1. Executive Summary (Non-Technical)

Engagement Overview

A penetration test was conducted against a Kubernetes-based environment to identify security weaknesses that could be exploited by an external attacker. The assessment focused on discovering exposed services, misconfigurations, and insecure access controls within the cluster.

Key Findings

The assessment identified **multiple critical security misconfigurations**, including exposed services and weak Kubernetes configurations, which could allow an attacker to gain unauthorized access and escalate privileges within the environment.

Business Impact

If exploited in a real-world environment, these vulnerabilities could result in:

- Unauthorized access to sensitive workloads.
- Compromise of containerized applications.
- Lateral movement within cloud infrastructure.
- Potential data breach and service disruption.

Overall Risk Rating - HIGH

2. Scope & Methodology

Scope

The penetration test was conducted against:

- A publicly accessible host simulating a Kubernetes cluster
- Exposed network services including web services and Kubernetes-related components

Out of Scope

- Denial of Service (DoS)
- Social engineering
- Physical access attacks

Methodology

The assessment followed industry-recognized methodologies:

- **OWASP Testing Guide**
- **OWASP Kubernetes Top 10**
- **MITRE ATT&CK Framework**
- Black-box testing approach (no credentials provided)

3. Attack Surface Overview

Identified Open Services

Port	Service	Observation
22	SSH	Exposed to Public Network
6443	Kubernetes API	Exposed and accessible
10250	Kubelet API	
30180	Nginx	Public-facing web service
31111	Gitea	Public-facing web service
31112	SSH	

These services significantly expanded the attack surface and enabled further enumeration.

4. Detailed Findings

Finding 1: Kubernetes API Exposure

Severity:	High
OWASP Kubernetes:	K01 – Insecure Workload Configuration
OWASP Web:	A02 – Security Misconfiguration
MITRE ATT&CK:	T1046 – Network Service Discovery

Description

The Kubernetes API server was accessible from the public network without sufficient access restrictions. This allowed unauthenticated or weakly authenticated enumeration of cluster resources.

Impact

An attacker could:

- Enumerate cluster resources
- Identify workloads, namespaces, or secrets
- Prepare for privilege escalation or container compromise

Evidence

- Nmap scan revealed port `6443` open
- Kubernetes API responded to enumeration attempts

Recommendation

- Restrict Kubernetes API access using firewall rules
- Enable strong authentication and authorization
- Use private endpoints where possible

Finding 2: Security Misconfiguration in Exposed Services

Severity: Medium
OWASP Web: A02 – Security Misconfiguration
MITRE ATT&CK: T1595 – Active Scanning

Description

Public-facing services such as nginx were accessible without sufficient hardening, allowing attackers to enumerate directories and application behavior.

Impact

Attackers could:

- Gather sensitive information
- Identify vulnerable endpoints
- Chain findings with Kubernetes misconfigurations

Recommendation

- Harden web server configurations
- Disable unnecessary endpoints
- Implement proper access controls

Finding 3: Kubernetes Enumeration & Discovery

Severity: High
OWASP Kubernetes: K04 – Insecure API Access
MITRE ATT&CK: T1087 – Account Discovery
MITRE ATT&CK: T1613 – Container and Resource Discovery

Description

The exposed environment allowed attackers to enumerate Kubernetes resources, revealing cluster structure and potential privilege boundaries.

Impact

This discovery phase enables:

- Lateral movement
- Privilege escalation
- Full cluster compromise

Recommendation

- Implement RBAC least privilege
- Audit Kubernetes permissions
- Monitor API access logs

5. Attack Path Summary (Kill Chain)

Phase	Technique	Description
Reconnaissance	Active Scanning	Nmap Service Discovery
Initial Access	Exploit Public-Facing Service	Access to exposed APIs
Discovery	Kubernetes Enumeration	Cluster Resource Discovery
Privilege Escalation	Misconfiguration Abuse	Weak RBAC and API exposure

6. Risk Rating Matrix

Risk	Likelihood	Impact	Overall
Kubernetes API Exposure	High	High	Critical
Service Misconfiguration	Medium	Medium	Medium

7. Remediation Summary (Prioritized)

Immediate (0–30 days)

- Restrict Kubernetes API exposure
- Enforce authentication and RBAC
- Harden exposed web services

Short Term (30–60 days)

- Implement Kubernetes network policies
- Enable logging and monitoring
- Conduct configuration audits

Long Term (60–90 days)

- Continuous security scanning
- Kubernetes security training for engineers
- Regular penetration testing

8. Conclusion

The assessment revealed that insecure Kubernetes configurations significantly increase the risk of compromise. Addressing these issues will reduce the likelihood of unauthorized access, data exposure, and infrastructure compromise.

A defense-in-depth approach is strongly recommended.

9. Appendix

Tools Used

- Nmap
- Kubernetes CLI (kubectl)
- Web enumeration tools

Frameworks Referenced

- OWASP Top 10 (Web & Kubernetes)
- MITRE ATT&CK