SECURITY ASSESSMENT

Ra Report

Submitted to: Sprints

Security Analyst:

Mohamed Awad Mohamed
Mary Alfons Shokry
Mohamed Gamal
Yomna Mohamed Abdelfatah
Mahmoud Mohamed Ismail

Date of Testing: 22/10/2024
Date of Report Delivery: 24/10/2024

Table of Contents

Contents

Security Engagement Summary

Engagement Overview

This engagement was requested by the TryHackMe Security Team to assess the security vulnerabilities within the Ra environment. The goal of the engagement was to identify potential security weaknesses, provide remediation guidance, and improve the overall security posture. The assessment was conducted using both automated and manual testing methods to ensure thorough coverage of possible attack vectors.

Scope

The scope of this engagement included the entire Ra TryHackMe room environment. The focus was on identifying vulnerabilities related to weak cryptographic implementations, hidden files, and misconfigured services. This scope was chosen to address the most critical security threats specific to this machine.

Executive Risk Analysis

The scope of this engagement included the entire Ra TryHackMe room environment. The focus was on identifying vulnerabilities related to weak cryptographic implementations, hidden files, and misconfigured services. This scope was chosen to address the most critical security threats specific to this machine.

Executive Recommendation

It is recommended to prioritize addressing the weak cryptographic implementations, as they represent the highest risk. Remediation should also focus on fixing service misconfigurations and securing hidden files to prevent data leakage.

Significant Vulnerability Summary

Medium Risk Vulnerabilities

- Weak Hashing Algorithm
- Hidden Files Exposed

Low Risk Vulnerabilities

- Misconfigured Services

Significant Vulnerability Detail

Risk Level: MEDIUM

The Ra environment was found to be using weak cryptographic hashing algorithms (such as MD5 or SHA-1), which are susceptible to collisions and brute force attacks. This vulnerability was validated through manual testing by attempting to crack the hashes using common tools. It is recommended to switch to stronger hashing algorithms, such as SHA-256 or bcrypt, to protect sensitive information.

Hidden Files Exposed

Risk Level: MEDIUM

Several hidden files were discovered within the environment that contained sensitive information. These files could be easily accessed, allowing attackers to retrieve critical information. It is recommended to properly secure or remove unnecessary hidden files, and apply appropriate file permissions to prevent unauthorized access.

Misconfigured Services

Risk Level: LOW

Misconfigured services were identified, which allowed for unnecessary access to certain functionalities that could be exploited by attackers. While the risk is low, it is advisable to review and correct service settings to follow best practices for security.

Methodology

The following steps were taken during the assessment of the Ra lab:

1. Reconnaissance:

   - Nmap was used to scan the target machine for open ports and services.
   - Command: nmap -sV -A [target IP]
   - Result: Open ports such as HTTP (80) and SSH (22) were discovered.

2. Service Enumeration:

   - Enumeration of HTTP and SSH services was performed to gather further details.
   - HTTP responses were manually inspected to discover hidden files containing sensitive data.

3. Password Cracking:

- John the Ripper was used to crack weak hashes retrieved from sensitive files.
- Command: john [hash file] --wordlist=/usr/share/wordlists/rockyou.txt
- Result: Weak passwords were successfully cracked using brute force.

4. Vulnerability Validation:

   - Manual testing was used to validate hidden files and assess the risk of exposed sensitive data.

Assessment Toolset Selection

The following tools were selected to conduct the security assessment, each contributing to different stages of the process:

1. Nmap: Used to scan for open ports and running services.
2. John the Ripper: Employed to crack weak cryptographic hashes found in hidden files.
3. Manual Testing: Used to validate hidden files, weak hash algorithms, and misconfigurations.