Melvin Jennings

Task 1 answer:

**1. Threat Intelligence Report:**

**Types of Attacks**

* **SQL Injection**
* **Cross-Site Scripting (XSS)**
* **Remote Code Execution (RCE)**
* **Privilege Escalation**

**Exploiting a Vulnerability for Network Access**

When a vulnerability in a web application is exploited, it can serve as a foothold into the network. For example, if an attacker exploits a vulnerability that allows them to execute arbitrary code, they could use this access to interact with the server’s operating system, escalate privileges, and move laterally within the network. This could lead to accessing sensitive data, compromising other systems, or deploying malware.

**Preventive Measures**

* **Regular Patch Management**
* **Web Application Firewalls (WAF)**
* **Input Validation and Sanitization**
* **Security Testing**

**2. Incident Response Plan**

**Containment**

* **Immediate Isolation**
* **Backup and Preserve Evidence**
* **Communication**

**Eradication**

* **Root Cause Analysis**
* **Patch and Remediation**
* **Malware Removal**

**Recovery**

* **System Restoration**
* **Monitor for Re-infection**
* **Post-Incident Review**

**3. Network Security Measures**

**Intrusion Detection and Prevention Systems (IDS/IPS)**

* Deploy IDS/IPS solutions to monitor and analyze network traffic for signs of suspicious activity or known attack signatures. These systems can detect and block malicious traffic before it reaches critical systems.

**Firewalls**

* Implement stateful firewalls to control incoming and outgoing network traffic based on predetermined security rules. Ensure that firewall configurations are regularly reviewed and updated to reflect the current threat landscape.

**Network Segmentation**

* Divide the network into segments based on function, sensitivity, or access level. For example, separate the web application servers from the internal network to limit the attacker's ability to move laterally if they compromise the web server.

**Multi-Factor Authentication (MFA)**

* Implement MFA for all users accessing sensitive systems, especially for administrative accounts, to reduce the risk of unauthorized access due to stolen credentials.

Task 2 answer:

**Five Docker Security Best Practices**

1. **Use Official or Trusted Images:**
   * Always use official or verified images from trusted repositories. These images are more likely to be regularly updated and audited for security vulnerabilities.
2. **Minimize the Image Size:**
   * Use minimal base images like alpine to reduce the attack surface by including only the necessary components and dependencies.
3. **Run Containers as Non-Root User:**
   * Avoid running containers as the root user to minimize the impact of a security breach. Create a non-root user and use that for running the application.
4. **Use Multi-Stage Builds:**
   * Use multi-stage builds to separate the build environment from the runtime environment. This reduces the final image size and removes unnecessary tools that could be exploited.
5. **Enable Docker Content Trust (DCT):**
   * Use Docker Content Trust to ensure that images are signed and verified before being pulled, providing a mechanism to ensure the integrity and authenticity of Docker images.

**Implementing a security practice in a Dockerfile:**

It is titled “Dockerfile”. It implements running the container as a non-root user.

**2. Kubernetes Security Configuration**

**Three Kubernetes Security Features**

1. **Pod Security Policies (PSP):**
   * PSPs define a set of conditions that a pod must meet to be accepted into the system, such as running as a non-root user, restricting the use of host networking, and controlling access to sensitive host resources.
2. **Role-Based Access Control (RBAC):**
   * RBAC allows for fine-grained control over who can access specific resources within the Kubernetes cluster. It defines roles and permissions that determine what actions users or service accounts can perform.
3. **Network Policies:**
   * Network policies define how pods communicate with each other and with other network endpoints. They allow for the creation of firewall rules within the cluster, controlling traffic flow at the IP address or port level.

**Kubernetes YAML Configuration with SecurityContext:**It’s titled securityContext.yaml

**3. IaaS Security Measures**

**Concept of Infrastructure as a Service (IaaS) and Its Security Implications**

**Infrastructure as a Service (IaaS)** is a cloud computing model where providers offer virtualized computing resources over the internet. These resources include virtual machines, storage, and networking, allowing organizations to run their workloads in a scalable and flexible environment without investing in physical infrastructure.

**Security Implications:**

* **Shared Responsibility Model:** In IaaS, the cloud provider is responsible for securing the physical infrastructure, while the customer is responsible for securing their data, applications, and operating systems. This includes configuring firewalls, managing identity and access, and ensuring proper encryption.
* **Network Security:** Customers must ensure that their virtual networks are properly segmented and secured using firewalls, VPNs, and other network security measures to prevent unauthorized access.
* **Data Protection:** It's essential to protect data at rest and in transit using encryption, ensuring that sensitive information is not exposed to unauthorized parties.