A **kill chain** outlines the stages of a potential attack, mapping each stage to corresponding vulnerabilities or weaknesses in the system. The kill chain for the provided findings can be structured to highlight how these vulnerabilities may be exploited and what stages of the attack lifecycle they affect.

**Kill Chain for Identified Vulnerabilities**

**1. Reconnaissance**

**Description:** The attacker gathers information about the target environment, including network configurations, accessible systems, and exposed metadata.

* **Vulnerability Exploited:**
  + **Excessive Internet Access in Containers:** Forward proxy allows unnecessary internet access, enabling attackers to collect information about the environment.
  + **Unauthorised Access to Server Metadata (IMDSv1):** Metadata service exposure allows attackers to retrieve sensitive instance details such as IAM roles and credentials.
* **Impact:** This stage allows the attacker to identify potential entry points, gather credentials, or discover services to target in subsequent stages.

**2. Weaponization**

**Description:** The attacker crafts malicious payloads or scripts to exploit discovered vulnerabilities.

* **Vulnerability Exploited:**
  + **GitHub Access in Containers:** Access to external repositories allows downloading malicious payloads or tools directly into the container.
  + **Secrets Manager Key Exposure:** Visible secrets in AWS Secrets Manager may be used to generate malicious requests or access sensitive resources like databases.
* **Impact:** The attacker can prepare exploits or tools tailored to the environment's weak points, including misconfigured network isolation or exposed secrets.

**3. Delivery**

**Description:** The attacker delivers the malicious payload to the target system or container.

* **Vulnerability Exploited:**
  + **Privileged Containers on EC2 Instances:** Privileged containers allow attackers to directly interact with the host system, increasing the likelihood of successful payload delivery.
  + **Excessive Internet Access:** Unnecessary internet access facilitates the download of malicious payloads.
* **Impact:** This stage enables the attacker to insert malicious code or establish communication channels with external servers.

**4. Exploitation**

**Description:** The attacker triggers the vulnerability to gain initial access or escalate privileges.

* **Vulnerability Exploited:**
  + **IMDSv1 Exploitation:** SSRF attacks against IMDSv1 can grant access to metadata and IAM credentials.
  + **Privilege Escalation via Root Containers:** Privileged containers allow attackers to gain root access to the EC2 host.
  + **Non-Root User Exploitation:** Even non-root users can run vulnerable packages to exploit the environment.
* **Impact:** The attacker gains unauthorized access to critical resources, escalates privileges, or deploys backdoors in the system.

**5. Installation**

**Description:** The attacker establishes persistence by installing malicious software, backdoors, or command-and-control tools.

* **Vulnerability Exploited:**
  + **GitHub Repository Access:** Open-source or malicious tools downloaded from GitHub can be used to establish persistence.
  + **Secrets Manager Exposure:** Exposed secrets can enable attackers to install malware that masquerades as legitimate software.
* **Impact:** The attacker ensures their foothold remains, even if their initial attack vector is remediated.

**6. Command and Control (C2)**

**Description:** The attacker establishes communication with the compromised systems to exfiltrate data or execute commands.

* **Vulnerability Exploited:**
  + **Network Isolation Lapse:** Lack of granular container-level isolation allows attackers to move laterally within the container environment.
  + **Privileged Access in Containers:** Root access to the EC2 host provides unrestricted network and system control.
* **Impact:** The attacker can execute arbitrary commands, steal sensitive data, or spread the attack across other containers or hosts.

**7. Actions on Objectives**

**Description:** The attacker achieves their final goals, such as data exfiltration, system disruption, or lateral movement.

* **Vulnerability Exploited:**
  + **Secrets Manager Key Exposure:** Access to sensitive keys allows attackers to extract data from databases or APIs.
  + **Network Isolation Weakness:** Lack of Zero Trust controls permits lateral movement to other containers or applications.
  + **IMDSv1 Exploitation:** Exposed metadata service grants access to IAM credentials, enabling further attacks.
* **Impact:** The attacker achieves objectives like exfiltrating sensitive data, compromising additional systems, or disrupting the environment.

**Summary of the Kill Chain**

| **Stage** | **Vulnerabilities Exploited** | **Impact** |
| --- | --- | --- |
| Reconnaissance | Excessive Internet Access, IMDSv1 | Discovery of environment details, IAM credentials, and exposed services. |
| Weaponization | GitHub Access, Secrets Exposure | Creation of custom payloads or scripts to exploit vulnerabilities. |
| Delivery | Privileged Containers, Internet Access | Delivery of malicious payloads or tools to containers or host systems. |
| Exploitation | IMDSv1 Exploitation, Privileged Containers, Non-Root Exploitation | Unauthorized access, privilege escalation, or execution of malicious code. |
| Installation | GitHub Access, Secrets Exposure | Installation of backdoors, persistence tools, or malware. |
| Command and Control | Network Isolation Weakness, Privileged Containers | External communication for data exfiltration, lateral movement, or remote execution. |
| Actions on Objectives | Secrets Exposure, Network Isolation Weakness, IMDSv1 Exploitation | Data theft, lateral movement across systems, or complete compromise of EC2 instances and containerized applications. |

**Key Recommendations**

1. **Restrict Internet Access**: Apply strict egress network policies to limit internet access to only essential endpoints.
2. **Enforce Privileged Access Management**: Remove privileged containers and apply the least privilege principle.
3. **Enable IMDSv2**: Configure EC2 instances to enforce the use of IMDSv2 to prevent metadata exploitation.
4. **Secrets Management**: Limit IAM access to AWS Secrets Manager, ensure secrets are encrypted, and implement logging for access events.
5. **Granular Network Isolation**: Implement Zero Trust principles for inter-container communication using Kubernetes network policies or service mesh solutions.
6. **Patch Management**: Centralize container patching processes to enforce periodic updates.