Session10 Assignment(Red Team Operations Unmasking Your Defenses)

Red Team Engagement Plan for Amazon

1. Scope: Target Systems and Objectives

Target Systems:

1.Cloud Infrastructure:

- Amazon Web Services (AWS) components like S3 buckets, EC2 instances, and Lambda functions.

- APIs and cloud gateways exposed externally.

2.External Infrastructure:

- Public-facing web servers and applications (e.g., Amazon's e-commerce platform).

- Identity and Access Management (IAM) configurations.

3. Employees:

- Corporate email accounts and internal communication channels.

- Social engineering opportunities targeting employees.

4. Third-Party Integrations:

- External vendors or service providers with access to Amazon's systems.

Objectives:

- Test AWS configurations, focusing on misconfigurations like exposed S3 buckets or over-permissive IAM policies.

- Assess the resilience of Amazon's public-facing infrastructure against advanced attacks.

- Evaluate employee susceptibility to phishing and other social engineering techniques.

- Simulate scenarios where attackers aim to compromise high-value targets (e.g., customer databases or financial systems).

2. Engagement Phase

Phase 1: Planning

- Define Rules of Engagement (ROE):

- Specify boundaries (e.g., avoid disrupting customer-facing services).

- Obtain written approval for testing AWS resources.

- **Reconnaissance**
  - Use OSINT tools to gather information about Amazon's external footprint (e.g., subdomains, public repositories).
  - Identify AWS-related endpoints or credentials potentially exposed in public forums or GitHub.
- **Team Preparation:**
  - Set up a secure infrastructure for command and control (C2).
  - Prepare custom phishing campaigns targeting employees.

Phase 2: Execution

1. Initial Access:

- Launch spear-phishing attacks targeting Amazon employees to steal AWS credentials.
- Identify and exploit misconfigured public AWS services like open S3 buckets.

2. Privilege Escalation:

- Analyze IAM policies for over-privileged roles or misconfigurations.
- Exploit server-side vulnerabilities to escalate privileges in AWS.

3. Lateral Movement:

- Use AWS services like Lambda or Systems Manager to navigate within the cloud environment.
- Identify high-value assets such as databases or customer-sensitive information.

4. Data Exfiltration:

- Simulate exfiltrating sensitive data (e.g., customer information from RDS databases).

5. Persistence:

- Deploy backdoors, such as unauthorized IAM users or roles.

Phase 3: Reporting

- **Findings:**
  - Document all misconfigurations and vulnerabilities in the AWS environment.
  - Highlight the effectiveness of Amazon's monitoring and response systems.
- **Report:**
  - Provide detailed insights into the attack vectors used and recommendations for mitigation.
- **Presentation:**
  - Conduct a debriefing session for Amazon's security team.

3. Potential Attack Vector

- **Credential Theft:** Leverage exposed or stolen credentials to access AWS services.

- **Social Engineering:** Target employees to reveal sensitive information (e.g., MFA codes).

- **API Abuse:** Exploit weak or unprotected APIs.

- **Misconfigured Services:** Access publicly exposed S3 buckets or exploit overly permissive IAM policies.

- **Supply Chain Attacks:** Target third-party vendors integrated into Amazon's ecosystem.

4&5. Tools Used

1. Cobalt Strike:

- **Purpose:** Simulates advanced persistent threats (APTs) for command and control.

- **Example:** Establish a foothold after stealing AWS credentials to execute post-exploitation activities.

2. Metasploit:

- **Purpose:** Exploitation of vulnerabilities in external-facing applications or cloud infrastructure.

- **Example:** Exploit a vulnerability in Amazon's public-facing APIs to gain unauthorized access.

3. BloodHound:

- **Purpose**: Map trust relationships and privilege paths in hybrid environments, including AD-integrated AWS setups.

- **Example:** Identify excessive permissions in Amazon's IAM configurations.

4. Empire:

- **Purpose:** Post-exploitation framework for maintaining access to compromised systems.

- **Example:** Use PowerShell or Python scripts to manipulate AWS services or exfiltrate sensitive data.

5. Mimikatz:

- **Purpose:** Credential harvesting from Windows-based systems.

- **Example:** Extract cached credentials from Amazon employees' machines during on-premise assessments.