1.

Hi all, my name is Kevin and this is an overview on Lazard's data protection and access control practices

2.

Data storage and security is done in 3 components, amazon RDS, S3 and KMS. RDS is a managed data base service and it is how we create, manage and operate in our data bases.

Amazon S3 is what holds the actual objects. objects could be documents or images for an application. These objects are organized in buckets.

For security, encryption and decryption is done with amazon KMS which is a key management service. KMS allows you to choose between an amazon given key or customer managed key and these keys can be attached to S3 buckets and RDS databases.

3.

Another measure for security is data classification. Lazard has 4 different classifications for its data. There is public, internal, confidential, and restricted data. These classifications are set based on the content of the data and the effect it could have on the company if it were to be leaked.

Public data is classified as information that is available to be shared with the general public. This could be annual reports interviews with news media, business cards or information posted on lazards public website.

Internal data is data that is required to perform normal day to day work and can only be accessed by lazard personnel. This includes operating procedures, policies and interoffice memos, job titles, and departmental structures.

Data is classified as confidential if in the event of unauthorized disclosure, compromise, or destruction, it would have a direct or indirect impact on Lazard. This includes shareholder personal information, internal and external audit reports, and balance sheets.

Finally, Restricted data is sensitive data that is intended for only a limited group of individuals. This includes strategic planning information, information on mergers, and financial forecasts or results.

4.

When data is secured and protected properly, the next step is to decide who can actually access the data and who can't. This is done with access control.

The corporate environment has access control on 3 areas, the AWS components, the data, and the application. This is done using SCP policies, IAM Roles, bucket and key policies, and ACL controls.

5.

SCP and IAM is used to have access control over AWS components. SCP are organization policies, or parent level policies for AWS accounts. They allow or deny global access to AWS services.

IAM roles are used to manage specific roles and assign them to users. These roles can be attached to users, lambda functions, databases and more. The policies that are created can either be given by AWS or be customer managed.

SCP and IAM work hand and hand because IAM grants permission for particular roles to gain access, while SCP is an extra layer of security that can overrule the permissions set on IAM roles.

6.

For the data, there are bucket and key policies to protect the data and assets. Bucket policies are used to restrict access to S3 buckets and enforce security requirements. The Key policy is used to control access to encryption keys. Keys can be provided by a bucket policy to decrypt and encrypt an S3 bucket.

7. Finally, on the application, there are ACL control in place to grant users access to applications and its contents. ACL controls are defined by application groups, and groups are used to give granular access within an application.

1.

Hi everyone, here is an overview of the security measures in place on corporate applications.

2.

The corporate applications are split in 2 parts with a frontend and backend. The way that they communicate with each other is through an API that allows user requests to be taken and information gathered to deliver the info.

3.

The application layer API plays a key role in communication between the user interface and the backend.

The API takes in ACL groups and roles to display the applications that the users are allowed to access using a dashboard and menu. APIs are designed to by business requirements and dictates what applications are shown on the dashboard to a user.

4.

Another security measure on our applications Is that logging and monitoring happens on all activity for an application. The logs are centralized in sumo logic where it can be flagged for malicious activity and investigated by infosec.

5.

security is considered from the start of development. An application is developed in 4 stages, dev, test, staging and production. Each step tests the functionality of the code as well as its effect on other components in the application.

For each stage, we use Jenkins to run a deployment pipeline that makes sure the code is compliant with business requirements.

Within the pipeline, 3 security scans on done using Veracode, Sonarqube and Quality gateway to ensure secure coding and mitigate vulnerabilities from happening.

1.

Hi all, AI has exploded in popularity, and implementing it into an environment requires secure and proper implementation is necessary for a functional model.

2.

The corporate team has been developing a chatbot named Vera. In order to use this chatbot in their environment, they have been working to securely implement a ML mechanism that can connect with Lazard's internal data.

One security measure that is in place is that the model will sit in a private environment, only able to communicate within Lazard's internal data. Hosting the model on a private network helps protect the model from unauthorized access, data breaches, and other forms of cyber-attacks.

Another security measure that is in place is how you can interact with the model. The only way to connect to the model is through the SPFx user interface.

3.

Here is a diagram showing how the chatbot works starting from the user. The user can seamlessly execute tasks in multiple services following orchestrated business processes.

This input is passed through cloudfront and sent to the vera mechanism over HTTPS. From here, Vera gathers the data from its secure area and sends the information back to the user. All the information received and sent out is secured with HTTPS communication.

The communication between user, model, and data are all secured which protect the data while in transit.

4.

Other security measures that take protect the model are practices already in place but expanded on.

For example, one recommended practice is to limit the model's capabilities to only be functional. One way to do this is by implementing a human in the loop control. An example of this could be allowing the model to build a help desk ticket for you, but you have to send it out, the model cannot.

Another recommended practice is to have secure coding and proper input validation and sanitization. This should be used to help filter out malicious inputs and help validate training data and responses from the model.