Cloud Security Best Practices

4.4 Given a scenario, apply security best practices

Description

- In this episode, we will touch on many cybersecurity best practices to help us create a more secure cloud environment. These best practices includes Zero Trust, Benchmarks, System Hardening, Encryption strategies, Container Security, API Security, and more.

Resources

- https://www.cisecurity.org/cis-benchmarks
- https://learn.microsoft.com/en-us/azure/well-architected/security/harden-resources
- https://www.netwrix.com/windows-server-hardening-checklist.html
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/security_hardening/index
- https://media.defense.gov/2022/Aug/29/2003066362/-1/-1/0/CTR_KUBERNETES_HARDENING_GUIDANCE_1.2_20220829.PDF
- https://jwt.io/

Learning Objectives

- Define and explain common cybersecurity best practices such as Zero Trust, Benchmarks, Hardening, Encryption, Container Security, API Security and more

Notes

- Zero Trust
    - Verify EVERYTHING Explicitly
        - Always authenticate and authorize based on all available data points
            - User identity
            - Device health
            - Service or workload
            - Data classification
            - Anomalies
    - Least Privilege Access
        - Limit user and application access rights to only what is necessary
        - Apply just-in-time (JIT) and just-enough-access (JEA)
            - This will help minimize exposure to sensitive data and systems
    - Assume Breach
        - Assume that an attack can happen at any point and design your environment
            - This limits the potential impact of a breach
                - This includes segmenting access to minimize lateral movement within the network
- Benchmark
    - Center for Internet Security (https://www.cisecurity.org/cis-benchmarks)
    - Vendor-specific
- Hardening
    - Azure Hardening Guide (https://learn.microsoft.com/en-us/azure/well-architected/security/harden-resources)
    - Windows Server Hardening Guide (https://www.netwrix.com/windows-server-hardening-checklist.html)
    - Red Hat Hardening Guide (https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/security_hardening/index)
    - Kubernetes Hardening Guide (https://media.defense.gov/2022/Aug/29/2003066362/-1/-1/0/CTR_KUBERNETES_HARDENING_GUIDANCE_1.2_2022
- Patching
    - *AWS System Manager Patch Manager*

    - *Azure Update Manager*
- Encryption
    - Data in transit
        - HTTPs
        - SSH
    - Data at rest
        - *Azure Storage* is encrypted by default
            - Uses AES-256
        - *AWS KMS* (show when creating S3 bucket)
- Secrets management
    - *AWS Secrets Manager*
    - *Azure Key Vault*
    - *Google Cloud Secrets Manager*
- API security
    - Common Threats
        - Sensitive Data Exposure
        - Broken Access
        - Injections
            - WAFs
            - Encryption
                - JWT Security
                    - https://jwt.io/
    - *AWS CloudTrail*
    - *Azure API Management*
- Principle of least privilege
- Container security
    - Privileged and UnPrivileged
        - Refers to the level of permissions and access rights a container has to the underlying host system
            - **Privileged**
                - Root-level access
                    - Should be avoided except in special cases
                        - Running Docker in Docker
            - **Unprivileged**
                - Runs with limited permissions
                    - Restricted to the resources and capabilities explicitly allowed by the container runtime and security policie
- Storage security
    - Object storage
        - Data is stored as "objects"
            - Each object contains
                - The Data
                - Metadata about the Data
                - Unique ID
        - Highly scalable
        - Low cost
        - Built-in redundancy

        - Use Cases
            - Store unstructured data like...
                - Multimedia files
                - Backups
                - Logs
                - Archives
        - *AWS S3*, *Google Cloud Storage*, *Azure Blob Storage*
    - File storage
        - Organizes data in a hierarchical structure
        - Accessed with NFS or SMB
        - Compatible with apps that utilize a traditional file system
        - Use Cases
            - Shared file access
            - Home directories
            - Dev environments
            - Databases
        - *AWS Elastic File System* (EFS), *Google Cloud Filestore*, *Azure Files*