Cybersecurity Lab Portfolio - Alen Guner

Lab 1: Brute Force Detection & Alerting in a SIEM (Wazuh + Linux)

Objective:

Simulate a brute-force attack on a Linux SSH service and configure a SIEM (Wazuh) to detect, alert, and log the incident. Learn how to analyze logs and build alert rules based on attacker behavior.

Tools Used:

- Kali Linux (Attacker)
- Ubuntu Linux (Target)
- Wazuh SIEM (Manager & Agent)
- SSH (OpenSSH)
- Fail2Ban (optional)
- VirtualBox or VMware

Steps Taken:

1. Environment Setup:
- Deployed two virtual machines: Ubuntu (target) and Kali (attacker).
- Installed Wazuh Manager on a third VM or monitored directly from Ubuntu.
- Enabled SSH on the Ubuntu machine and set up weak credentials.

2. Attack Simulation:
- Used hydra from Kali to launch a brute-force attack against the Ubuntu SSH service:
  hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://[target_ip]

3. SIEM Integration:
- Installed Wazuh agent on Ubuntu and configured to forward logs to Wazuh Manager.
- Enabled SSH log monitoring and authentication failure rule sets.
- Created a custom alert for more than 5 failed login attempts within 1 minute.

4. Analysis:
- Viewed real-time alerts in Wazuh dashboard.
- Investigated /var/log/auth.log and correlated IP addresses, timestamps, and usernames.
- Created a timeline of attack and response.

5. Optional Hardening:
- Set up fail2ban to block attacker IPs automatically.
- Tested rule efficacy by repeating the attack post-hardening.

Outcome / What I Learned:

- Identified brute-force patterns based on authentication logs.
- Created custom SIEM rules and alert thresholds.

Cybersecurity Lab Portfolio - Alen Guner

- Improved familiarity with log parsing, alert correlation, and basic incident response workflows.
- Gained insight into SOC monitoring tools and detection strategies.

Keywords:

SIEM, Brute Force Detection, Wazuh, Linux Security, Incident Response, Log Analysis, Hydra, IDS/IPS

Cybersecurity Lab Portfolio - Alen Guner

Lab 2: Cloud IAM Misconfiguration Detection & Remediation (AWS)

Objective:

Identify and remediate overly permissive IAM roles in a simulated AWS environment. Simulate privilege escalation paths and secure access policies for least privilege.

Tools Used:

- AWS Free Tier (IAM Console)
- Cloudsplaining / ScoutSuite
- TryHackMe (Cloud IAM room)
- IAM Policy Simulator
- AWS CLI (optional)

Steps Taken:

1. IAM Setup:
- Created multiple IAM users, groups, and roles in AWS Free Tier.
- Assigned various managed and custom policies, including an overly permissive admin role.

2. Risk Analysis:
- Ran Cloudsplaining to identify excessive permissions.
- Identified users with full `iam:*` or `s3:*` privileges.
- Simulated policy escalation paths using TryHackMe cloud labs.

3. Remediation:
- Refactored IAM policies using least privilege principles.
- Removed unused policies and restricted `iam:PassRole` permissions.
- Applied policy boundaries and session duration controls.

4. Documentation:
- Created an IAM role permission map.
- Documented misconfiguration examples and how to fix them.

5. Optional Tools:
- Used IAM Policy Simulator to validate permissions before deployment.
- Audited results with AWS CLI queries and ScoutSuite reports.

Outcome / What I Learned:

- Gained practical skills in identifying dangerous IAM policies.
- Learned how attackers exploit privilege escalation paths.
- Developed confidence using AWS IAM Console and security auditing tools.
- Improved understanding of least privilege and compliance needs.

Cybersecurity Lab Portfolio - Alen Guner

Keywords:

AWS, IAM, Cloud Security, Privilege Escalation, Least Privilege, Policy Simulator, Cloudsplaining, ScoutSuite, Identity Management