OffSec AI Red Teamer

Exam Report

OSID: OS-XXXXX
student@example.com

September 1, 2026

v1.0

Table of Contents

1 OffSec AI Red Teamer Exam Report 3

1.1 Introduction 3

1.2 Objective 3

1.3 Requirements 3

2 Executive Summary 4

2.1 Overview 4

2.2 High-Level Attack Path 4

3 Chain 1 5

3.1 HOSTNAME (IP: X.X.X.X) 5

3.1.1 Compromise 5

3.1.2 Privilege Escalation 5

3.1.3 Proof 5

1 OffSec AI Red Teamer Exam Report

1.1 Introduction

The OffSec AI Red Teamer (OSAI) exam report contains all efforts that were conducted in order to pass the OffSec certification examination. This report should contain all items that were used to pass the exam, and it will be graded from a standpoint of correctness and completeness across all aspects of the exam. The purpose of this report is to ensure that the candidate has a full understanding of red team methodologies in AI-enabled environments, as well as the technical knowledge required to meet the qualifications for the OffSec AI Red Teamer certification.

1.2 Objective

The objective of this assessment is to perform a hands-on red team engagement against an AI-enabled enterprise environment. The candidate is tasked with identifying and exploiting both AI-focused and traditional attack vectors across two independent attack chains, a standalone host, and a shared Domain Controller, demonstrating a complete exploitation path from external access through to internal compromise.

Each chain begins from a single publicly accessible entry point that allows a pivot into the internal network and spans three machines; both chains converge on the Domain Controller. A separate standalone host with an AI-focused vector becomes reachable once a foothold is established from either chain. A total of 100 points is available across the environment, and the candidate must obtain at least 75 points to pass.

1.3 Requirements

The candidate is required to write a report describing the exploitation process for each target in the environment. The report must document all attacks, including every step taken, all commands issued, any code or scripts written, and the relevant console output. Where an existing script or exploit is used, the candidate must provide a link to its source or a copy of the code, along with documentation of any modifications made to it. Each stage of the attack should be supported by screenshots showing the various steps and stages of the exploitation process.

In addition, the report must include a summary and overview of the vulnerabilities identified, the attacks performed, and the overall exploitation process. The documentation should be thorough enough that the attacks can be replicated step-by-step, via a copy/paste approach, by a technically competent reader.

Because effective use of AI is a core component of this assessment, any AI tooling used during the engagement — including prompts, model interactions, and AI-assisted payload generation — should be documented to the same standard as any other tool or technique.

2 Executive Summary

2.1 Overview

TODO This section should provide a high-level, non-technical overview of the engagement suitable for a management audience. Summarize the scope of the assessment, the overall outcome, and the most significant vulnerabilities discovered across the AI-enabled environment.

2.2 High-Level Attack Path

TODO The following summarizes the path taken to compromise the environment. Replace each item with your own attack narrative.

1. **Chain 1** — the externally accessible entry point was compromised and used to pivot into the internal network, followed by lateral movement and privilege escalation across the two subsequent hosts in the chain.
2. **Chain 2** — a second, independent external entry point was compromised, providing an alternate route into the internal network and onward to its two subsequent hosts.
3. **Standalone host** — reached from the internal network once a foothold was established, this AI-focused target was compromised via its exposed AI vector.
4. **Domain Controller** — both chains converged on the Domain Controller, which was compromised to retrieve the shared endgame proof.

3 Chain 1

3.1 HOSTNAME (IP: X.X.X.X)

3.1.1 Compromise

TODO Document the enumeration that revealed the attack surface (port scans, service/version detection, web/API discovery, exposed AI components, etc.). Describe the vulnerability and the full exploitation process in copy/paste-reproducible detail. For AI vectors, document the prompts, payloads, or model-interaction steps used. Link or paste any script/exploit used and note every modification you made to it.

For the Domain Controller, document how it was compromised from this chain. Both chains converge here; the proof flag can be submitted only once even if both chains are completed.

Include the exact commands and the relevant output or screenshots.

```
# example nmap -sC -sV -p- <TARGET_IP>
```

3.1.2 Privilege Escalation

TODO If a privilege-escalation step was required to reach the flag, document the enumeration that identified it and the exact steps used to escalate. Include before/after proof of privilege (e.g., `whoami`).

3.1.3 Proof

TODO Show the contents of the flag file and a screenshot proving access on the target. Traditional-vector flags are named `trad_vector.txt`, AI-vector flags `ai_vector.txt`; both live in the compromised user's home directory or `C:\Users\Administrator\Desktop\` (Windows) / `/root/` (Linux). Domain Controller proof is `proof.txt` in `C:\Users\Public\`.

```
type C:\Users\Administrator\Desktop\ai_vector.txt   # Windows
cat /root/ai_vector.txt                            # Linux
type C:\Users\Public\proof.txt                      # Domain Controller
```

[paste flag value and screenshot here]

End of Report

This report was rendered
by SysReptor with
♥