Single System Assessment

HACKTHEBOX

Inlanefreight

Client:	Inlanefreight CEO
Date:	October 28, 2025
Report Version:	1.0
Primary Pentester:	Alexander Agricola
Reviewed By:	

Confidential – For Internal Use Only

Contents

1 Disclaimer

This report has been prepared exclusively for the use of **Inlanefreight** and is intended to provide an assessment of the security posture of the systems within the defined scope. The findings and recommendations reflect the state of the assessed systems at the time of testing.

There is no guarantee that all vulnerabilities have been identified or that no new vulnerabilities may exist. The report does not constitute a warranty of security.

This report may not be shared with third parties by **Inlanefreight**, the author **Alexander Agricola**, or any other individual or entity who may come across this report, without prior written consent from the other relevant parties. The pentesters involved accept no responsibility for any consequences resulting from the use of this report or the implementation of its recommendations.

2 Executive Summary

This [2] test was conducted on ACME Corp's network to identify vulnerabilities...

Key findings include[1]:

- High severity SQL injection vulnerability in the customer portal.
- Outdated software versions exposing critical security risks.
- Weak password policies discovered in the internal system.

Recommendations are provided to mitigate these risks.[2]

3 Scope

The objective of this penetration test is to evaluate the security posture of a single system within the client's environment. The assessment is limited to the following boundaries:

- **Target System:** 10.129.243.43
- **Services and Applications in Scope:** All services and applications hosted on the client system
- **Testing Type:** Black Box Test
- **Testing Limitations:** No social engineering, no denial-of-service testing, and no lateral movement to other hosts

The goal is to identify vulnerabilities that could potentially allow unauthorized access, data leakage, or compromise of system integrity.

All findings, recommendations, and risk ratings are based solely on the assessed system and its configurations at the time of testing. sco White Box Test

4 Methodology

The penetration test was conducted from a Kali Linux system connected to the client's network via OpenVPN. The assessment was performed in a straightforward manner, without attempting to evade detection mechanisms.

The primary goal of the engagement was to identify vulnerabilities that could allow administrative access to the target system. Standard security testing techniques were applied to safely verify potential weaknesses within the agreed scope and limitations.

5 Findings

5.1 High Level Summary

This report summarizes the key vulnerabilities identified during the penetration test. A total of 4 issues were discovered, categorized by severity below:

Critical	High	Medium	Low	Informational
1	0	0	0	0

Key observations:

- Sensitive Information Exposed in Public Repositories
- Passwords and critical information are stored in an insecure way.
- Vulnerable Software components were installed on the target machine.

A detailed breakdown of each finding is available in the following section.

5.2 Risk Classification and CVSS Explanation

Penetration testing uses a simplified risk categorization to focus remediation on the issues that matter most. The Common Vulnerability Scoring System (CVSS) is an industry-standard formula generating a risk score between 0.0 and 10.0, which provides a baseline severity rating for each vulnerability.

The table below summarizes the risk categories and their typical CVSS equivalency:

Risk Category	CVSS	Rationale
Critical	8.1 – 10.0	Severe vulnerabilities that are easy to exploit. Immediate remediation is recommended.
High	6.1 – 8.0	Significant risk that can be exploited. Address promptly after critical issues are resolved.
Medium	4.1 – 6.0	Important but potentially harder to exploit. Remedial work should be completed within approximately three months.
Low	2.1 – 4.0	Minor risk or difficult to exploit. Address over the long term as part of routine security cycles.
Informational	0.0 – 2.0	Observations that are not directly exploitable but may provide insight for future hardening.

Contextual Considerations

While CVSS provides a useful baseline, it does not always capture risks specific to the client's environment. For example, architectural issues such as a "flat network design" or exposed internal services may not have a high CVSS score, but in the context of the client's systems, they may represent critical risk. The report indicates the actual criticality based on the environment. This ensures that remediation priorities reflect both standardized scoring and client-specific impact.

5.3 Technical Findings

5.3.1 Sensitve Information Exposed in Public Repositories

5.3.2 Remote Code Execution via API Token and Automated CI/CD Pipeline

5.3.3 Weakly Encrypted Credentials in MGRemote Configuration File

Severity: High
CVSS 3.1 Score: CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N 7.8 (High)
Affected Domain: lock.htb

Description and Root Cause:
During the assessment, an MGRemote configuration file was identified on the client system containing a password in its default encrypted form. The encryption mechanism used is weak and reversible, meaning that anyone with access to the file can easily recover the plaintext credentials. This issue stems from reliance on the product's default configuration instead of implementing secure credential storage practices.

Security Impact:
Credential exposure occurs when attackers gain access to a configuration file, enabling them to retrieve plaintext credentials. If these credentials belong to privileged accounts, it can lead to privilege escalation, giving the attacker elevated access within the system. Additionally, the compromised credentials may be reused across different systems, facilitating lateral movement and allowing attackers to navigate through the environment undetected. Even after remediation efforts, there is a persistence risk, as stolen credentials can be exploited if they are not rotated properly.

Remediation:

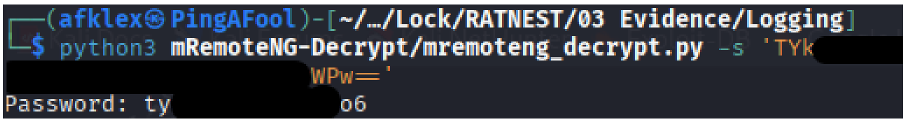
- Remove or Replace Default Encryption: Avoid relying on the default reversible encryption provided by MGRemote. At a minimum, configure MGRemote with a strong, custom password rather than leaving credentials in the default encrypted form. Where possible, integrate secure storage mechanisms to prevent reversible encryption from being the sole protection.
- Use Strong Encryption/Secrets Management: Store credentials using a secure secrets manager or operating system–native secure storage (e.g., DPAPI, LSA secrets, or Vault-based solutions).
- Credential Rotation: Immediately rotate any accounts stored in the affected configuration files to prevent potential misuse.
- Least Privilege: Ensure the credentials used in MGRemote have minimal permissions required for functionality.
- Access Control: Restrict file system permissions to limit who can read or modify configuration files.
- Monitoring: Audit usage of the exposed accounts and enable alerting for suspicious or abnormal authentication attempts.

External References:

- https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html
- https://attack.mitre.org/techniques/T1552/001/

Credentials Files found.

On the Document folder of the user ellen.freeman we found a configuration file.



```
┌──(afklex㉿PingAFool)-[~/…/Lock/RATNEST/03 Evidence/Logging]
└─$ python3 mRemoteNG-Decrypt/mremoteng_decrypt.py -s 'TYk[redacted]
[redacted]WPw=='
Password: ty[redacted]o6
```

Figure 1: Decrypted Credentials MGRemote

A Appendices

A.1 Changelog

Date	Version	Changes
2025-09-01	v1.0	Initial draft delivered to client
2025-09-05	v1.1	Incorporated client feedback on scope

A.2 List of Abbreviations

CVSS Common Vulnerability Scoring System. 4

A.3 Glossary of Terms

Black Box Test A penetration test where the tester has no prior knowledge of the system. The system is tested from an external perspective, simulating an attacker with no insider information.. 2

White Box Test A penetration test where the tester has full knowledge of the system, including source code and architecture.. 2

A.4 References

[1] Cve-2023-xxxxx. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-XXXXX, 2023. Critical RCE vulnerability.

[2] Gordon Lyon. Nmap network scanning. https://nmap.org/book/, 2023. Accessed: 2025-08-04.