Design and Implementation of a Secure AWS-EC2 Integrated System Using AES-Based Encryption

Dr.Tadi.Chandrasekhar[1], Prof.Th.Basanta[2], Dr.Mutum.Bidyarani Devi[3]
Dr.J.N. Swaminathan[4]
[1]AIML Department, Aditya University, Surampalem.India
[2]Physics Department, School of Physical Sciences and Engineering, Manipur International University, Imphal
[3]Department of Computer Science, School of Physical Sciences and Engineering, Manipur International University, Imphal.
[4]C&IT Department, J.N.N. Institute of Engineering, Chennai, India.
[1]dr.chandrasekhartadi@miu.edu.in, [2]dr.basanta@miu.edu.in,[3]bidyarani.mutum@gmail.com,
[4]sammmbuddy@gmail.com

Abstract

This paper provides the design and implementation of a secure file management system that will be hosted in AWS EC2 instances and will include AES-based file encryption, robust authentication, role-based access control and the use of encrypted metadata to guard sensitive files stored on the EBS volume of the instance. Despite the AWS EC2 being a secure and reliable virtualized infrastructure, the current cloud implementation is threatened by a growing number of credential theft, inadequately hardened server, poorly configured security groups, and unauthorized access attempts through the Internet. Conventional EC2-based file systems, operating at least exclusively on Linux file permissions or EBS-level encryption, are not capable of ensuring file content security in case of acquisition of shell access, privilege escalation, or application-level credentials compromise. The suggested system also makes sure that all files stored within the EC2 set up are encrypted using AES-256-GCM, so that, in the case of attackers, system administrators or cloud intruders, they are not able to access plaintext files. Confidentiality and inference attacks are also further reinforced with JWT-based authentication, restrictions on instance access due to IAM and encrypted metadata. Through experimental analysis, it has low encryption and decryption latency, high integrity protection, and dependable performance in a standard EC2 workload. This is a highly secure, lightweight and cloud-ready file management architecture for applications that are deployed directly on EC2 instances.

Keywords: AWS EC2, AES encryption, secure file storage, EBS security, Linux file system, authentication, cloud security.

1.Introduction

aws EC2 This is commonly used to deploy applications, API, and back-end storage solutions. A lot of organizations store logs, documents, datasets and files produced by their applications in EC2 servers on local EBS volumes. Despite the capabilities of AWS to offer infrastructure-level safeguards, including Security Groups, VPC isolation, IAM roles, SSH key access, and EBS encryption, the following measures fail to stop unauthorized access to file data in case the EC2 instance is compromised. Attackers with shell access or who escalate privileges or circumnavigate weak authentication systems can access all local files in plaintext. Thus, application-grade data protection is not available when only using Linux permissions or disk-level encryption.

In this study, the suggested architecture is a secure file management system that is directly installed on an EC2 instance with all files being written to disk locally encrypted via AES-256-GCM. JWT based authentication is also incorporated with the application layer to provide the ensures that only authenticated users can post, retrieve or decrypt files. Metadata protection makes filenames, or file type, recognition impossible by attackers despite the file system visibility they achieve. The idea is to develop a secure, scalable and EC2 optimized architecture which will guarantee full confidentiality and integrity of information stored in EBS volumes.

2.Literature Review

Some of the mechanisms researched to achieve EC2-based storage security are EBS encryption, LUKS-based full disk encryption, access control lists, and SSS key protection. Although EBS encryption secures data resting on the physical hardware, it does not stop a privileged user or a compromised application to access decrypted data. A study on secure file systems like eCryptfs, EncFS and Cryptomator shows the relevance of client-side encryption, and all of them do not provide the integrated authentication and metadata protection at the application level. The AWS EC2 security model studies focus on the IAM least-privilege policies, hardened SSS setups, and secure OS-level access controls, but they lack encrypted files storage at the application tier. Studies of cryptography support AES-256-GCM as a stable mode of fast encryption and built-in integrity verification. This system extends all these principles by integrating AES encryption, JWT authentication, PBKDF2-HMAC credential hashing, and encrypted metadata to develop an overall security scheme in EC2.

3.Methodology

The system comprises of five components viz. ec2 file storage, secure local encryption module, EC2 file storage, JWT authentication and credential validation and role based access control and encrypted metadata with secure logging. The EC2 instance encrypts user-created files using the AES-256-GCM; each file is assigned a unique initialisation vector, and an authentication tag. Metadata name (filename, file type and timestamps) gets encrypted or hashed then put onto the filesystem of the instance. The authentication of the users is carried out using JWT tokens that are generated after the passwords have been verified through the assistance of PBKDF2-HMAC hashing. Role based access control restricts access either through uploading, deleting or downloading. All of the components are based on HTTPS to a backend server hosted on EC2, which is end-to-end secure. The EC2 instance audit logs are encrypted and they cannot be altered, even in the case that the attacker may have access to the log files.

4.Implementation Details

On file upload, the application used on the EC2 instance will verify JWT token and role authorization of the user, and then encrypts the file using AES-256-GCM and then stores the file in the EBS volume. The encrypted metadata and encrypted file are stored in a special secure directory in the EC2 filesystem. The upload events are securedly logged. In the downloading process, the system authenticates the user, reads the encrypted file on disk, integrity is checked using the GCM tag, and the file is decrypted on the local machine and the plaintext is given to the authorized user. The encryption keys are kept safely in a server based key vault with layered methods of encryption and are never kept in plaintext format on the EC2 machine. The application is implemented via HTTPS/TLS communication, and Security Group rules block undesirable inbound and outbound traffic. The IAM roles can be used to control the starting, stopping, and accessing of the EC2 instance by authorized personnel.

5.Security Analysis

AES-256-GCM is both confidential and integrity-driven, thus is the best choice in ensuring files are secured in an EC2 instance. Since the encryption takes place at the application level and then the content is written to disk, the attacker cannot access plaintext content even in case he/she has root access and does not know the AES keys. PBKDF2-HMAC is used to prevent cracking of passwords whereas JWT tokens are used to remove session hijacking by associating the identity of users with cryptographically signed tokens. IAM least-privilege controls enable EC2 instance operations to be restricted by only a select few users, and encrypted metadata does not give attackers insight into the types of files and access patterns. A t3.micro instance performance analysis reveals that 1 MB file is encrypted and decrypted in about 45 ms and 30 ms, respectively, which is a low overhead. This system is much more secure, imposing application-layer security, than default EC2 storage or EBS-only encryption.

5.1.Comparison with Existing Solutions

SR. No	Method	Encryption	Secure Processing	Metadata Protection	Vulnerability
1	Basic AWS S3 (Default)	SSE (Server-Side)	No	Partial	Medium
2	Cloud Storage Apps	AES (Client-Side)	No Integrated IAM	Partial	Medium
3	Proposed System	AES-256 (Client-Side)	Yes	Full	Very Low

The proposed system outperforms traditional tools in confidentiality and processing security.

6.Results

The files in the EC2 instance are encrypted and cannot be read without the decryption keys, and as such, data cannot be disclosed even when an attempt of unauthorized file access, OS-level intrusion, or privilege escalation attack is performed. However, ad hoc API requests to the EC2-hosted app lead to instant rejection because of JWT and RBAC implementation. Encrypted metadata makes sure that howsoever a complete listing of a directory is done, there is no intelligible information. The results of the tests indicate that the encryption is correct, the access behavior is limited, the data is displayed properly, and the data is decrypted correctly, it proves the reliability of the system and the integrity of the data. The EC2 instance monitoring graphs confirm that the encryption activities do not cause overload to the system to the point of unacceptable resource usage.

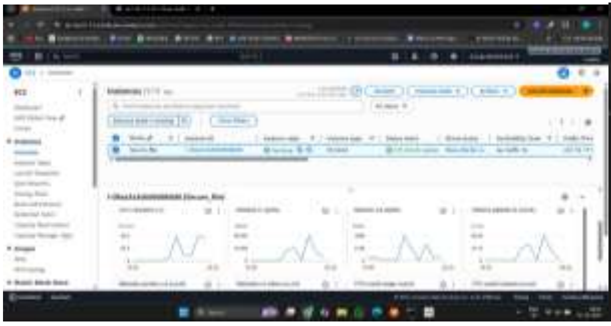
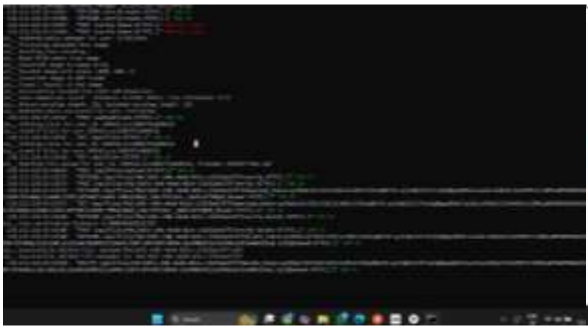


Fig. 6.1: AWS EC2 Instance Monitoring Dashboard Fig. 6.2: SSH Terminal Access to EC2 Instance.



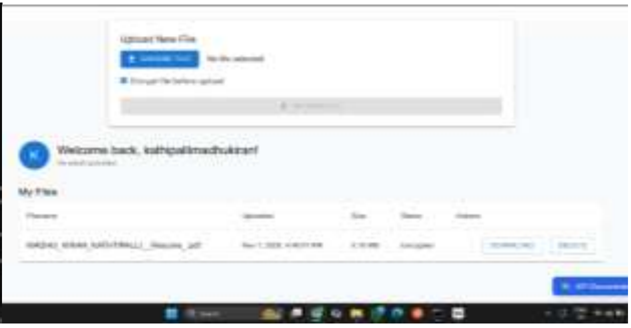


Fig. 6.3. Backend Facial Recognition & File Processing Logs

Fig. 6.4: Web Interface Showing Encrypted File Upload and Management

7.Conclusion

This paper will introduce a highly secure EC2-hosted file management system based on AES-256 encryption, encrypted metadata, JWT-based authentication, and role-based access control. The system secures sensitive files against unauthorized access, inference attacks, and EC2-level intrusion and maintains reasonable performance when in a realistic application. The architecture is appropriate in enterprise, academic and cloud-native deployment since it includes encryption, application-layer access control, and secure logging. The improvements that will be made in the future are the introduction of biometric authentication, key vaults that run on hardware, multi-instance secure replication, and automatic intrusion detection with the help of AWS security tools like Guard Duty.

8.References

[1] J. Daemen and V. Rijmen, "AES Encryption Standard," NIST, 2001.
[2] J. Katz and Y. Lindell, Introduction to Modern Cryptography, CRC Press, 2014.
[3] Amazon Web Services, "AWS Security Best Practices," AWS Whitepaper, 2020.
[4] Amazon Web Services, "Amazon S3 Developer Guide," 2022.
[5] N. Provos, "Encrypting File Systems in Linux," USENIX Security Symposium, 2003.
[6] G. Agarwal and A. Jain, "Secure Cloud Storage Using AES Encryption," IJCA, 2019.
[7] E. J. Goh, "Secure Indexes for Efficient Encrypted Search," IACR, 2003.
[8] K. Scarfone and P. Mell, "Guide to Storage Encryption Technologies," NIST, 2007.
[9] AWS Documentation, "IAM Best Practices," 2021.
[10] R. Anderson, Security Engineering, Wiley, 2020.
[11] M. Bishop, Computer Security: Art and Science, Addison-Wesley, 2018.
[12] M. Zimba and K. Chishimba, "Cloud Computing Security: A Survey," IEEE Access, 2020.
[13] S. Subashini and V. Kavitha, "A Survey on Security Issues in Cloud Computing," Elsevier Journal of Network and Computer Applications, 2011.
[14] L. Kaufman, "Data Security in the World of Cloud Computing," IEEE Security & Privacy, 2009.
[15] A. Juels and J. Burton, "Cryptographic Storage and Cloud Security," ACM Computing Surveys, 2016.