Theoritical and Pratical Tasks

1. SOC Fundamentals & Operations

1.1. Introduction

A Security Operations Center (SOC) is the centralized command unit responsible for maintaining and improving an organization's cybersecurity posture. It functions as the heart of security monitoring, where cyber threats are detected, analyzed, and remediated in real time.

1.2. Purpose of a SOC

The primary objective of a Security Operations Center is to ensure the continuous protection of an organization’s digital assets. Key purposes include:

1.2.1. Proactive Threat Detection

The SOC continuously collects and monitors logs, network traffic, and security alerts to identify malicious activities early. It aims at detecting threats before they manifest into large-scale attacks.

1.2.2. Incident Response

When a threat is detected, the SOC coordinates incident response procedures. These include isolating affected systems, containing the breach, performing root cause analysis, and restoring normal business operations.

1.2.3. Continuous Monitoring

SOC teams work 24/7 to ensure constant surveillance of endpoints, servers, firewalls, and applications. This enables real-time risk awareness and faster reaction to anomalies or attack attempts.

1.3. Roles Within a SOC

A SOC team consists of professionals with varying skill levels and responsibilities. Effective coordination among these roles drives robust cybersecurity defense.

1.3.1. Tier 1 Analyst (Monitoring & Triage)

- First level of alert monitoring.
- Investigates alerts generated by SIEM and other security tools.
- Performs initial triage, classifies alerts as true/false positives.
- Escalates severe incidents for deeper analysis.

1.3.2. Tier 2 Analyst (Incident Investigator)

- Performs in-depth analysis of escalated alerts.
- Conducts threat hunting using logs, packet captures, and endpoint data.
- Determines severity, impact, and scope of incidents.
- Responsible for partial remediation and containment actions.

1.3.3. Tier 3 Analyst / Threat Hunter (Advanced Detection)

- Handles sophisticated and high-severity cyber incidents.
- Conducts proactive threat hunting using behavioral patterns and intelligence feeds.
- Works closely with red teams to simulate attacks and improve defense.
- Develops automation rules, detection logic, and security playbooks.

1.3.4. SOC Manager

- Oversees day-to-day SOC operations.
- Manages team workflows, escalations, and resource allocation.
- Ensures compliance with organizational security policies and frameworks.
- Engages with executive leadership to communicate risk posture.

1.3.5. Threat Hunters

- Focus on identifying unknown (zero-day) threats.
- Uses anomaly detection, baselining, threat intel feeds, and MITRE ATT&CK mapping.
- Enhances detection mechanisms to eliminate blind spots.

1.4. Key Functions of a SOC

1.4.1. Log Analysis

- Collects and correlates logs from endpoints, servers, firewalls, cloud, and applications.
- Helps identify anomalies, suspicious activities, and forensic indicators.
- Performed using SIEM (Security Information and Event Management) tools.

1.4.2. Alert Triage

- Prioritizing security alerts based on severity, relevance, and business impact.
- Filters false positives to reduce alert fatigue.
- Ensures critical incidents are escalated immediately.

1.4.3. Threat Intelligence Integration

- Enriches detection capabilities with external intelligence feeds:
  - IOC (Indicators of Compromise)
  - TTP (Tactics, Techniques, Procedures)
  - Malware signatures
- Helps the SOC predict attacker behavior and refine defense strategies.

1.5. Summary

A Security Operations Center plays a crucial role in defending an organization from cyber threats. With structured roles, continuous monitoring, proactive threat intelligence, and effective incident response, a SOC ensures resilience against evolving digital attacks. Strong operational discipline, skilled analysts, and automation-driven workflows enhance the SOC's ability to protect critical assets and maintain business continuity.

2. Security Monitoring Basics

2.1. Introduction

Security monitoring is one of the foundational pillars of cybersecurity operations. It involves the continuous collection, analysis, and interpretation of security-related data across networks, systems, applications, and endpoints. The objective is to detect potential security threats, unauthorized activities, and operational abnormalities in real-time or near real-time. Effective monitoring forms the backbone of incident detection and response capabilities within an organization.

2.2. Primary Objectives of Security Monitoring

2.2.1. Detect Anomalies

Security monitoring focuses on identifying unusual behaviors, deviations from normal activity, and suspicious patterns. Anomalies may include unexpected data transfers, abnormal login times, unusual system resource usage, or unknown processes running in the environment. Early anomaly detection helps prevent breaches before damage is done.

2.2.2. Identify Unauthorized Access

Unauthorized access occurs when an attacker, insider, or malicious user gains entry into restricted systems or information. Monitoring login patterns, privileged access activity, authentication failures, and RDP/SSH sessions allows SOC teams to quickly spot access violations and respond before critical assets are compromised.

2.2.3. Observe Policy Violations

Monitoring ensures adherence to internal security policies and compliance frameworks. This includes tracking configuration changes, improper handling of confidential data, disabling security controls, or misuse of administrative privileges. Policy violations serve as early indicators of insider threats or operational weaknesses.

2.3. Security Monitoring Tools

Security monitoring tools enhance visibility, automate detection, and consolidate large volumes of security data into meaningful insights.

2.3.1. Security Information and Event Management (SIEM)

A SIEM platform is the central tool for log aggregation, correlation, and real-time alerting. SIEMs act as the primary visibility layer for a SOC. Popular SIEM solutions include:

SIEM Tool	Key Capabilities
Splunk	Advanced analytics, dashboards, threat detection, scalability
Elastic SIEM	Open-source, fast indexing, anomaly detection, search-based investigation
IBM QRadar	Threat correlation, compliance reporting, built-in intelligence
Microsoft Sentinel	Cloud-native SIEM with AI-based detection and automation

Table 1: SEIM Tool

2.3.2. Network Traffic Analyzers

Tools like Wireshark, Zeek, and Suricata help inspect network packets, detect malicious traffic patterns, and identify protocol misuse. Network monitoring complements log-based SIEM monitoring to achieve complete visibility. They enable analysts to:

- Capture and replay network packets.
- Detect command-and-control (C2) traffic.
- Identify malware communication or data exfiltration attempts.
- Investigate network-based incidents at packet level.

2.4. Key Metrics in Security Monitoring

Metrics help measure the effectiveness and maturity of the monitoring program.

2.4.1. False Positives & False Negatives

- **False Positive:** A benign activity flagged as malicious. Too many false positives cause alert fatigue and waste resources.

- **False Negative:** A real threat not detected by security systems. False negatives are more dangerous as they allow breaches to occur unnoticed.

2.4.2. Mean Time to Detect (MTTD)

MTTD measures how long it takes for the SOC to identify a threat after it occurs. Lower MTTD indicates efficient alerting, visibility, and investigation capability. Organizations work to reduce MTTD by:

- Improving log coverage and visibility.
- Tuning SIEM alerts with behavioral analytics.
- Implementing automated threat detection workflows.
- Using threat intelligence feeds to speed recognition.

2.5. Summary

Security monitoring serves as the first line of defense in detecting cybersecurity threats. By leveraging SIEM platforms, network analysis tools, and measurable detection metrics, organizations gain visibility into real-time activity and potential risks. When properly tuned, monitoring systems reduce exposure, improve response efficiency, and strengthen overall cybersecurity resilience.

3. Log Management Fundamentals

3.1. Introduction

Log management is an essential component of cybersecurity monitoring and digital forensics. Logs represent recorded system activities and user interactions, helping analysts detect suspicious behavior, investigate incidents, and maintain compliance. A robust log management strategy ensures that logs are collected, normalized, stored securely, and analyzed efficiently to support threat detection and operational visibility.

3.2. Log Lifecycle Overview

The log lifecycle consists of several stages designed to ensure the availability, quality, and usability of log data.

3.2.1. Collection

The first stage involves gathering logs from endpoints, servers, firewalls, databases, applications, and cloud environments. This is often done through agents or log collectors such as Fluentd, Logstash, Beats, or syslog forwarding mechanisms.

3.2.2. Normalization

Logs generated from different sources vary in structure. Normalization converts raw logs into a consistent format like **JSON** or **CEF (Common Event Format)**. This ensures compatibility with SIEM, analytics engines, and improves query efficiency.

3.2.3. Storage

Logs are stored in central repositories like ElasticSearch, Splunk indexers, or cloud log storage for long-term retention and fast retrieval. Storage considerations include indexing, compression, and security controls to prevent tampering.

3.2.4. Retention

Retention policies define how long logs are preserved based on compliance needs such as ISO, PCI-DSS, HIPAA, or institutional audit requirements. Critical logs may require multi-year retention, while routine logs may be archived periodically.

3.2.5. Analysis

Logs are queried using SIEMs or analytics tools to detect anomalies, correlate events, support incident response, and generate dashboards for real-time visibility.

3.3. Common Log Types

Understanding log categories helps analysts identify system behavior and trace incident timelines.

Log Type	Source	Use Case
Windows Event Logs	Windows hosts	Authentication failures, process creation, policy changes
Syslog	Linux systems, network devices	SSH access attempts, firewall events, service logs
HTTP Server Logs (Apache/Nginx)	Web servers	URL requests, response codes, potential exploitation attempts

Table 2: Log Types

3.4. Log Management

3.4.1. Log Collection Using Fluentd or Logstash

Configure Fluentd/Logstash input plugins to collect logs from Linux systems, Windows Event Forwarding, or network syslog sources. Test and forward logs to analyzers or SIEM tools.

3.4.2. Normalization

Standardize logs into CEF or JSON to make fields easier to parse and correlate across multiple systems. Normalization ensures uniformity, critical for SIEM correlation rules.

3.4.3. Query & Analysis

Queries help identify events like failed logins, privilege escalation, suspicious IPs, and malware execution. Tools such as Azure Sentinel (KQL), Elastic SIEM (KQL), or Splunk SPL provide powerful search and aggregation capabilities.

3.5. Practical Tasks

3.5.1. Log Collection Pipeline

Collect Syslog logs from Ubuntu using Fluentd, forward to Elastic SIEM.

Step 1 Install Fluentd on Ubuntu

curl -fsSL https://toolbelt.treasuredata.com/sh/install-ubuntu-bionic-td-agent4.sh | sh

Step 2 Configure Fluentd to Collect Syslog

Edit configuration:
sudo nano /etc/td-agent/td-agent.conf

Step 3 Restart Fluentd

sudo systemctl restart td-agent
sudo systemctl status td-agent

Step 4 Generate Test Logs

Use logger to simulate log events:
logger "Test message - Fluentd log pipeline working"
logger -p auth.info "Authentication Info Testing"
logger -p cron.err "Cron job error simulation"

3.5.2. KQL Query in Elastic SIEM

Query Windows failed login events using KQL.

Elastic SIEM Query

event.code : 4625
stats count by SourceIP

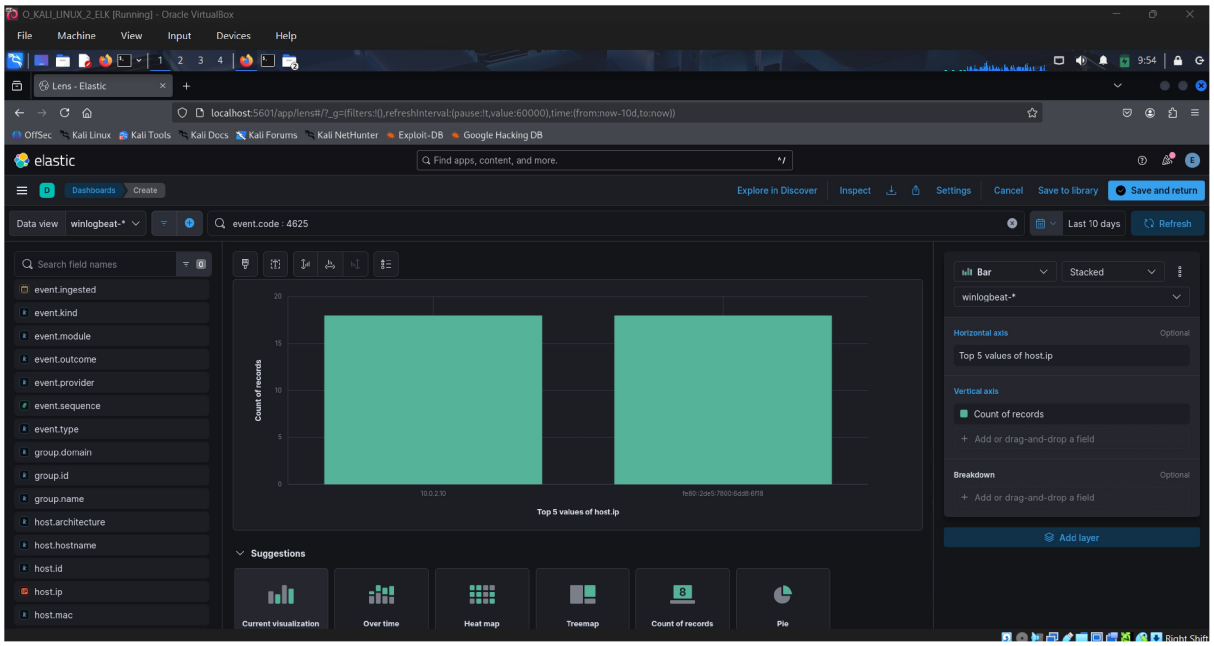

Image 1: event code 4625 by IP

Results :

SourceIP	Count
10.0.2.10	18

Table 3: Result

3.5.3. Logstash Normalization

Convert raw Apache access logs into clean JSON format.

Step 1 Install Logstash

sudo apt update

sudo apt install logstash -y

systemctl status logstash

Step 2 Create Logstash Pipeline

sudo nano /etc/logstash/conf.d/apache_json.conf

Step 3 Start Logstash Pipeline

sudo systemctl restart logstash

sudo tail -f /tmp/apache_logs.json

Step 4 Verify Output Format

sudo cat /tmp/apache_logs.json | *jq .*

3.6. Summary

Log management is fundamental to security operations, providing investigators valuable visibility into system behavior. By understanding log lifecycle stages, common log categories, and hands-on practices, analysts enhance their detection capabilities and incident response readiness.

4. Security Tools Overview

4.1. Introduction

Security tools are essential elements in modern cybersecurity operations. They enable analysts to detect threats, secure endpoints, identify vulnerabilities, and monitor network traffic. A well-rounded SOC analyst must understand how key tools work, where they are used, and how to deploy them in real-world scenarios.

4.2. Snort IDS Rule Testing

Intrusion Detection and Prevention Systems (IDS/IPS) inspect network traffic for malicious patterns. Snort, a widely used open-source IDS/IPS, allows administrators to write custom rules to detect or block suspicious traffic. It can operate in detection mode (alerting on threats) or prevention mode (actively blocking attacks). Install Snort on Ubuntu, write a rule to detect HTTP access to *malicious.com*, and test with curl.

Step 1 Install Snort

sudo apt update
sudo apt install snort -y

Step 2 Create Custom Snort Rule

Open rules file:
sudo nano /etc/snort/rules/local.rules

Add rule:
alert tcp any any -> any 80 (msg:"Malicious Domain"; content:"malicious.com"; http_uri; sid:1000001;)

Step 3 Test Rule

Run Snort in packet capture mode:
sudo snort -A console -q -c /etc/snort/snort.conf -i eth0

Execute:
curl http://malicious.com

Output

[**] [1:1000001:0] Malicious Domain [**]

4.3. Nessus Vulnerability Scan on Metasploitable2

Vulnerability scanners identify weaknesses in systems, applications, and configurations. Nessus is one of the most popular tools, capable of scanning operating systems, databases, and applications for known vulnerabilities. It provides detailed reports with CVSS scores, helping organizations prioritize patching and remediation efforts. Scan a vulnerable VM Metasploitable 2.

Step 1 Install Nessus Essentials

wget
https://www.tenable.com/downloads/api/v1/public/pages/nessus/downloads/nessus-10.5.1-ubuntu1110_amd64.deb
sudo dpkg -i nessus-.deb*
sudo systemctl start nessusd

Open in browser:
https://localhost:8834/

Step 2 Add Metasploitable as Scan Target

1. Create new **Basic Network Scan**
2. Target IP = Metasploitable2 IP
3. Run Scan

Vulnerability	CVSS	Description
VNC Server 'password' password	10.0	Gain a shell remotely
SSL Version 2 and 3 protocol Detection	9.8	Service detection, DROWN Attack
Bind Shell Backdoor Detection	9.8	Backdoor

Table 4: Findings

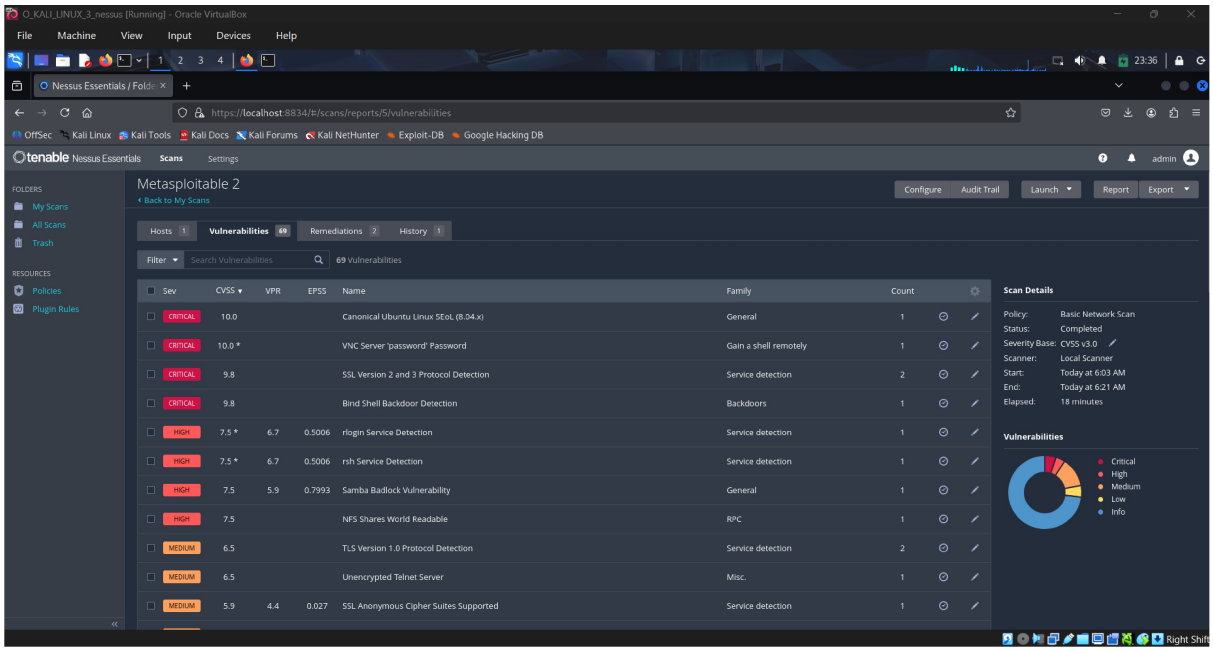

Image 2: Vulnerabilities

4.4. Summary

This hands-on exercise provides practical exposure to IDS and vulnerability scanning workflows. Snort rules build network detection skills, while Nessus scanning teaches threat assessment and risk evaluation.

5. Basic Security Concepts

5.1. Introduction

Basic security concepts form the foundation of cybersecurity knowledge. They provide a logical framework for protecting data, identifying weaknesses, and designing secure architectures. Every SOC analyst, security engineer, and defender should clearly understand these principles before dealing with advanced security operations or security tools.

5.2. CIA Triad

The CIA Triad is the core model of information security and defines the three most essential security objectives for protecting data.

5.2.1. Confidentiality

- Ensures information is accessible only to authorized users.
- Prevents unauthorized access, disclosure, or exposure.
- Achieved through:
  - Encryption (AES, TLS)
  - Access control & authentication
  - Data masking and classification

5.2.2. Integrity

- Maintains accuracy, consistency, and reliability of data.
- Ensures data is not altered, deleted, or tampered with.
- Achieved through:
  - Hashing (SHA-256)
  - Checksums and digital signatures
  - File integrity monitoring (Tripwire, Wazuh)

5.2.3. Availability

- Ensures systems and data remain accessible when needed.
- Prevents service disruptions from attacks or failures.
- Achieved through:

- Redundancy, backups, load balancing
- DDoS protection & rate limiting
- Disaster recovery planning

5.3. Threat vs Vulnerability vs Risk

Understanding the difference between these terms helps in incident assessment and prioritization.

Term	Meaning	Example
Threat	Any potential cause of damage or harm	Hacker, malware, insider abuse
Vulnerability	Weakness in a system that can be exploited	Open port, outdated patch, weak password
Risk	Likelihood of threat exploiting vulnerability & causing impact	Ransomware exploiting a missed security update

Table 5: Comparision

Formula representation:

Risk = Threat × Vulnerability × Impact

5.4. Defense-in-Depth

Defense-in-Depth is a layered security strategy, ensuring that even if one security layer fails, others still protect the environment. Like an onion, multiple layers defend the core.

Components Include:

1. Perimeter Security > Firewall, IDS/IPS
2. Endpoint Security > EDR, antivirus, patching
3. Identity & Access > MFA, RBAC
4. Network Segmentation > VLANs, micro-segmentation
5. Data Protection > Encryption, backups
6. Monitoring & Response > SIEM, SOC, threat intel

5.5. Zero Trust Architecture

Zero Trust is a modern security model built around the principle:

"Never trust, always verify."

Key Rules:

- No device, user, or application is trusted by default.
- Continuous authentication and authorization required.
- Least-privilege access enforced everywhere.
- Micro-segmentation prevents lateral movement.

Zero Trust Pillars:

Feature	Purpose
Identity Verification	MFA, SSO, passwordless auth
Device Validation	Posture checks, endpoint compliance
Context-based Access	Location, time, risk-score awareness
Continuous Monitoring	Detect anomalies & revoke access dynamically

Table 6: Zero Trust Pillar

5.6. Summary

Basic security concepts such as the CIA triad, threat-risk vocabulary, defense-in-depth, and zero trust create the foundation of every secure architecture. Those principles guide SOC analysts in decision-making, incident evaluation, and designing resilient systems against cyber attacks. Mastering these concepts is the entry point into professional cybersecurity.

6. Security Operations Workflow

6.1. Introduction

Security Operations workflow defines the structured process by which a SOC identifies, analyzes, and responds to potential security threats. This framework is essential for maintaining organizational resilience and ensuring timely incident management. A well-defined SOC workflow reduces response time, minimizes impact, and strengthens defense against cyber attacks.

6.2. Stages of Security Operations Workflow

The workflow is typically divided into four key phases — Detection, Triage, Investigation, and Response. Each stage plays a critical role in managing and resolving security incidents effectively.

6.2.1. Detection

Detection is the initial stage where potential threats or suspicious activities are identified. Alerts are generated from security monitoring tools such as:

- **SIEM (Security Information and Event Management):** Aggregates logs from endpoints, servers, network devices, and clouds, then detects anomalies and correlation patterns.
- **EDR (Endpoint Detection and Response):** Monitors endpoint behavior, blocks malicious activity, and generates alerts for execution, persistence, or lateral movement.

Additional sources such as firewall logs, IDS/IPS, user reports, and threat intelligence feeds also contribute to detection.

6.2.2. Triage

Once alerts are generated, the next step is to classify and prioritize them. The SOC evaluates:

- Alert severity (Low/Medium/High/Critical)
- Business criticality of affected assets
- Presence of malicious indicators or suspicious patterns
- Volume or frequency of alerts

During triage, analysts determine whether the alert is a false positive or a true positive incident requiring escalation.

6.2.3. Investigation

Investigation involves deep analysis to uncover the cause, scope, and behavior of the threat. Analysts use logs, endpoint data, network captures, and threat intelligence to validate the event.

Key investigative activities include:

- Log correlation across systems and devices
- Searching for Indicators of Compromise (IOCs)
- Reviewing user actions and process behavior
- Checking file hashes, URLs, or IPs in threat intelligence databases

Tools commonly used: SIEM dashboards, EDR console, packet analyzers, sandbox analyzers, OSINT platforms, malware scanners.

6.2.4. Response

After confirming an incident, corrective actions are taken to stop the attack and eliminate the threat.

Response consists of three major components:

1. **Containment**:
   Isolate infected systems, block malicious IP/domains, disable compromised accounts to prevent escalation.
2. **Eradication:**
   Remove malware, delete persistence mechanisms, apply patches, terminate malicious processes.
3. **Recovery & Post-Incident Actions:**
   Restore system operations, change credentials, increase monitoring, and update playbooks to prevent recurrence.

6.3. Summary

The Security Operations Workflow ensures structured and efficient handling of cyber threats. Each stage Detection, Triage, Investigation, and Response contributes uniquely to safeguarding the organization. With strong detection mechanisms, smart alert prioritization, thorough investigative procedures, and timely response, organizations can minimize risk and strengthen their cybersecurity posture.

7. Incident Response Basics

7.1. Introduction

Incident Response (IR) is a structured process used to detect, manage, and resolve security incidents with minimal organizational impact. A well-defined Incident Response plan enables faster recovery, reduces damage, and helps prevent recurrence. IR is typically executed in sequential phases, known collectively as the Incident Response Lifecycle.

7.2. Incident Response Lifecycle

The NIST Incident Response Framework outlines six fundamental stages:

7.2.1. Preparation

Preparation ensures an organization is ready to respond to incidents when they occur, rather than reacting blindly. Activities in this phase include:

- Developing IR policies, procedures, and playbooks
- Setting up detection systems (SIEM, EDR, IDS/IPS)
- Training SOC analysts & employees for phishing awareness
- Conducting tabletop exercises and simulations
- Maintaining backups and secure configurations

7.2.2. Identification

This stage involves detecting and confirming that a security event is indeed an incident. Analysis is done using:

- SIEM alerts
- EDR detections
- Logs from network devices, servers, cloud platforms
- User-reported suspicious activity

Analysts validate whether the event is a **true positive**, classify impact level, and assign priority.

7.2.3. Containment

Containment prevents the incident from spreading or causing further damage. It is typically done in two approaches:

- **Short-term containment:**
  Isolate infected endpoints, block malicious IPs/domains, disable accounts immediately.
- **Long-term containment:**
  Implement temporary policies, patches, segmentation, increased monitoring.

7.2.4. Eradication

Here, the root cause of the attack is removed permanently from the system.

- Delete malware, backdoors, and persistence mechanisms
- Patch exploited vulnerabilities
- Remove unauthorized accounts or registry entries
- Apply AV/EDR cleaning tools

7.2.5. Recovery

Systems are restored to normal functioning after eradication. During recovery:

- Restore systems from backup or rebuild images
- Monitor environment for signs of reinfection
- Re-enable services gradually and safely

7.2.6. Lessons Learned

Post-incident review strengthens future defenses by analyzing what went right and what needs improvement.
Activities:

- Conduct internal review meeting
- Update IR playbooks and SIEM detection rules
- Improve patching, logging, monitoring coverage
- Document timelines, root cause, and remediation steps
- Train employees based on findings

7.3. Summary

Phase	Purpose	Key Actions
Preparation	Build capability before attacks occur	Policies, tools, training
Identification	Detect & confirm incidents	Alerts, log review
Containment	Stop spread of attack	Isolation, blocking
Eradication	Remove threat completely	Malware deletion, patching
Recovery	Restore operations safely	Backup restore, monitoring
Lessons Learned	Improve for future incidents	Review, documentation

Table 7: Lifecycle

Incident Response is not just about stopping attacks it is about continuous improvement. By following each lifecycle phase with discipline, organizations can minimize breach damage, restore operations quickly, and strengthen security posture over time. A mature IR program blends preparation, automation, skilled analysts, and regular refinement of defenses.

8. Log Analysis Practice

8.1. Objective

Log analysis is a fundamental skill in cybersecurity, enabling analysts to detect suspicious activity and investigate potential incidents. Windows Event Viewer provides detailed records of system and security events, such as failed login attempts or new service installations, which can reveal brute-force attacks or unauthorized changes. Browser history analysis adds another layer of visibility, helping identify visits to malicious domains. By combining built-in tools like Event Viewer and wevtutil with specialized utilities such as Eric Zimmerman's LECmd, practitioners can gain hands-on experience in detecting threats and understanding attacker behavior.

8.2. Log Analysis Using Windows Event Viewer

- Detect failed login attempts
- Identify malicious account access activity
- Observe brute-force attack patterns

Step 1 Open Event Viewer

Win + R > eventvwr.msc > Enter

Windows Logs > Security

Step 2 Filter for Key Event ID 4625

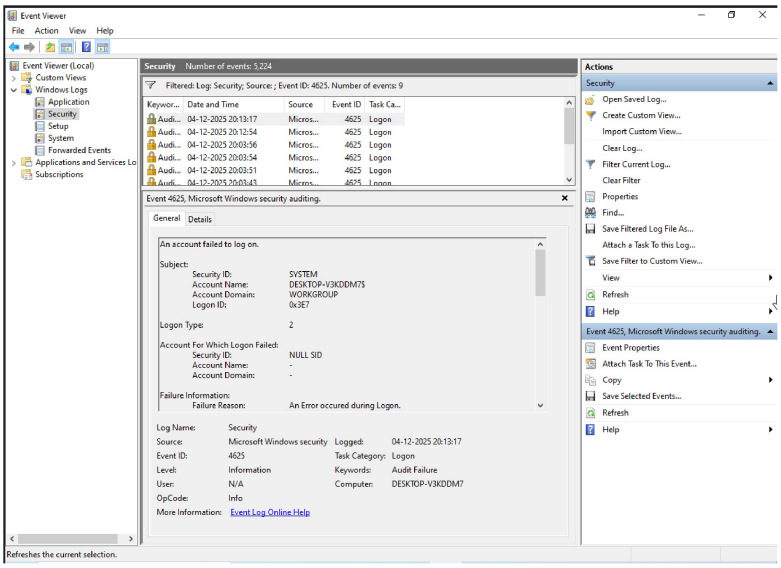

Image 3: Event Viewer

8.3. Browser History Analysis

Identify malicious browsing, phishing attempts, or C2 communication.

Step 1 Download Eric Zimmerman Tools

Extract tools into a folder such as:

C:\Tools\Zimmerman

Step 2 Retrieve Chrome History Database

Chrome history location:

C:\Users\gyane\AppData\Local\Google\Chrome\User Data\Default\History

Copy the file to analysis folder (close Chrome first).

Step 3 Parse Browser History with SQLECmd

SQLECmd.exe -d "C:\Users\Ganesh\AppData\Local\Google\Chrome\User Data\Default" --csv "output"

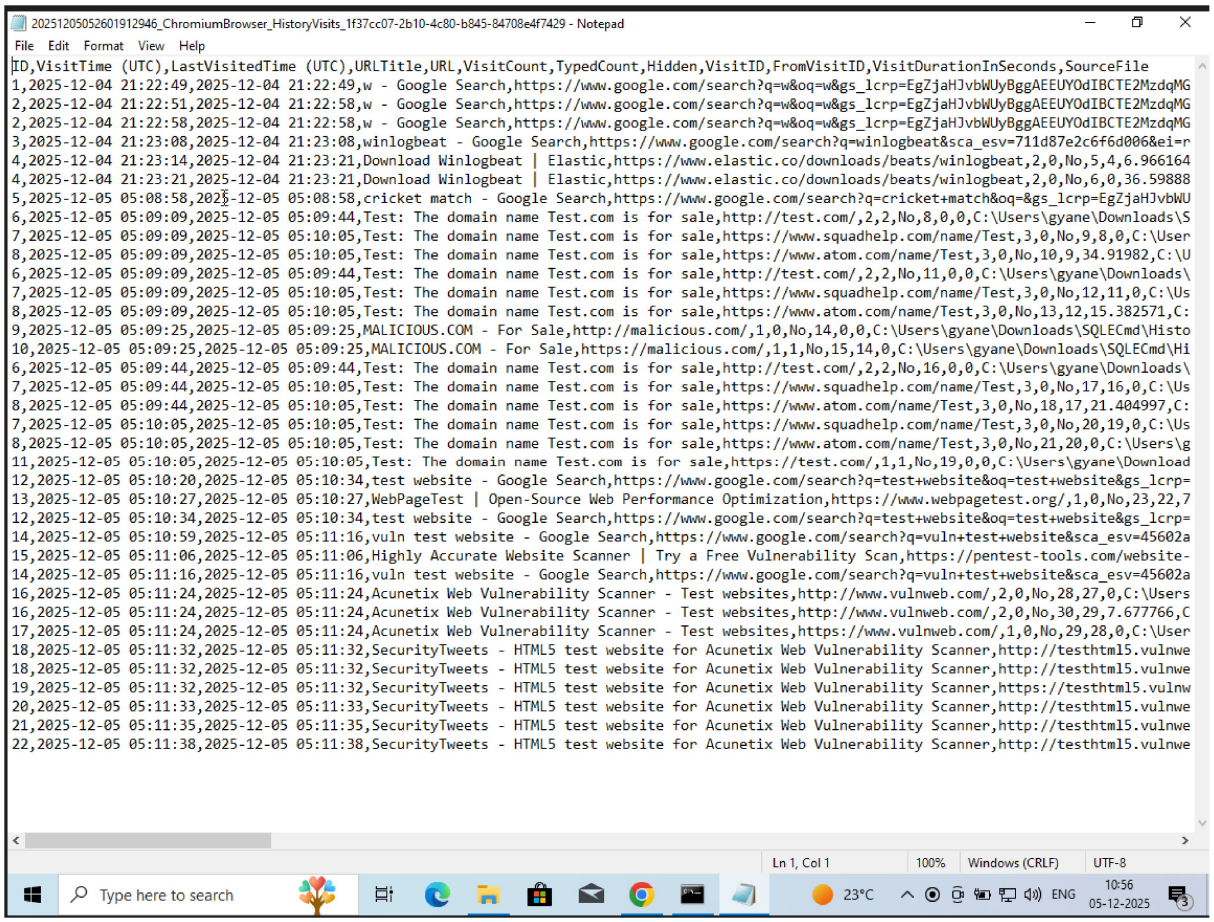

Image 4: Chrome History

8.4. Summary

Through these exercises, learners develop practical expertise in monitoring and analyzing system activity. Filtering for Event IDs such as 4625 and 7045 highlights common attack patterns, while parsing browser history demonstrates how adversaries may exploit user activity. Exporting logs, detecting brute-force attempts, and identifying malicious URLs provide a realistic foundation for incident response. Together, these practices reinforce the importance of log analysis as a cornerstone of security operations and prepare analysts to respond effectively to real-world threats.

9. Set Up Monitoring Dashboards

9.1. Objective

Create dashboards to visualize security events such as Top Source IPs, Critical Event IDs, and alert patterns using Kibana. These dashboards improve SOC visibility and accelerate incident analysis.

9.2. Dashboard Setup

9.2.1. Create Dashboards in Kibana

Step 1 Open Kibana UI

Access: *http://localhost:5601*

Analytics > Dashboards

Create Visualizations

- Create Visualization for Top 10 Source IPs
- Visualize Frequency of Critical Event IDs

Step 2 Build Dashboard

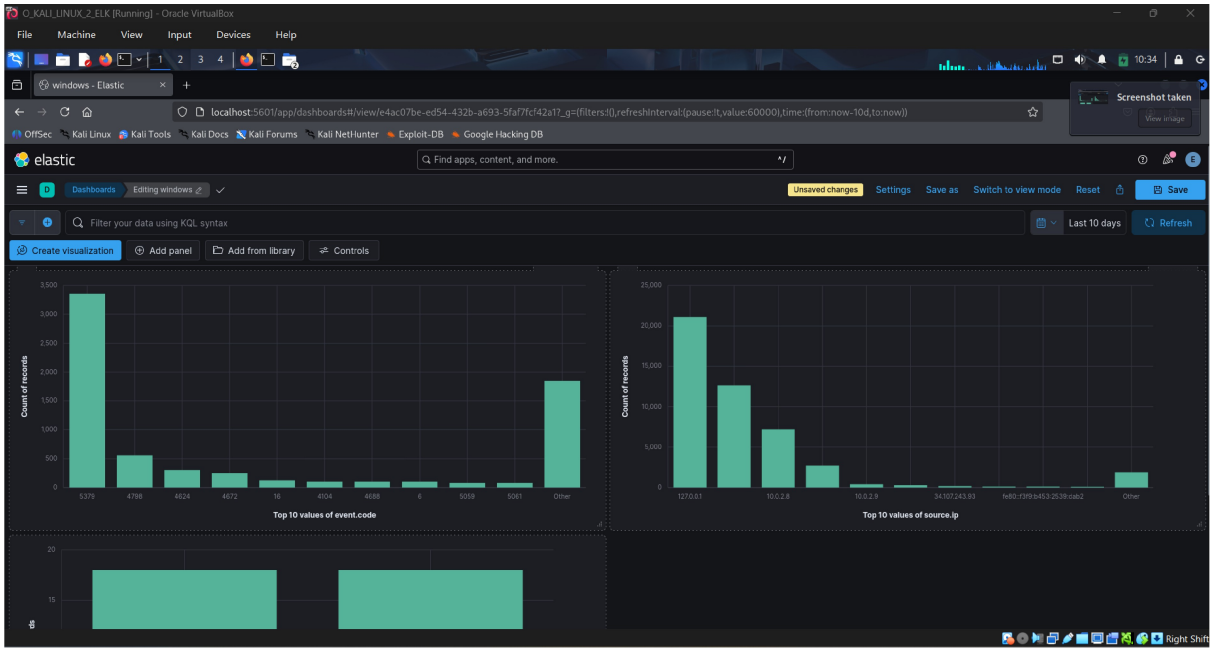

Image 5: Dashboard

9.2.2. Prebuilt Dashboard

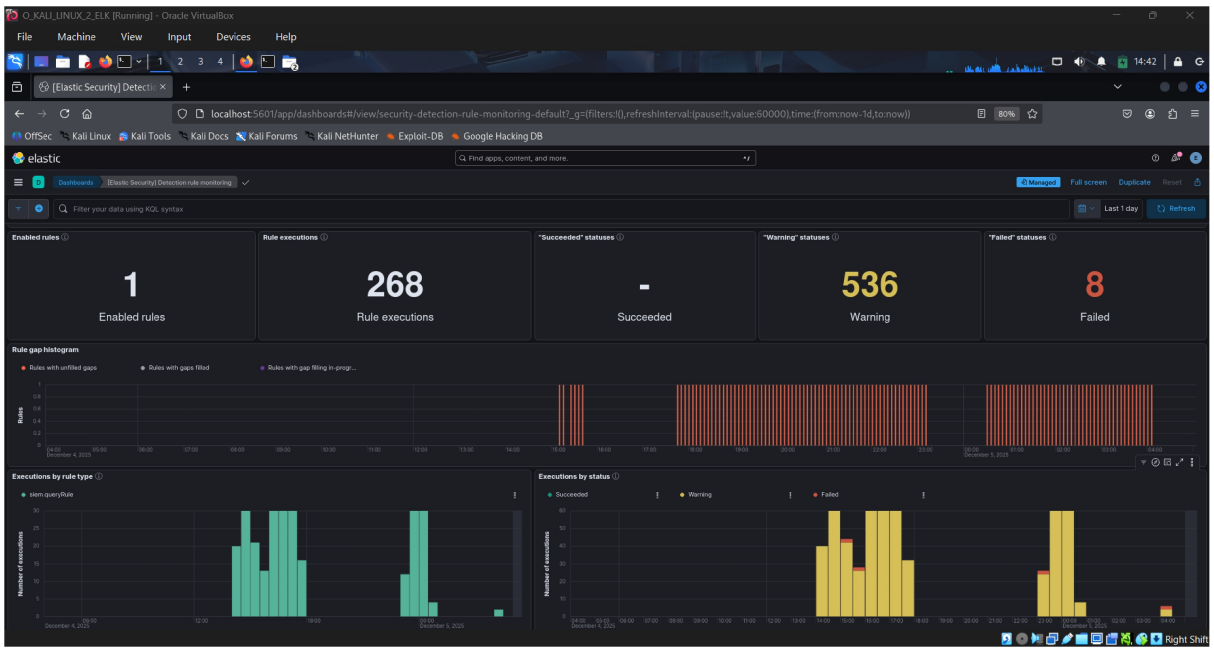

Image 6: Dashboard

9.3. Summary

Building monitoring dashboards in Kibana or Grafana transforms raw security data into actionable insights. By visualizing the top 10 source IPs generating alerts, analysts can quickly identify potential attackers or misconfigured systems flooding the environment. Tracking the frequency of critical Event IDs highlights recurring issues and helps prioritize incident response. These dashboards not only improve situational awareness but also streamline threat detection, making it easier for security teams to spot anomalies, investigate patterns, and strengthen defenses in real time.

10. Configure Alert Rules

10.1. Introduction

Configuring alert rules is a critical step in transforming raw log data into actionable security intelligence. By defining thresholds and conditions, analysts can automatically detect suspicious activity such as brute-force login attempts. Tools like Elastic SIEM and Wazuh provide flexible rule engines that allow customization based on organizational needs. Practicing with failed SSH login scenarios helps learners understand how alerts are triggered, validated, and refined, ensuring that detection logic is both effective and reliable.

10.2. Elastic SIEM Alert Rule Configuration

Detect abnormal authentication failures specifically 5 or more failed login attempts within 5 minutes.

Inputs Used

Parameter	Value
Rule Name	Detect 5+ failed logins in 5 minutes
Index Pattern	security-login-*
Condition	count ≥ 5
Time Window	5 minutes
Action	Generate SIEM alert

Table 8: Rule

Setup Steps

1. Log into **Kibana > Security > Rules > Create New Rule**
2. Select **Custom Query**
3. Configure KQL query:

event.action: "failed-login" OR authentication.failure:true

Set threshold condition:

Trigger alert when: count() is above 5

From: Last 5 minutes

Index: security-login-*

Testing Method

Execute repeated failed SSH login attempts:

for i in {1..7}; do ssh ukali@10.0.2.8; done

10.3. Wazuh Custom Alert Rule

Detect 3 or more failed SSH login attempts within 2 minutes and generate Wazuh alerts for analyst response.

Rule Creation Steps

Open Wazuh rules file and add the rule to it:

sudo nano /var/ossec/etc/rules/local_rules.xml

Save & restart Wazuh:

sudo systemctl restart wazuh-manager

Testing Simulation

Perform failed login attempts:

ssh root@10.0.2.8

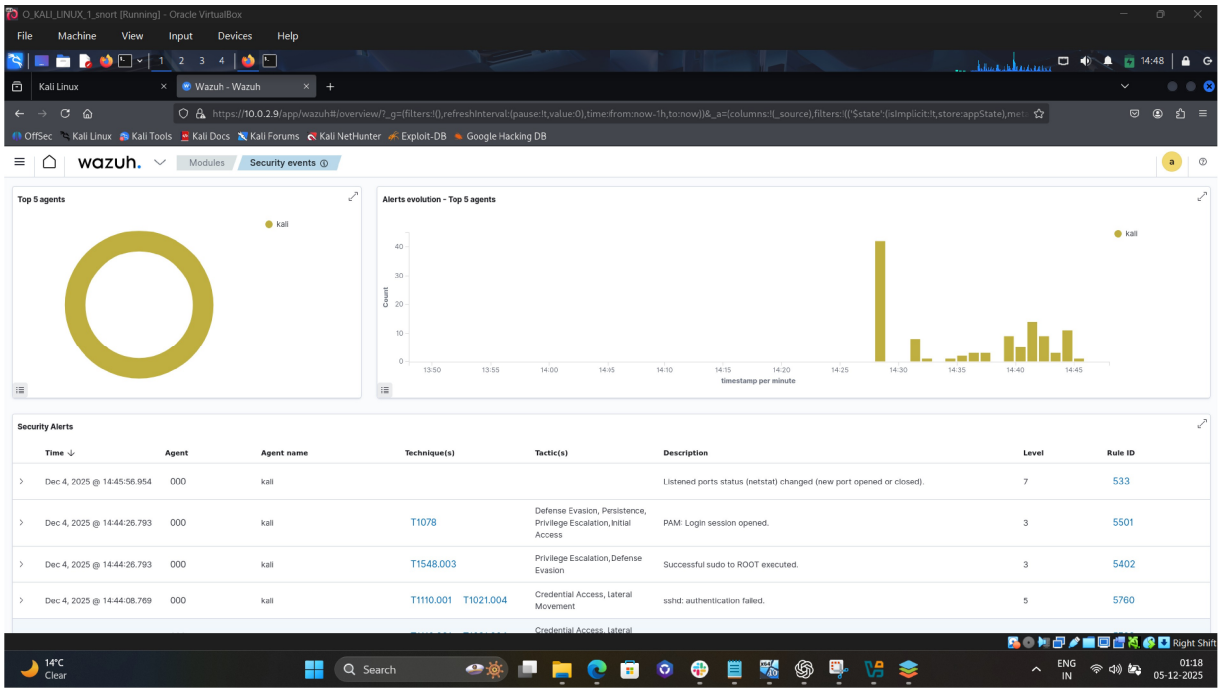

Image 7: Wazuh Dashboard

```
┌──(kali㉿kali)-[~]
└─$ sudo tail -f /var/ossec/logs/alerts/alerts.log

2025-12-04T14:42:32.223027-05:00 kali unix_chkpwd[62792]: password check failed for user (kali)

** Alert 1764877354.80917: - syslog,sshd,authentication_failed,gdpr_IV_35.7.d,gdpr_IV_32.2,gpg13
_7.1,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,pci_dss_10.2.4,pci_dss_10.2.5,tsc_CC6.1,
tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,
2025 Dec 04 14:42:34 kali→/var/log/auth.log
Rule: 5760 (level 5) → 'sshd: authentication failed.'
Src IP: 10.0.2.8
Src Port: 57592
User: kali
2025-12-04T14:42:34.287699-05:00 kali sshd-session[62498]: Failed password for kali from 10.0.2.
8 port 57592 ssh2

** Alert 1764877364.81399: - syslog,sudo,pci_dss_10.2.5,pci_dss_10.2.2,gpg13_7.6,gpg13_7.8,gpg13
_7.13,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,nist_800_53_AC.6,tsc_CC6.8
,tsc_CC7.2,tsc_CC7.3,
2025 Dec 04 14:42:44 kali→/var/log/auth.log
Rule: 5402 (level 3) → 'Successful sudo to ROOT executed.'
User: root
2025-12-04T14:42:44.484379-05:00 kali sudo:     kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ;
COMMAND=/usr/bin/tail -f /var/ossec/logs/alerts/alerts.log
tty: pts/0
pwd: /home/kali
command: /usr/bin/tail -f /var/ossec/logs/alerts/alerts.log

** Alert 1764877364.81972: - pam,syslog,authentication_success,pci_dss_10.2.5,gpg13_7.8,gpg13_7.
9,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,
2025 Dec 04 14:42:44 kali→/var/log/auth.log
Rule: 5501 (level 3) → 'PAM: Login session opened.'
User: root(uid=0)
2025-12-04T14:42:44.490474-05:00 kali sudo: pam_unix(sudo:session): session opened for user root
(uid=0) by kali(uid=1000)
uid: 1000
```

Image 8: Alert Logs

10.4. Summary

Feature	Elastic SIEM	Wazuh Rule
Threshold	5 attempts in 5 mins	3 attempts in 2 mins
Detection Type	Query-based	Log-based rule engine
Response Time	Fast & visual alerts	Custom granular detection
Best For	Enterprise SIEM	Endpoint threat analytics

Table 9: Summary

Both alerting mechanisms successfully detected SSH brute-force behavior. Elastic allowed broader threat visualization, while Wazuh gave fine-tuned detection via custom rule logic. Combined, they offer strong defense against authentication-related attacks.

11. Conclusion

Through this learning path, we have gained both theoretical and practical cybersecurity skills. We now understand SOC fundamentals, analyst roles, monitoring objectives, log management processes, and key security concepts such as the CIA triad, defense-in-depth, and incident response workflows. We have also practiced hands-on tasks including log analysis in Windows Event Viewer, browser history parsing with Zimmerman's tools, documenting events, building dashboards in Kibana, and configuring alert rules in Elastic SIEM and Wazuh. Together, these experiences have strengthened my ability to detect, investigate, and respond to threats effectively, reinforcing both technical expertise and operational readiness.

12. References

- IBM - Security Operations Center (SOC) https://www.ibm.com/think/topics/security-operations-center
- Wikipedia - Security Information and Event Management (SIEM) https://en.wikipedia.org/wiki/Security_information_and_event_management
- Elastic SIEM https://www.elastic.co/siem
- Syslog https://www.rfc-editor.org/rfc/rfc5424
- Logstash https://www.elastic.co/logstash
- MITRE ATT&CK https://attack.mitre.org
- NIST Cybersecurity Framework https://www.nist.gov/cyberframework
- Wazuh Rule Syntax https://documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/rules.html
- Detect Brute Force (Wazuh) https://documentation.wazuh.com/current/proof-of-concept-guide/detect-brute-force-attack.html
- Snort IDS https://www.snort.org/downloads
- Nessus https://www.tenable.com/products/nessus
- Rapid7 - Metasploitable 2 https://docs.rapid7.com/metasploit/metasploitable-2/