# TryHackMe Advent of Cyber 2025

# Day 4 Challenge Report

*Artificial Intelligence in Cybersecurity*

## 1. Executive Summary

This report documents the completion of Day 4 of the TryHackMe Advent of Cyber 2025 event. The challenge explored the practical applications of Artificial Intelligence in cybersecurity across offensive security, defensive security, and software development. Through four interactive stages, demonstrated how AI assists in exploit generation, log analysis, and vulnerability identification.

## 2. Challenge Overview

**Objective:** Understand AI's role in cybersecurity, utilize AI assistants for red team, blue team, and software security tasks, and recognize critical considerations when deploying AI in security contexts.

**Interface:** Interactive AI chatbot accessible at http://{VM_IP}

## 3. AI in Cybersecurity: Core Capabilities

AI has evolved from a simple search alternative to an integral component of cybersecurity workflows, enhancing productivity and handling time-consuming tasks that previously required significant manual effort.

| AI Feature | Cybersecurity Application |
|---|---|
| Large-Scale Data Processing | Analyzes vast datasets from multiple sources simultaneously (system logs, network logs, endpoint telemetry) |
| Behavior Analysis | Establishes baseline behavior patterns and flags anomalies that deviate from normal activity |
| Generative AI | Summarizes complex event chains and provides contextual analysis for security incidents |

### 3.1 Defensive Security (Blue Team)

AI agents accelerate detection, investigation, and response capabilities in defensive security operations. These automated assistants continuously process telemetry data and contextualize alerts, improving both speed and reliability.

**Key Applications:**

- AI-assisted firewalling and intrusion detection systems
- Automated threat response (device isolation, email blocking, login flagging)
- Real-time alert contextualization and correlation

### 3.2 Offensive Security (Red Team)

AI significantly reduces the time spent on laborious reconnaissance and information gathering tasks, allowing penetration testers to focus on activities requiring human expertise and critical thinking.

**Key Applications:**

- OSINT collection and analysis
- Scanner output analysis and filtering
- Attack surface mapping
- Exploit script generation and adaptation

### 3.3 Software Security

While AI-driven software development presents risks, AI proves valuable when used appropriately in the development lifecycle, particularly for vulnerability detection and code review.

**Key Applications:**

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Code review and vulnerability identification
- Development consultation and problem-solving

***Notable Observation:*** *AI excels at identifying vulnerabilities but struggles with writing inherently secure code.*

## 4. Practical Implementation

### 4.1 Stage 1: Red Team - Exploit Generation

Utilized AI to generate a Python exploit script targeting a SQL injection vulnerability in a web application login form.

**Vulnerability Details:**

- **Type:** SQL Injection
- **Location:** Username field in login form
- **Exploit Payload:** alice' OR 1=1 -- -
- **Method:** String concatenation enabling arbitrary SQL query injection

The AI assistant generated a complete Python script using the requests library to automate the SQL injection attack against the vulnerable endpoint (http://MACHINE_IP:5000/login.php).

**Flag Captured: THM{SQLI_EXPLOIT}**

## 4.2 Stage 2: Blue Team - Log Analysis

Leveraged AI to analyze web server access logs and identify the SQL injection attack executed in Stage 1.

**Log Entry Analyzed:**

```
198.51.100.22 - - [03/Oct/2025:09:03:11 +0100] "POST /login.php HTTP/1.1" 200 642 "-"
"python-requests/2.31.0" "username=alice%27+OR+1%3D1+--+-&password=test"
```

**AI Analysis Results:**

1. **Source IP:** 198.51.100.22 identified as attack origin
2. **Timestamp:** 3 October 2025 at 09:03
3. **Target:** login.php endpoint
4. **Attack Vector:** SQL injection via username parameter
5. **User Agent:** python-requests/2.31.0 (automated tool signature)
6. **Payload:** URL-encoded SQL injection string

**Defensive Value:**

- Confirms application vulnerability to SQL injection
- Highlights importance of input validation and sanitization
- Demonstrates value of monitoring user input in password parameters

## 4.3 Stage 3: Software Security - Vulnerability Identification

Employed AI to review the application's source code and identify the vulnerability exploited in previous stages.

**Vulnerable Code Pattern:**

```
$user = $_POST['username'] ?? '';
$pass = $_POST['password'] ?? '';
```

**AI Vulnerability Assessment:**

The ?? (null coalescing) operator provides safe default values, but the code fails to sanitize user input before processing. This lack of input validation allows attackers to inject malicious SQL code through the username and password parameters.

**Recommended Remediation:**

7. **Input Sanitization:** Validate and sanitize all user inputs before processing
8. **Prepared Statements:** Use parameterized queries to prevent SQL injection
9. **Input Validation:** Implement validation before storing variables
10. **Output Encoding:** Prevent XSS attacks through proper output encoding

**Final Flag Captured: THM{AI_MANIA}**

# 5. Critical Considerations for AI in Cybersecurity

While AI offers significant benefits, deployment in cybersecurity contexts requires careful consideration of inherent limitations and risks.

## 5.1 Operational Risks

- **Output Verification:** AI-generated content cannot be assumed 100% correct - manual verification essential
- **Offensive Testing Caution:** AI may cause service disruption through race conditions or system overwhelm
- **Ownership:** Organizations don't own AI-generated output

### 5.2 Data and Model Concerns

- **Training Data Quality:** AI effectiveness depends on training data quality and relevance
- **Decision Transparency:** Black-box nature makes decision reasoning difficult to understand
- **Reliability:** Performance may degrade when encountering unexpected scenarios
- **Privacy and Security:** Sensitive data handling and model security require careful management

### 5.3 Professional Expectations

Organizations increasingly expect security professionals to demonstrate experience with AI tools rather than avoidance. AI should be viewed as a force multiplier that handles tedious tasks, allowing humans to focus on complex analysis and decision-making that requires expertise and judgment.

## 6. Key Takeaways

- AI transforms cybersecurity workflows by automating time-consuming tasks
- Blue teams benefit from enhanced detection, analysis, and automated response
- Red teams leverage AI for reconnaissance and exploit development
- AI excels at vulnerability identification but struggles with secure code generation
- Output verification remains critical - AI is not infallible
- Organizations expect AI proficiency, not avoidance
- Careful consideration required for deployment in production environments

## 7. Conclusion

Day 4 of the TryHackMe Advent of Cyber 2025 provided practical exposure to AI applications across the cybersecurity spectrum. Through hands-on interaction with AI assistants, successfully demonstrated capabilities in exploit generation, log analysis, and vulnerability identification.

The challenge reinforced that AI serves as a powerful force multiplier rather than a replacement for human expertise. While AI accelerates routine tasks and provides valuable insights, human oversight, critical thinking, and verification remain essential components of effective cybersecurity operations.

**Challenge Status:** **COMPLETED ✓**