Microosft Security, Compliacne and Identity Fundamentals: Describe the concepts of security, compliance, and identity

Shared responsibility module

shared-responsibility-model-graphic

- On-premises datacenters: customer is responsible for everything from physical security to encrypting data properly
- Infrastructure as a Service (IaaS): Most responsiblity still lies with customer. Although they are using the providers infra, they are responsible for software, os, network, applications and protecting data
- Platform as a Service (PaaS): PaaS provides a platform for building, testing, and deploying software quickly. The customer is still responsible for applications and data.
- Software as a Service (SaaS): The least amount of responsibility for the customer - everything is taken care of except the data, devices, accounts, and identities.
-

Defence in depth

A layered approach to security - if one layer gets breached, the next will (ideally) not allow any further access.

Examples of layers of security are:

- **Physical** - limiting access to a datacenter or building to only valid personnel
- **Identity and access** - security controls such as MFA and conditional access to make any changes
- **Perimeter security** of your network to protect against DDos and large scale attacks.
- **Network security** - segmentation, access controls, limiting communications between resources
- **Computer layer security** - securing access to virtual machines on-prem or in the cloud by closing certain ports.

- **Application layer security** - ensuring applications are secure and do not have security vulnerabilities.
- **Data layer security** - controls to manage acces to business and customer data, with encryption to protect it.

Confidentiality, Integrity, Availability (CIA)

The above layers are all in the aim of ensuring confidentiality, integrity, and availability, which is referred to as CIA:

- Confidentiality - keeping confidential data (customer, passwords, financial) encrypted.
- Integrity - keeping data properly stored and accurate and ensuring that is has not been tampered with or altered.
- Availability - needs to be available to those who need it.

The Zero Trust model

Zero Trust assumes that everything is on an open and untrusted network - even resources behind firewalls, operating on the principle of "trust no one, verify everything".

The Zero Trust guiding principles

- Verify explicitly - Always authenticate and authorize based on available data points (user identity, location, device, service or workload, data classification, anomalies)
- Least privileged access - Limit user access with just-in-time and just-enough-access (JIT/JEA), risk-based adaptive policies, and data protection
- Assume breach - Segment access by network, user, devices, and application. Use encryption to protect data and analytics to get visibility, detect threats, and improve overall security

Foundational pillars of Zero Trust

Identities: when an identity tries to access a resource, it must be verified with strong authentication and follow least privilege access principles

Devices: create a large attack surface. Monitoring for health and compliance is an important aspect.

Applications: the way data is consumed. Discover all applications used - sometimes referred to as shadow IT because not all of them are centrally managed. It also includes managing

permissions and access.

Data: should be classified, labelled, and encrypted based on its attributes.

Infrastructure: represents a threat vector. Assess for version, config, and JIT acess and use telemetry to detect attacks and anomalies. Allows you to automatically block or flag risky behaviour.

Networks: should be segmented, including in-network microsegmentation. Real-time threat protection, end-to-end encryption, monitoring, and analytics should be employed.

Encryption and hashing

There are two top-level types of encryption: symmetric, and asymmetric. Symmetric uses the same key to encrypt/decrypt. Asymmetric uses a paired key (public and private)

Encryption for data at rest

Stored on a device (e.g. server) where a key is needed to decrypt it.

Encryption for data in transit

Data is encrypted when moving from one location to another e.g. across the internet or a private network. HTTPS is an example of this.

Encryption for data in use

Secures data in nonpersistent storage e.g. RAM or CPU cache. An enclave is created that protects the data while the CPU processes it.

Hashing

An algorithm that converts text to a unieuq fixed length value called a hash.

It is used to store passwords. When a user enters their password, the algorithm that created the stored hash creates a hash of the password. If they both match, then the password is correct.

To prevent further risk, passwords are often "salted", which is when a fixed length random value is added to the input of the hash function so that unieuq hashes are created for the same input.

Compliance Concepts

Data residency: regulations govern the physical locations of where data can be stored, how and when it can be transferred, processed or accessed internationally.

Data sovereignty: data (especially personal data) is subject to laws and regulations of the contry/ region of where it is physically collected, held, and processed.

Data privacy: providing notice and being transparent about the collection, processing and use of personal data.