TRYHACKME
SOC LEVEL 01 DOCUMENTATION

Raghavi Preeti Ayyathurai Jaya

February 03, 2026

Table of Contents

1 Introduction

I would like to start this introduction by pointing out two concepts that were emphasized repeatedly throughout most of my classes: **zero trust** and **visibility: you can't protect what you don't know about**.

The platform that consistently reinforced these principles through practical, hands-on exercises was **TryHackMe**. While learning through TryHackMe, it became clear that these concepts were reinforced in almost every room.

TryHackMe provides a structured environment where one can gain meaningful hands-on experience, and the first learning path I chose to pursue on the platform was the **SOC Level 01** path.

In this report, I aim to bring together the key topics I have learned so far, highlight the important insights gained throughout the path, and outline the tools and frameworks introduced along the way.

1.1 Scope of Knowledge Covered

This document is scoped to the SOC Level 01 learning path on TryHackMe. The scope reflects the full set of defensive security topics and SOC-oriented domains introduced within this path. The knowledge areas included within this scope are:

- Blue Team fundamentals and the role of the SOC
- SOC analyst responsibilities, workflows, and escalation processes
- Core SOC security solutions, including SIEM, EDR, and SOAR
- Cyber defense frameworks used to understand adversarial behavior
- Phishing analysis and investigation techniques
- Network traffic analysis
- Network security monitoring
- Web security monitoring
- Windows security monitoring
- Linux security monitoring
- Malware concepts relevant to SOC investigations

- Threat intelligence usage and analysis tools
- SIEM-based triage and alert investigation
- SOC Level 1 capstone challenges and applied incident investigations

2 Insights Gathered

The SOC Level 01 learning path begins by establishing foundational cybersecurity concepts and then progressively transitions into role-based defensive security perspectives. Early modules introduce core cybersecurity fundamentals before pivoting toward the various security roles that professionals may choose to specialize in, providing context for how different functions contribute to an organization's overall security posture.

The path then explores key defensive frameworks, including the **Cyber Kill Chain** and the **MITRE ATT&CK** and **MITRE D3FEND** frameworks. These frameworks illustrate how a threat actor typically progresses from reconnaissance to achieving their objectives, helping reinforce the importance of visibility at each stage of an attack. Understanding these models clarified how adversarial behavior can be mapped, detected, and disrupted.

Fundamental risk concepts were also emphasized, including the distinction between **vulnerabilities**, **threats**, and **risk**, along with methods for prioritizing threats based on potential impact and likelihood. This provided a structured way to think about defensive decision-making rather than treating all alerts or issues with equal urgency.

The learning path then shifts toward detection and response by introducing first-line defensive tools commonly used in SOC environments, including **SIEM**, **EDR**, and **SOAR** platforms. These tools were presented as core components for centralized visibility, alert correlation, and response orchestration.

Phishing analysis was explored in depth, focusing on how to identify phishing attempts through common indicators and behavioral patterns. The module also covered email authentication mechanisms such as **SPF**, **DKIM**, and **DMARC**, and demonstrated how failures in these verification mechanisms can signal suspicious or malicious activity.

Network security modules introduced hands-on traffic analysis using **Wireshark**, emphasizing packet-level inspection and the use of filters to isolate relevant traffic. Various scanning techniques commonly used by threat actors were examined, along with methods to identify spoofing attempts and **man-in-the-middle**

(MITM) attacks through traffic analysis.

Next, the path expanded into network-based detection mechanisms by introducing **IDS** and **IPS** concepts, followed by practical exposure to writing detection rules using **Snort**, reinforcing how signatures and rules can be used to detect malicious activity at the network level.

The web security module extended the understanding of attack surface from the network layer to the application layer, where malicious activity often closely resembles legitimate user behavior. A key insight was the importance of correlating subtle indicators such as **unusual request patterns**, **suspicious User-Agent strings**, **abnormal query parameters**, and **missing referrer headers**, rather than relying on any single signal in isolation.

The module also highlighted that effective detection of web shells and application-layer denial-of-service attacks requires visibility across multiple layers, including application logs, file system activity, network behavior, and host-level telemetry. This reinforced the idea that web-based threats are best identified through contextual analysis and correlation, especially in high-noise environments where attackers deliberately attempt to blend into normal traffic.

2.1 Expanded Host-Based Detection

As the learning path progressed into specialized monitoring, the focus shifted toward host-level telemetry and the limitations of default logging.

2.1.1 Windows Security Monitoring & LotL

Windows detection strategies centered on identifying ”Living off the Land” (LotL) techniques where attackers use legitimate system tools for malicious purposes.

- **Discovery Patterns:** Adversaries frequently execute commands such as **whoami**, **net user**, and **systeminfo** immediately after gaining access to map privileges and system architecture.
- **Network & App Discovery:** Identification of internal network positioning is often attempted via **ipconfig /all** and **netstat -ano**, while **tasklist /v** is used to identify active security software.
- **Ingress Tool Transfer:** Built-in utilities like **certutil.exe**, **curl.exe**, and PowerShell’s **Invoke-WebRequest** are monitored as primary vectors for downloading external payloads.

- **Telemetry Enhancement:** The use of Sysmon was emphasized to provide granular visibility into process creation (Event ID 1) and network connections (Event ID 3), which standard Windows Event Logs may miss.

2.1.2 Linux Security Monitoring & Persistence

Linux detection requires a move away from easily manipulated shell history toward kernel-level auditing.

- **Logging Limitations:** Standard Bash history (.bash_history) is unreliable for SOC investigations as it can be bypassed by leading spaces, running commands in scripts, or switching to alternative shells like /bin/sh.
- **The Audit Framework (Auditd):** By implementing auditd, a SOC can monitor specific syscalls such as execve (process execution) and connect (network activity) regardless of the shell used.
- **Syscall Analysis:** Practical investigation involves using ausearch to correlate type=SYSCALL with type=CWD to identify what was executed and from which directory.
- **Persistence Mechanisms:** Adversaries often maintain access by modifying Cron Jobs in /etc/crontab or creating malicious Systemd services in /etc/systemd/system/.
- **Detection of Persistence:** Monitoring for unauthorized file creations in system directories and tracking the execution of management tools like crontab -e or systemctl is critical for early eviction.

2.2 Proactive Threat Hunting

Threat hunting was introduced as a proactive maturity level beyond standard alert triage.

- **Hypothesis-Driven Hunting:** Instead of waiting for an alert, Analysts develop a hypothesis based on current threat intelligence or known TTPs.
- **The Hunting Workflow:** The process involves formulating a theory, identifying the necessary telemetry (like Sysmon or Auditd logs), and searching for ”weak signals” that indicate a bypass of automated controls.
- **Tool Enrichment:** Leveraging threat intelligence platforms helps SOC analysts enrich their findings and prioritize hunts against the most relevant adversary groups.

A Tools and Frameworks Referenced

Category	Tool / Framework	Purpose
Security Principles	Zero Trust	Security model emphasizing continuous verification and minimal implicit trust
Security Principles	Visibility	Emphasizes awareness of assets, activity, and exposure to enable effective protection
Adversary Framework	Cyber Kill Chain	Describes stages of an attack from reconnaissance to objective completion
Adversary Framework	MITRE ATT&CK	Knowledge base of adversary tactics and techniques
Defensive Framework	MITRE D3FEND	Maps defensive techniques to adversary behaviors
SOC Platform	SIEM	Centralized log collection, correlation, and alerting
Endpoint Security	EDR	Endpoint monitoring and threat detection
Automation Platform	SOAR	Security orchestration, automation, and response
Email Security	SPF	Email sender authentication mechanism
Email Security	DKIM	Cryptographic email authentication
Email Security	DMARC	Policy framework for email authentication enforcement
Network Tool	Wireshark	Packet-level network traffic analysis
Network Defense	IDS	Intrusion Detection System
Network Defense	IPS	Intrusion Prevention System
Detection Engine	Snort	Signature-based network intrusion detection and rule creation
Endpoint Telemetry	Sysmon	Provides advanced Windows logging for process and network events
Linux Auditing	Auditd	Monitors system calls to overcome shell logging limitations
Log Analysis	ausearch	A tool for searching and parsing Linux audit logs for investigation
Persistence Target	Cron / Systemd	Common Linux mechanisms used by attackers for survival after reboot
Threat Method	LotL(Living off the Land)	Using legitimate binaries to perform malicious actions

Table 1: Tools and Frameworks Referenced in the SOC Level 01 Learning Path