

Enhanced ML Based Security framework for Cloud Security

Presented

By

T V S Vishnu Vardhan (2023202021)

L Sai Chaitanya Reddy (2023202023)

V Subhash Chandra Bose (2023202012)

International Institute of Information Technology, Hyderabad

November 12, 2024

Outline

- 1 Introduction
- 2 Cloud Security
- 3 Challenges in Cloud Security
- 4 Access Control and Data Security
- 5 Network Security
- 6 Existing Frameworks
- 7 Proposed Framework
- 8 Drawbacks
- 9 Improvements
- 10 References

Introduction

- Cloud Computing - efficient data handling, processing and storage
- Benefits
 - ▶ Resource Scalability
 - ▶ Cost Effectiveness
 - ▶ Availability of Resources
- However, this also brings some security issues, security of sensitive data and management
- Key management will have creation, distribution, revoking and storage of the required keys for cryptographic algorithms
- So, key management is important for overall security
- Compromising of key management leads to data leaks, accessing of sensitive information and leakage of important information

Introduction

- This is a major challenge for storing sensitive information like bank information and medical records.
- Cloud Security should protect information from attacks like
 - ▶ Intrusion
 - ▶ Data leakage
 - ▶ Data Loss
 - ▶ Internal and external threats
 - ▶ Attacks on infrastructure, applications and data
- Cloud security applications - typically deployed as SaaS
- SaaS Layer provides security

Cloud Security

- This work - Enhances security and integrity between application and KMS
- Uses Machine Learning and actively detect any anomaly and detects baseline deviation
- Guarantees security by reducing probability of unauthorized access and security issues

Challenges in Cloud Security

- Many cloud platforms - Each has unique system for secure cloud
- This leads to minimum standardization across different systems
- Also, ensuring accuracy of existing policies and checking if they align with actual enforcement is a big challenge

Access Control and Data Security

- Fundamental aspect of cloud security
- Amazon - AWS IAM Systems - Commonly used for authentication and authorisation, which enforces policies and manage privileges
- Similarly, Google has IAM, Azure has RBAC, Kubernetes has its own RBAC, etc.
- Another important aspect - Data Security
- Encryption methods - Used when storing and transfer of data

Network Security

- Network security - implemented to protect from network attacks and data interception
- Uses techniques like Firewalls, VPNs, Intrusion Detection Systems
- For efficient management of cloud security, organisations uses SIEM (Security Information and Event Management) Systems
- SIEM Systems gather and access information and logs them
- Helps in detection and responding to any security issues

Existing Frameworks

- CSA
 - ▶ Offers thorough recommendations and best practices
 - ▶ Addresses several aspects like Data protection, identity, access control and incident handling
- NIST
 - ▶ Has a special article published
 - ▶ Users can use security controls and suggestions provided in article
 - ▶ Addresses aspects like key management, access control, encryption
- TCI
- PCI DSS
 - ▶ Offers security requirements for handling Credit Card Information in cloud

Proposed Framework

- KMS - Common Component
- Normally uses envelope encryption for key management
- In development environment, server-side applications are built through VMs and deployed on cloud
- In building mobile application, same environment is used, which makes client side REST API calls to server-side apps hosting REST APIs
- Mobile apps - Can be any business apps like Amazon, Flipkart, etc.

Continued....

- Permission Detection Engine
 - ▶ Evaluates permissions and verifies the validity
- Applications - deployed on different mobile devices
- When apps are used by users,
 - ▶ User authentication
 - ▶ REST API Calls to server-side applications on cloud VMs, securely using OAuth2.0 based protocol and HTTPS Layer security
- Here, KMS Layer acts as key vault, client and server gets required keys from KMS
- Cloud directory acts as OAuth2.0 authorization server

Continued....

- ML will be used to detect outliers during interactions of apps with KMS
- Log data from KMS is stored in a big data repository
- For each user that interacts with KMS, will be placed in clusters
- Checks will be made on a scheduled basis for any deviation using an algorithm like K Means
- If any anomaly, the user will be blocked from using KMS

User Authentication

- Initially, when a user registers, a pair of keys, Public Key (P0) and Private key (P1) will be generated and associated with identity of user
- Public Key (P0): Retrieved and stored in the user's profile in the application's database
- Private Key (P1): Remains securely within AWS KMS and is never exposed or transmitted outside.
- The user profile includes reference to the private key stored in AWS KMS.
- This adds extra layer of security

Data encryption and Access Control

Data Encryption:

- For each data record:
 - ▶ DEK is generated using AWS KMS.
 - ▶ Data is encrypted using the DEK.
 - ▶ DEK is encrypted with a role-based CMK in AWS KMS.
 - ▶ Encrypted data and encrypted DEK are stored.

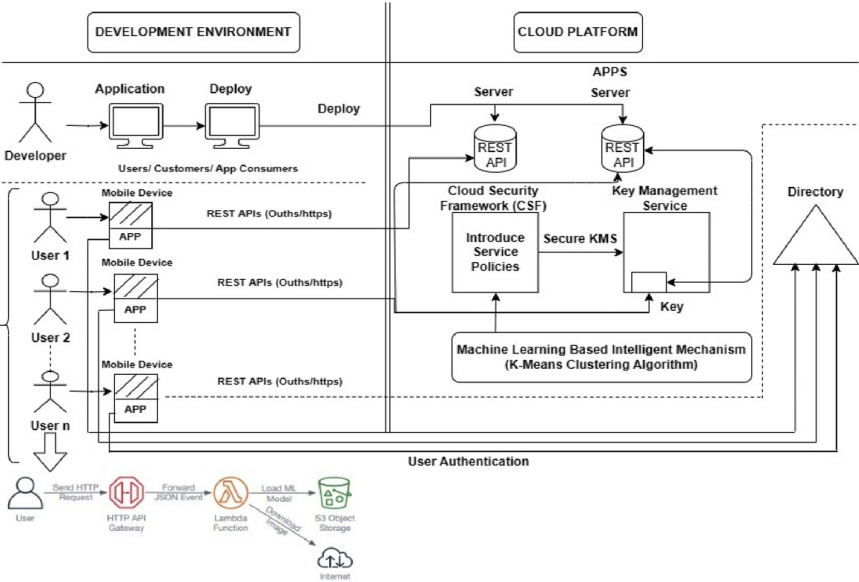
Data Access via REST API:

- User makes an API request with the access token.
- Server verifies the token and user permissions.
- Server decrypts the DEK using AWS KMS and the role-based CMK.
- Data is decrypted using the plaintext DEK.
- Decrypted data is returned to the user.

Flow in a Medical Use Case

- **Doctor's Application**
 - ▶ Login with Credentials
- **Authentication Server**
 - ▶ Access Control Step
 - ▶ Generate and sign challenge with Doctor's private key
- **Doctor's Application**
 - ▶ Receive and verify signed challenge
- **Authentication Server**
 - ▶ Generate and sign JWT
- **Doctor's Application**
 - ▶ Receive and store JWT
 - ▶ Prepare and sign update request
- **Server**
 - ▶ Verify JWT, signature, and permissions
 - ▶ Decrypt, update, and re-encrypt, and store updated data

Framework



Drawbacks

- Scalability : This might not be scalable, as Lambda functions and KMS are used for Key management
- Offline Accessibility : Since there should be continuous interaction between client and server, there is no possibility of offline accessibility
- False Positives and False Negatives: These will be issue at a large scale
- Single Point of Failure : If any of the Lambda function or KMS has a breakdown, entire system will be disturbed

Improvements - Scalability

- Scalability : This might not be scalable, as Lambda functions and KMS are used for Key management
- Since Lambda function is causing issue, Amazon EC2 instances with auto-scaling groups can be used
- Based on active no. of users, the system will automatically scale
- This also allows more flexibility in handling high loads
- Can be customized with instance types which are optimized for some applications
- Elastic Kubernetes Service (EKS) service provided by Amazon also can be used for rapid scaling and isolation
- To avoid issue with KMS
 - ▶ Local caching of frequently accessed keys
 - ▶ Batch multiple calls in an API together and make a request, to reduce no. of requests to KMS

Improvements - Offline Access

- Offline access
- Implementation of local caching, where we store essential data needed for communications
- Amplify DataStore service can be used for this
- Synchronizes data whenever there is connection
- Another way is to implement for conditions with low bandwidth
- Although internet connectivity is needed, low bandwidth also works
- CloudFront Service can be used to implement this

Improvements - Single Point of Failure

- Single Point of Failure
- Hybrid Cloud Management System
- Establishing a backup KMS using another service provider
- Ex : Using Google KMS and AWS KMS
- Deploying the services in multiple regions, so we can switch to another region

Improvements

- Encrypting sensitive data all the time - during rest and transit
- Implementing multi factor authentication on top of cryptographic encryption to achieve more security
- K-Means algorithm also can be optimized for more efficiency and avoiding false positives and false negatives

References I

- [1] Sundararajan, E., Arumugam, A., Kumar, S.V.: A study on key management techniques in cloud computing. *Procedia Comput. Sci.* 115, 450–457 (2017)
- [2] Ashwath, A., Kumari, S., Kumar, R.: Security analysis of key management services in cloud computing. *Int. J. Comput. Sci. Inf. Security* 17(12), 53–59 (2019)
- [3] El Kafhali, S., El Mir, I., Hanini, M.: Security threats, defense mechanisms, challenges, and future directions in cloud computing. *Arch. Comput. Methods Eng.* 29(1), 223–246 (2022)
- [4] Rabbani, M., Wang, Y.L., Khoshkangini, R., Jelodar, H., Zhao, R., Hu, P.: A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing. *J. Netw. Comput. Appl.* 151, 102507 (2020)

Thank you!