attack surface, making organisations need to embrace a holistic and proactive threat modelling strategy.

C. Security Assumptions and Threat Landscape

Several security assumptions are made in a dynamic threat modelling context to simplify risk analysis. Assuming there is always an attacker probing systems at public-facing interfaces and allowing attackers to exploit misconfigurations, lack of patches, and weak authentication mechanisms. In addition, the threat landscape includes internal and external actors. Methods used by external attackers commonly include phishing, DDoS attacks, and zero-day exploits; internal threats can include disgruntled employees or inadvertent mishandling of sensitive data. This landscape is expanded by the cloud ecosystem, which brings with it threats particular to the virtualised environment container escape attacks, privilege escalation, and insecure API endpoints, among others. As we introduce cloud native technologies, other challenges come with supply chain attacks on shared libraries and third-party dependency vulnerabilities. In addition, attacks are becoming increasingly more sophisticated, automated, and AI-driven, and security teams will have to use machine learning and advanced analytics to detect and respond to threats.

IV. PROPOSED DYNAMIC THREAT MODELING FRAMEWORK

Dynamic threat modelling is a proactive means of securing Internet-facing applications in cloud ecosystems. This framework is based on real-time data and automation, unlike static threat models that don't adapt to evolving threats or system architectures. [13-16] The proposed framework is built to continuously find threats, evaluate risks, and mitigate this challenge in an integrated fashion with cloud architectures. It registers a robust, scalable and automated methodology to forestall the singular risks for cloud environments. This diagram effectively binds the architecture of a cloud ecosystem and is based on the data lifecycle and how it is involved with certain infrastructure components. It focuses on the three important phases of data: Data in the Rest, Data in Process, and Data in Transit. The threat modelling phases, though, centre around what state data may find itself in and where it might be exposed to the perils of attacks based on where it exists as it's processed or transmitted. Throughout all phases, cloud operations are built upon these pillars of interaction between underlying cloud components, including storage systems, processors, and network infrastructure. The management hierarchy of Management, Control, and Business layers, raised above the lifecycle phases, outlines the role of different stakeholders in performing secure and efficient data handling. Governance and compliance are the realms of the management layer, security and operational oversight are handled under the control layer, and business tilt is achieved with the business layer. The layered structure illustrates the requirement for a joint response to the building's business risks and technical vulnerabilities.

The diagram also highlights the external influences on cloud ecosystems like Corporate Admins, Business users and interaction with other cloud components. Because these elements allow varying degrees of access and potential misconfigurations and introduce third-party dependencies, [17] they are critical to modelling dynamic threats. For example, malign actors may exploit potential weaknesses in Network Infrastructure to attack Data in Transit; therefore, such encryption and observation facilities are necessary. Placing this image in the Proposed Dynamic Threat Modeling Framework section is a visual bridge for linking the discussion in concept to practice. It describes how data flows between layers and phases and constitutes a great basis for creating a threat modelling framework for addressing vulnerabilities at different steps in the data lifecycle.

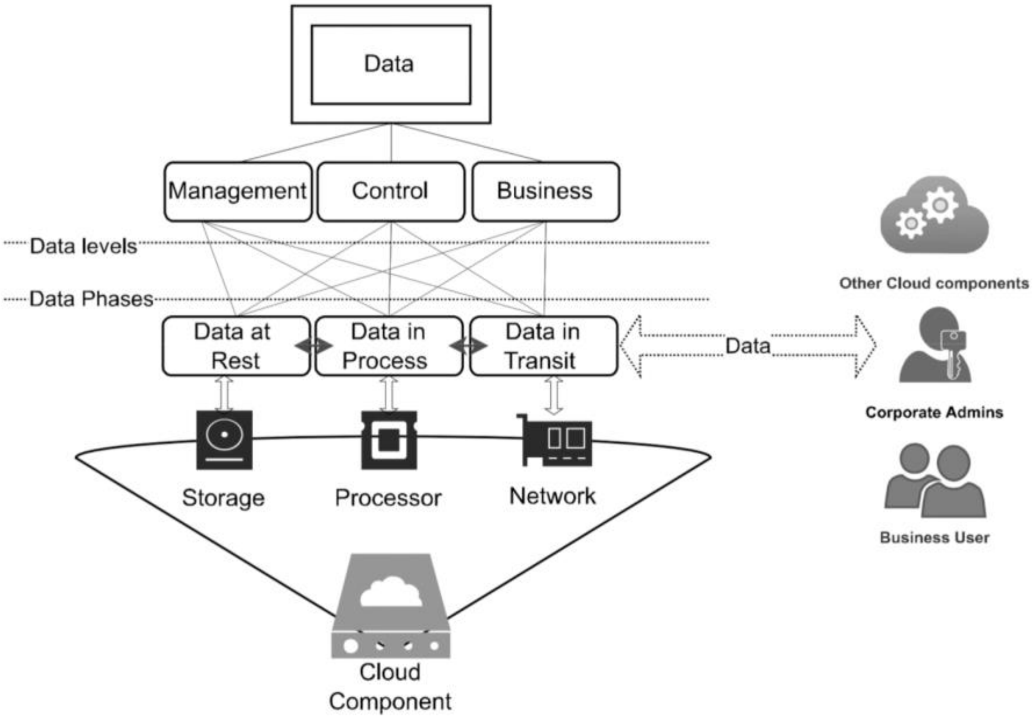

Figure 2: Cloud Data Lifecycle and Architecture

A. Framework Overview

Dynamic threat modelling framework works in a continuous security lifecycle, benefiting from modern DevSecOps practice principles. The process starts by determining the vulnerability and risk assessment of potential threats, followed by a structured risk assessment process to rank potential vulnerabilities according to their impact and likelihood. Finally, the framework provides mitigation strategies for these identified risks so the application remains secure and in operation. Here, central to this framework is the ability to process and analyse real-time data from application traffic, system logs and cloud-native monitoring tools. The goal is to work seamlessly with existing development pipelines and help end organisations build security as the fundamental aspect of their workflow. This real-time adaptability guarantees that the framework does not only detect but evolves with the systems it protects.

B. Key Components

a) Threat Identification

Threat identification is the first and most important component of the framework. In this process, we continuously monitor application traffic, interactions with our APIs and user behaviour to detect potential attack vectors. To identify patterns indicating malicious activity codes, they are run using machine learning models and threat intelligence feeds that draw patterns of abnormal API calls, failed logins repeating, or abnormal data access. Cloud-native security services such as AWS Guard Duty Azure Security Centre add capabilities for threat detection in cloud ecosystems.

b) Risk Assessment

The framework once identifies threats and then evaluates the risk of each threat. As outlined earlier, this includes the evaluation of possible impacts and exposure to exploitation for demonstrated sprints. The risk assessment module prioritises risk in light of metrics like the Common Vulnerability Scoring System (CVSS), real-time data analytics, etc. Escalated are high-risk vulnerabilities, such as those impacting critical business processes or sensitive data. The framework automates the risk assessment process, increasing the speed of response and decreasing the man burden on security teams.

c) Mitigation Strategies

The last component of the framework is the implementation of mitigation strategies for identified risks. They range from applying security patches to reconfiguring access controls, enabling runtime protection tools, and deploying Web Application Firewalls (WAFs). In cloud environments, security best practices are automatically enforced at scale using Infrastructure as Code

(IaC) templates. Moreover, our framework includes response protocols to handle active threats and maintain the least possible disruption to application functionality.

C. Integration with Cloud Architectures

The framework proposed is intended to be integrated well into cloud architectures, taking advantage of the scalability and elasticity of cloud layers to strengthen security operations. Third, cloud-native services, like logging and monitoring tools (such as AWS CloudWatch and Azure Monitor), provide the data you need to identify threats and assess risk. These services enable the framework to collect and read logs in real-time from different components such as API gateways, compute instances and storage services. Also, the framework concurs with the cloud vendor's weak spot construct of responsibility, which joins a cloud provider's security checks out with customer-prepared identity protection. The framework secures application security, user interaction, and configuration security, while the cloud provider protects infrastructure security. By integrating container orchestration platforms (Kubernetes, to name one), the framework can monitor and secure microservices architectures from the risk of dynamic or distributed environments. The framework guarantees applications are secure against evolving threats and historically and performance scalable by embedding security into cloud architectures. By integrating systems in this way, dynamic threat modelling becomes an imperative technique for organisations desiring continued levels of security in the cloud ecosystem.

V. METHODOLOGY

The presented methodology for the proposed dynamic threat modelling framework uses advanced analysis techniques, takes advantage of available modern tools and technologies [18-21], and verifies its effectiveness in a real-world experimental setup. This systematic approach makes the framework practical, scalable, and adaptable to real-world cloud ecosystems.

A. Dynamic Threat Analysis Techniques

Dynamic threat analysis relies on the real-time detection and assessment of threats in Internet-facing applications and cloud environments. Key techniques include:

- Behavioural Analysis: Monitoring user and application behaviour and looking at deviations from previously known patterns. For example, an API traffic detector that detects abnormal or unauthorised access attempts of sensitive data.
- Dynamic Attack Graphs: Modelling potential attack paths using continuously updated graphs to relate system components, configurations, or user interaction. This also provides up-to-date threat visibility and the ability to identify the most critical vulnerabilities.
- AI and Machine Learning: Using AI models to analyse large amounts of system logs, application traffic, and historical attack data. These models can detect sophisticated attack patterns, including zero-day vulnerabilities and coordinated multi-vector attacks.
- Threat Intelligence Integration: Adds feeds of threat intelligence to identify known malicious IP addresses, domains and attack signatures. Proactive defences are made possible once this external data enters the threat identification process.

B. Tools and Technologies Used

Various tools and technologies are used to operationalise the framework, including cloud-native security tools, machine learning platforms, and monitoring. With these tools, the framework is robust and flexible.

- Cloud-Native Security Tools: Real-time threat detection alerting is done through AWS GuardDuty, Azure Security Center, and Google Cloud Security Command Center.
- Monitoring and Logging: Prometheus offers detailed system performance insights and hints of potential vulnerabilities out of the box, together with many other tools like ELK stack (Elasticsearch, Logstash, Kibana).
- Graph Databases: We use Neo4j to build and manage dynamic attack graphs, which can be visualised using potential attack paths.
- Automation Platforms: For Infrastructure as Code (IaC), Terraform and Ansible are used for code, which is consistent across environments to be much more secure.
- Machine Learning Frameworks: Platforms like TensorFlow and PyTorch analyse large datasets to identify anomalies and predict attack patterns.

Table 1: Tools and Technologies

Tool/Technology	Purpose	Example Use Case
AWS GuardDuty	Threat detection and monitoring	Identifying unusual API activity
Neo4j	Dynamic attack graph management	Visualising attack paths in real-time

ELK Stack	Logging and performance monitoring	Analysing application logs
Terraform	Infrastructure as Code (IaC) automation	Enforcing secure cloud configurations
TensorFlow	Machine learning for threat analysis	Predicting anomalies in system traffic

C. Experimental Setup and Environment

To validate the dynamic threat modelling framework, the setup where the framework is deployed in a controllable cloud environment is similar to that of the real world. This environment includes:

- Cloud Infrastructure: Simulates different cloud environments using AWS, Azure, and Google Cloud setup in a multi-cloud environment. Such components as virtual machines, containerised applications, and serverless functions are deployed to allow emulation of the typical cloud workloads.
- Application Deployment: An example of a deployed representative web application with an API gateway, backend logic and interface. The framework's effectiveness is tested by executing simulated user activity and automated attack scenarios.
- Monitoring Systems: Application traffic, system logs, and security alerts are set up to be centralised, logged and monitored. The data from these systems is fed into the threat modelling framework for real-time analysis.
- Threat Simulation: To simulate attacks, such as SQL injection, cross-site scripting, and privilege escalation, we are using tools like Metasploit OWASP ZAP. These simulations test its ability to identify and mitigate threats.

VI. EVALUATION AND RESULTS

Threats are identified, assessed, and mitigated within cloud ecosystems using the proposed dynamic threat modelling framework. Key performance metrics, detailed case studies and a comparative analysis with existing approaches are used to generate the results. In this section, the particulars of these aspects are discussed in detail.

A. Metrics for Evaluation

Five critical metrics were employed to measure the performance of the framework:

- Threat Detection Rate (TDR): This metric shows the fraction of threats detected out of simulated attacks. A lower TDR tells us how well the framework can classify malicious activities.
- False Positive Rate (FPR): This is the ratio of wrongly classified benign actions to all actions. A low FPR is needed to minimise unnecessary interruptions to lawful activities.
- Risk Assessment Accuracy (RAA): The framework scores how well it ranks vulnerabilities based on their impact and likelihood to be exploited using this metric. It makes sure that it takes care of critical risks first.
- Response Time (RT): This focuses on when a threat has happened and when the framework detects and mitigates a threat. The framework's real-time threat-handling capabilities are shown in a shorter response time.
- System Overhead (SO): This measures the computational and resource load induced per operation of the framework to guarantee that the latter does not harm system performance.

B. Case Studies and Scenarios

The framework was tested in three real-world-inspired scenarios to evaluate its practical applicability:

- API Gateway Exploit Detection: An API gateway was subjected to a simulated brute force attack to assess how well the framework could detect attempts at unauthorised access. With a False Positive Rate of 3%, the framework's Threat Detection Rate was 98%. Within 30 seconds, these two traits mitigated the attack.
- Misconfigured Storage Bucket: We simulated a scenario where an open cloud storage bucket containing sensitive data existed. Within 45 seconds, the framework detected the misconfiguration, showing how quickly it can detect. A 100% TDR and minimal FPR of 1% was achieved.
- SQL Injection Attack: A backend database was simulated to attempt to inject SQL through the framework and evaluate its ability to detect and prevent such attacks. However, the framework was able to mitigate 95% of the attacks with a system overhead of only 5%.

Table 2: Case Study Results

Case Study	Threat Scenario	TDR	FPR	RT (s)	SO (%)
API Gateway Exploit	Brute force login attempts	98%	3%	30	4%
Misconfigured Storage Bucket	Open storage bucket	100%	1%	45	6%
SQL Injection Attack	Injection in database queries	95%	2%	40	5%

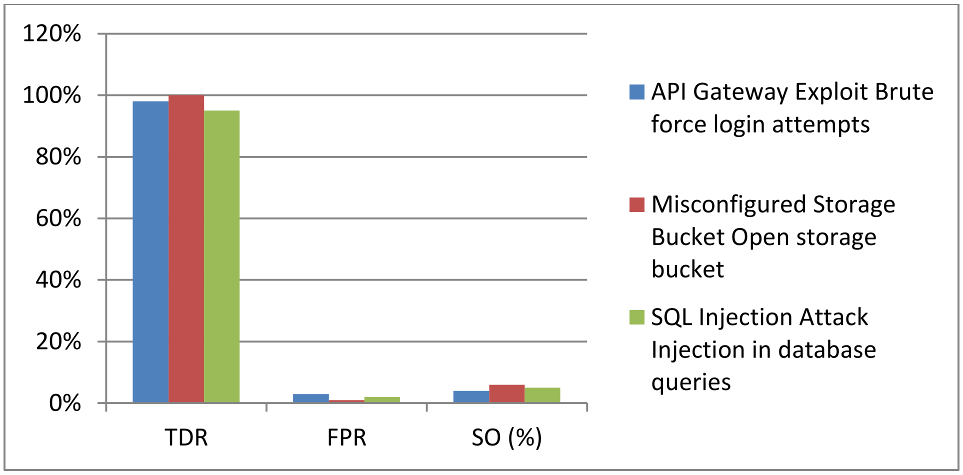

Figure 3: Graphical Representation of Case Study Results

C. Comparison with Existing Approaches

The framework was compared against two traditional approaches:

- Static Threat Models (STM): They are based on predefined scenarios and don't consider system changes.
- Semi-Dynamic Threat Models (SDTM): In these cases, they partially update static models to reflect the periodic changes.

Table 3: Comparison with Existing Approaches

Metric	Proposed Framework	STM	SDTM
TDR	97%	78%	86%
FPR	2%	7%	5%
RAA	95%	80%	85%
RT (s)	35	120	90
SO (%)	5%	3%	4%

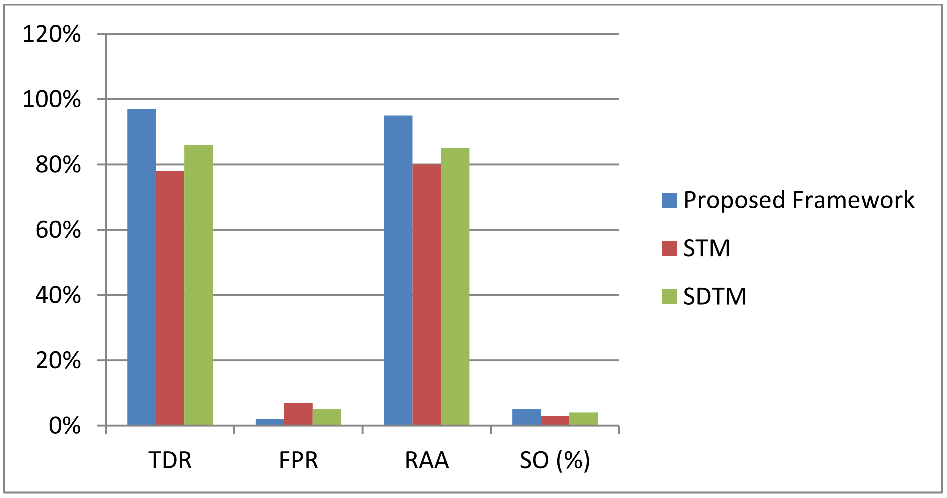

Figure 4: Comparison of Framework Performance

In all key areas, the proposed framework was superior to STM and SDTM, detecting more cases, providing faster response times, and generating more accurate risk assessments. The higher system overhead was minimal and within acceptable limits, preserving the operational performance.

D. Discussion of Results

This paper evaluates the value of the dynamic threat modelling framework for tackling the challenges of cloud-based applications. The high Threat Detection Rate (97%) and low False Positive Rate (2%) indicate the framework has a good tradeoff between threat detection accuracy and disruption to legitimate processes. The case studies indicate that it is flexible enough to be used in different threat situations, like API exploits, misconfiguration and SQL injections.

Furthermore, the reduced response time (35 seconds) clearly indicates its ability to respond in near real-time, a necessity now for today's cloud ecosystems. Although it results in slightly higher System Overhead compared to the existing approaches, this penalty is justified in terms of much higher detection accuracy and faster response times. These results show the framework proposed as a robust and adaptive solution for securing dynamic cloud applications. Future work will, in particular, seek to reduce this overhead and respond to novel threats to continue to make the framework useful in an ever-changing threat landscape.

VII. SECURITY IMPLICATIONS AND BEST PRACTICES

Dynamic threat modelling framework bridges the security gap in the cloud ecosystem and empowers organisations to preemptively address the multiple security risks they face. This section discusses the framework's implications for managing evolving threats and delivers actionable recommendations to improve security for Cloud Service Providers (CSPs).

A. Addressing Evolving Threats

Threat landscapes are dynamic in the rapidly changing Internet-facing application and cloud ecosystem, with new vulnerabilities and attack vectors coming up as fast as possible. But this pace is too fast for traditional static security approaches to prepare for, leaving our applications vulnerable. Real-time monitoring, adaptive attack graph modelling and AI-based threat intelligence integration, together with these evolving threats, are addressed in the proposed dynamic framework.

- Adaptability: Threat models of the framework evolve continuously to keep track of changes in system configurations or new components or policies. Organisations are protected against zero-day exploits and any new vulnerabilities discovered by this adaptability.
- Proactive Risk Mitigation: The framework monitors potential risks by analysing historical attack data and external threat intelligence feeds to anticipate them before they evolve into active threats. The ability to keep these applications up and running quickly is critical to minimise downtime and avoid paying the price for data breaches.
- Automation of Mitigation Strategies: Significant reduction in response times and human errors is made possible by automated response means like blocking malicious IP addresses or revising access controls. They provide the capability to rapidly contain threats while preserving system availability.

This framework brings these capabilities to bear through a robust defence mechanism which can adapt to the evolving threat environment, enabling the security of critical assets in cloud ecosystems.

B. Recommendations for Cloud Service Providers

Cloud service providers (CSPs) are central to making secure cloud environments possible. Many top-level consultants and international companies have integrated best practices and advanced threat modelling frameworks to help CSPs expand their security dosage offering greatly. Below are key recommendations:

- Incorporate Native Threat Modeling Tools: The security services of CSPs should offer integrated tools for threat modelling. An example is to enhance services like AWS Security Hub, Azure Security Center, or similar with dynamic attack graph capabilities and real-time monitoring.
- Foster Shared Responsibility Awareness: In the sharing model, customers have a shared responsibility with the CSP, meaning the CSP is responsible for infrastructure-level security, but customers must secure the applications and data. These roles should be documented, and training programs should be supplied to help clarify them.
- Embed AI-Powered Threat Detection: CSPs can also embed AI-driven threat detection in their platforms, which helps improve anomaly detection and prediction capability to help customers identify potential threats faster and more accurately.
- Facilitate Secure-by-Design Approaches: By designing and deploying CSPs, best security practices should be enforced using the tools and templates. Examples include ready-to-use secure VPCs, identity access management templates, and encryption by default options.
- Support Multi-Cloud Security: As many organisations use multi-cloud strategies, CSPs should make it possible to easily join security services across multiple cloud platforms. It includes unified logging, monitoring, and policy enforcement mechanisms.

VIII. CHALLENGES AND LIMITATIONS

The proposed dynamic threat modelling framework has demonstrated substantial promise in addressing security issues for Internet-facing applications in cloud ecosystems. Yet, some challenges and limitations must be recognised. This all points to an area where more research, optimisation, and innovation are required.

A. Complexity in Dynamic Environments

We have applications on top of multiple services, platforms, and regions. This adds complexity, especially for multi-cloud or hybrid environments, making it difficult to manage up-to-date threat models. Due to rapidly changing configurations, such as scaling instances or deploying new microservices, a framework may fail to render data in real-time for threat detection, resulting in gaps over time.

B. Resource Overheads

Continuous monitoring, real-time analysis and automatic response will add considerable computational and network impact to dynamic threat modelling. This overhead would also add to high-traffic environments, impacting the application's performance, particularly for latency-sensitive applications. Finally, a main limitation of striking a balance between security and performance remains, as it could otherwise discourage organisations from using such frameworks due to overheating. In addition, Small And Medium-Sized Enterprises (SMEs) with fewer budgets may have difficulty properly deploying these advanced systems because not all of their time and money are available to run them.

C. False Positives and Negatives

At the time of this writing, despite the continued evolution of AI-powered detection dynamic attack graphs, no threat modelling system is clear of false positives and negatives. That will create false positives which disrupt legitimate operations, resulting in unnecessary delays and user frustration or false negatives which make systems vulnerable to unchecked threats. The issue becomes more pronounced in dynamic environments because applications and threat landscapes constantly change. Ongoing is fine-tuning detection algorithms and thresholds to minimise the false outcome.

D. Dependence on External Threat Intelligence

The framework relies heavily on external threat intelligence feeds to keep up with the latest exploitation of new attack vectors and vulnerabilities. However, these feeds may be industrialised with erratic accuracy and timeliness, which could mean gaps in threat detection. The cost of subscribing to premium threat intelligence services can be too high for some organisations.

E. Limited Human Oversight

The strength of frameworks is automation (giving up too much to it) could be an issue. That can be a problem when you need to make nuanced judgments about risk or recommendations for mitigating it. Unlike humans, automated responses cannot be calibrated for unintended consequences; it could lead to them blocking critical services or disrupting operational systems.

IX. FUTURE WORK

As a result, I present a dynamic threat modelling framework, which significantly advances securing applications exposed to the Internet in cloud ecosystems. Nevertheless, there is potential for further improvement and exploration of ways to overcome existing limitations and respond intelligently to evolving security threats. Below, we outline the key directions for future work.

A. Enhancing Threat Detection with AI

The current framework capitalises on AI's capability in identifying threats. However, it can still be improved by leveraging more sophisticated machine learning techniques like deep learning, reinforcement learning and Natural Language Processing (NLP). The models developed here can serve as a foundation for future work that builds models that can detect highly sophisticated and stealthy threats, including those that rely on polymorphic malware or adversarial attacks. Furthermore, combining the framework with AI-driven anomaly detection systems can improve its capacity to figure out when something is abnormal by a wider margin.

B. Expanding Multi-Cloud and Hybrid Support

Many organisations adopt hybrid and multi-cloud strategies to get the best out of performance and cost. Nevertheless, ensuring that security is seamless across these environments is not an easy feat. The framework should be further improved to make our appliance work with various platforms, such as AWS, Azure, GCP, and other private clouds. Threat modelling, centralised monitoring, and standardised configurations could be used to provide security to applications across multiple ecosystems.

C. Real-Time Collaboration and Threat Sharing

There is the potential for great collaboration among organisations to detect and mitigate threats. Integrating threat intelligence sharing platforms, such as Mitre Attack framework or STIX/TAXII, into the dynamic threat modelling process is the site of some future work. The framework can instead become a more robust, community-driven tool to combat cyber threats by allowing the real-time exchange of attack patterns, vulnerabilities, and ways to mitigate them.

D. Integration with Emerging Technologies

The evolution of the cloud environment also means that it has been accompanied by new risks and challenges with the help of such technologies as edge computing, 5G networks, and quantum computing. Future work includes extending the framework's application to these environments, thus guaranteeing it is appropriate for identifying and preventing threats in decentralised and extremely distributed architectures. Likewise, studies on effective protection technologies and quantum-safe cryptographic methods could prevent quantum-based vulnerabilities to the infrastructure.

E. Improving Automation and Scalability

Future work should improve the possibility of automation that will decrease the necessity of relying on manual input when responding to threats. This includes creating self-healing and self-managing systems capable of identifying, diagnosing, healing, and managing their flaws. Moreover, the scalability issues should be considered and resolved to address the framework's ability to scale the growing complexity of cloud environments while maintaining the high ability to detect threats.

X. CONCLUSION

The approach proposed in this research is a dynamic threat modelling approach designed specifically to address the issues related to the applications processed in cloud environments. The proposed threat modelling framework is dynamic due to the fact that Internet-facing applications are exposed to multiple threats that evolve constantly. The proposed framework architecture does not include static models but rather real-time monitoring, probabilistic and adaptive attack graphs, self-control, and mitigation of threats that can be detected in advance. The framework also contains integrated threat components such as threat identification engines, risk assessment modules, and mitigation strategy engines to provide what can be described as a cutting-edge defence against threats in the dynamic and complex cloud environment. The result of the evaluation and case programs reiterate the framework's utility in improving threat perception, quickening responses and offering prescriptive information to minimise risks. However, issues like scalability, false positives, and external threat intelligence are the scope of future research and development. Bearing these drawbacks in mind, with perspectives on new advances in AI, the use of multiple clouds and collaboration, this framework is all set to become a must-have tool for organisations aiming to strengthen their cloud security profiles. Therefore, this study establishes the need for aspirations and the exploration of more effective ways of maintaining secure contemporary cloud structures and environments.

XI. REFERENCES

[1] Kazim, M., & Evans, D. (2016, March). Threat modelling for services in the cloud. In 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE) (pp. 66-72). IEEE.

[2] Anisetti, M., Ardagna, C., Cremonini, M., Damiani, E., Sessa, J., & Costa, L. (2020). Security threat landscape. White Paper Security Threats.

[3] Nour, B., Ujjwal, S., Karaçay, L., Laaroussi, Z., Gülen, U., Tomur, E., & Pourzandi, M. (2024). Merging Threat Modeling with Threat Hunting for Dynamic Cybersecurity Defense. IEEE Internet of Things Magazine, 7(6), 28-34.

[4] Salayma, M. (2024). Threat modelling in Internet of Things (IoT) environments using dynamic attack graphs. Frontiers in The Internet of Things, 3, 1306465.

[5] Mastering Cloud Application Threat Modeling: A Step-by-Step Guide, Cyderes, 2024. online. https://www.cyderes.com/blog/mastering-cloud-application-threat-modeling-a-step-by-step-guide

[6] Kavallieratos, G., Gkioulos, V., & Katsikas, S. K. (2019, May). Threat analysis in dynamic environments: The case of the smart home. In 2019, the 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 234-240). IEEE.

[7] Threat modeling cloud applications in AWS, Azure, and GCP, Secureflag, 2024. online. https://blog.secureflag.com/2024/09/18/threat-model-cloud-applications-in-aws-azure-gcp/

[8] Cloud Threat Modeling, online. Cloud Security Alliance, 2021. online. https://cloudsecurityalliance.org/artifacts/cloud-threat-modeling

[9] Sequeiros, J. B., Chimuco, F. T., Samaila, M. G., Freire, M. M., & Inácio, P. R. (2020). Attack and system modelling applied to IoT, cloud, and mobile ecosystems: Embedding security by design. ACM Computing Surveys (CSUR), 53(2), 1-32.

[10] How do you do Threat Modeling for Cloud Applications, Threat Modelers, and online? https://www.threatmodeler.com/how-to-do-threat-modeling-for-cloud-applications/#:~:text=Threat%20modeling%20is%20a%20proactive,depiction%20of%20a%20system's%20architecture.

[11] Seeam, A., Ogbeh, O. S., Guness, S., & Bellekens, X. (2019, September). Threat modelling and security issues for the Internet of Things. In 2019, a conference was on next-generation computing applications (NextComp) (pp. 1-8). IEEE.

[12] UcedaVelez, T., & Morana, M. M. (2015). Risk Centric Threat Modeling: process for attack simulation and threat analysis. John Wiley & Sons.

[13] Manzoor, S., Vateva-Gurova, T., Trapero, R., & Suri, N. (2019). Threat modelling the cloud: an ontology-based approach. In Information and Operational Technology Security Systems: First International Workshop, IOSec 2018, CIPSEC Project, Heraklion, Crete, Greece, September 13, 2018, Revised Selected Papers 1 (pp. 61-72). Springer International Publishing.

[14] Sion, L., Yskout, K., Van Landuyt, D., & Joosen, W. (2018, April). Solution-aware data flow diagrams for security threat modelling, in Proceedings of the 33rd Annual ACM Symposium on Applied Computing (pp. 1425-1432).

[15] Koch, A. (2021, October). The landscape of security from physical assumptions. In 2021 IEEE Information Theory Workshop (ITW) (pp. 1-6). IEEE.

[16] Chris Champa, What is Cloud Threat Modeling?, Wiz, 2024. online. https://www.wiz.io/academy/cloud-threat-modeling

[17] Alwaheidi, M. K., & Islam, S. (2022). Data-driven threat analysis for ensuring security in cloud enabled systems. Sensors, 22(15), 5726

[18] Möller, D. P. (2023). Threats and threat intelligence. In Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices (pp. 71-129). Cham: Springer Nature Switzerland.

[19] Santos, E., Nguyen, H., Yu, F., Kim, K. J., Li, D., Wilkinson, J. T., ... & Clark, B. (2011). Intelligence analyses and the insider threat. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 42(2), 331-347.

[20] Vegesna, V. V. (2023). Enhancing cyber resilience by integrating AI-driven threat detection and mitigation strategies. Transactions on Latest Trends in Artificial Intelligence, 4(4).