CyArt Red Teaming – Week 2 Security Operations Report

1. Advanced Threat Analysis

Advanced threat analysis focuses on understanding attacker behavior and predicting potential attack paths before incidents occur. The STRIDE threat modeling methodology helps identify security weaknesses such as spoofing or privilege escalation within web applications. By mapping trust boundaries and entry points, defenders can visualize risks more effectively. The MITRE ATT&CK; framework was studied to align detection strategies with real-world adversary techniques. Supply chain attacks and zero-day vulnerabilities were also reviewed to understand how sophisticated threat actors maintain long-term persistence inside networks.

Threat Modeling Table (Sample)

Asset	Threat	STRIDE Category	Mitigation
User Login Portal	Credential Spoofing	Spoofing	MFA + Logging
Database	Data Tampering	Tampering	Access Control Policies
API Gateway	Service Abuse	DoS	Rate Limiting

2. Security Framework Implementation

Security frameworks provide structured guidance for building mature defense strategies. The NIST Cybersecurity Framework was applied across Identify, Protect, Detect, Respond, and Recover phases. ISO 27001 control mapping was reviewed to improve logging, secure development, and backup strategies. Framework alignment ensures that organizations can maintain consistent security posture while responding effectively to incidents such as ransomware campaigns.

Framework Function	Implementation Example	Outcome
Identify	Asset Inventory	Improved Visibility
Protect	Access Controls	Reduced Unauthorized Access

Detect	SIEM Monitoring	Early Threat Detection
Recover	Backup Strategy	Faster Restoration

3. Incident Response Fundamentals

Incident response activities were structured around preparation, detection, containment, eradication, and recovery. SOC workflows and playbooks help analysts respond quickly by providing predefined actions for common threats. Simulated phishing scenarios demonstrated how attackers gain initial access and how monitoring tools identify anomalies. The response workflow emphasized communication, evidence collection, and maintaining operational continuity during incidents.

4. Threat Hunting and Detection Engineering

Threat hunting exercises were performed using Elastic Security and Sigma rules to detect suspicious PowerShell activity. Log analysis focused on Event ID 4688 process creation events to identify abnormal command execution. Detection engineering requires continuous tuning of rules to minimize false positives while improving visibility into attacker behavior.

Timestamp	Process	Command Line	Notes
2025-08-18 10:00	powershell.exe	-Command Write-Host	Suspicious Execution
2025-08-18 10:05	cmd.exe	/c whoami	Privilege Check

5. Malware Analysis and Vulnerability Management

Basic malware analysis involved static inspection using tools such as strings and peframe to identify embedded indicators within executable files. Dynamic behavior analysis was compared with sandbox reports to understand runtime activity. Additionally, vulnerability management workflows were implemented using OpenVAS scanning and centralized tracking platforms. High-risk vulnerabilities were prioritized based on CVSS scoring to guide remediation planning.

Vulnerability	CVSS Score	Risk Level	Mitigation
VSFTPD Backdoor	7.5	High	Patch Service
Weak SSH Config	6.1	Medium	Disable Root Login
Outdated Apache	5.4	Medium	Upgrade Version

6. Risk Assessment and Incident Reporting

Risk assessment exercises included calculating Annualized Loss Expectancy (ALE) to quantify potential financial impact. For example, a ransomware scenario with a Single Loss Expectancy of $10,000 and an Annual Rate of Occurrence of 0.2 resulted in an ALE of $2,000. Clear documentation ensures that technical findings can be understood by both security teams and business stakeholders.

Scenario	SLE	ARO	ALE
Ransomware Attack	$10,000	0.2	$2,000