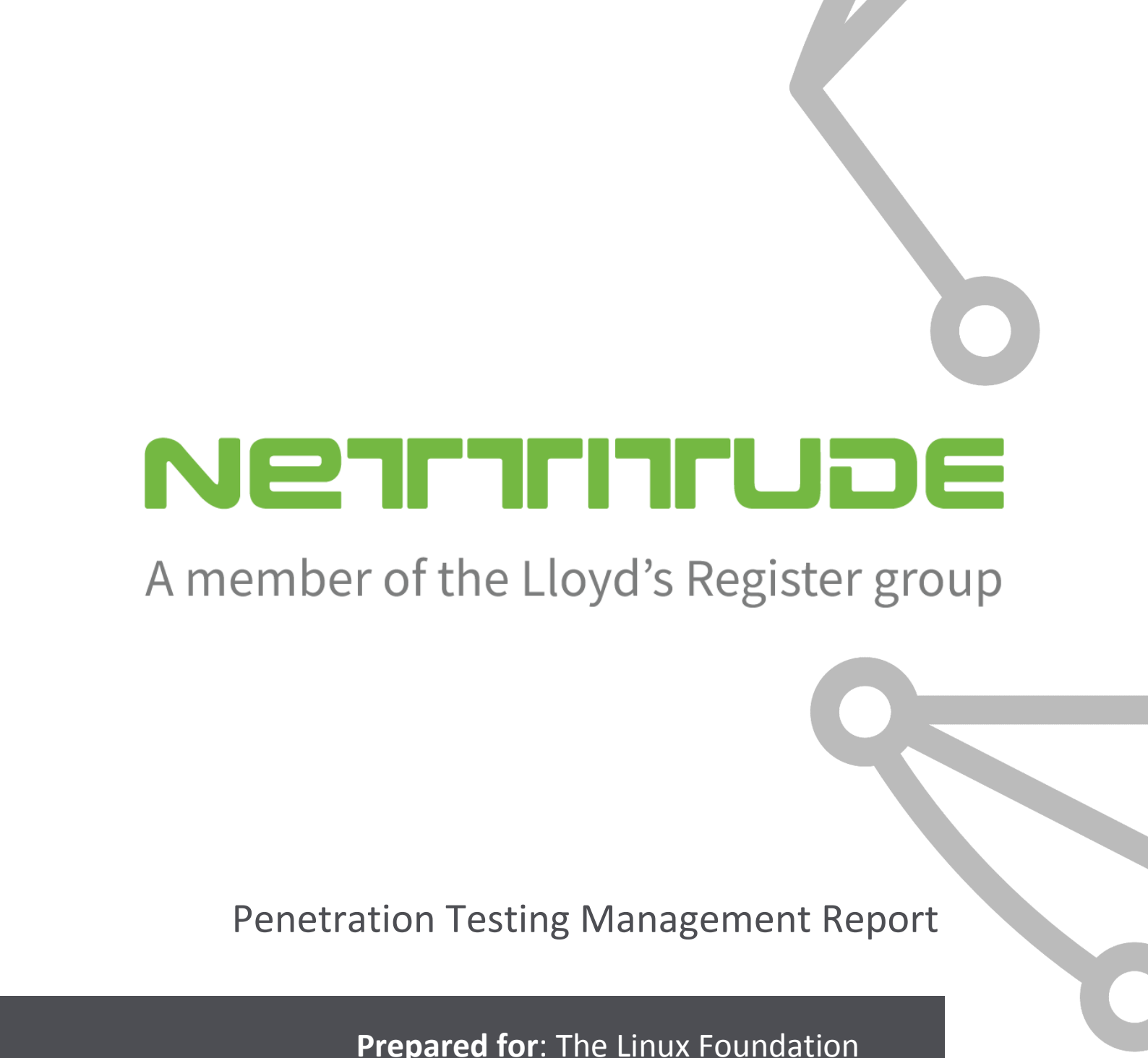


NETITUDE

A member of the Lloyd's Register group

# Penetration Testing Management Report

**Prepared for**: The Linux Foundation
**System**: Hyperledger Fabric v1.4 and v2.0
**Type**: Security Assessment

**Author**: Graham Shaw
**Date:** 8 August 2019
**Version**: 1.0

PCI Security Standards Council

CREST

CHECK
IT Health Check Service

# Report Contents

## High Level Assessment

The Linux Foundation engaged with Nettitude in June 2019 in order to assess the overall security posture of their environment.

Based on The Linux Foundation's risk profile, primary security concerns and the vulnerabilities identified at the point of the engagement, Nettitude have found the overall security posture to **require attention**.

### Nettitude were able to:

- Describe how an attacker might be able to gain access to the content of Private Data Collections despite only having access to a hash.
- Identify three further anomalies in the source code which detracted from what would otherwise have been a strong security posture.

### Overall Security Posture

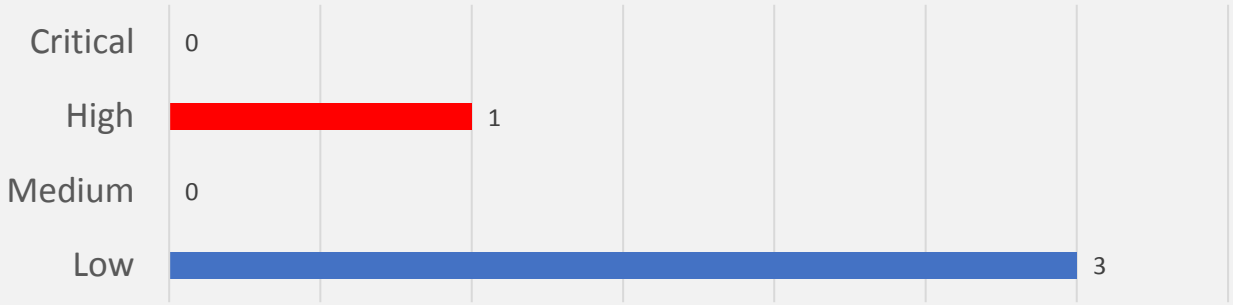
STRONG

MODERATE

**REQUIRES ATTENTION**

WEAK

### Vulnerabilities by Severity




### Limitations and Constraints

No limitations were encountered during the engagement.

## System Analysis

Hyperledger Fabric incorporates a feature called Private Data Collections whereby a transaction can be committed to the blockchain without making the details public. This is achieved by recording a hash of the data on the blockchain in place of the data itself.

The hashing algorithm used for this purpose is SHA256. At the time of writing there were no known attacks which can break the security guarantees offered by a standard implementation of SHA256. This includes the guarantee that, given the output of SHA256, it should not generally be possible to determine the input.

However, no hash function can fully prevent an attacker from attempting to guess the input and then determine whether that guess was correct. This is not a concern if the set of possible inputs is too large to have a significant chance of guessing correctly. However, the data that private data collections are intended to store is likely to be highly structured, making a brute-force search of plausible values a serious concern.

It is possible to defend against attacks of this type by 'salting' the data. That involves adding enough additional randomness to make the input to the hash function impracticable to guess. However, note that this differs from the type of salt used to protect password hashes in databases, in that it must be kept private.

The remaining three findings do not detract greatly from the security posture, and are of limited or no concern in the context of the current codebase. Remediation is, however, recommended in order to remove any risk of this state of affairs changing in the future. The issues are:

- Use of a cryptographic library function in a manner contrary to the relevant documentation.
- An input validation issue which would result in a panic as opposed to the expected error.
- Allowing a mode bit in chaincode tarfiles which appears to be a misreading of the relevant specification and which could in principle have undefined effect (but is most likely to have no effect).

In addition to these findings, a further seven issues were encountered during the investigation which do not appear to have any bearing on security, but nevertheless appeared worthwhile to report. These are listed in the technical report.

Otherwise, Hyperledger Fabric appeared to be well designed and well implemented. In particular, the use of gRPC and Protocol Buffers results in much of the outer attack surface being located in automatically generated code, produced using means which have already been very well exercised by third parties. The use of Go as an implementation language also reduces the opportunity for mishaps compared to some of the alternatives. The general-purpose nature of Chaincode continues to present some opportunities for mischief, however these have been partially mitigated since Nettitude's previous report of September 2017, and there is an unavoidable design trade-off between expressive power and risk.

## Next Steps

Nettitude recommends that The Linux Foundation perform the following post engagement activities in the order of priority indicated.

| | Activity | Description | Priority |
|---|---|---|---|
| 1 | Debrief from Nettitude | Nettitude will deliver a formal debrief to The Linux Foundation in order to ensure that the findings of this engagement have been fully comprehended and to help assist in the formulation of a remediation plan. | ++++ |
| 2 | Private Data Collections | Salt private data prior to hashing | +++ |
| 3 | Low severity issues | Address low-severity coding anomalies | ++ |

## Distribution List

| Nettitude | Name | Title |
|---|---|---|
| | Graham Shaw | Security Consultant |
| | Lilith Toro | Security Consultant |
| | Jose Lopes | Security Consultant |
| | Miles Corn | Account Manager |

| The Linux Foundation | Name | Title |
|---|---|---|
| | David Huseby | Security Maven |

## Revision History

| Version | Issue Date | Issued by | Comments |
|---|---|---|---|
| 0.1 | 25 July 2019 | Graham Shaw | Initial Draft |
| 0.2 | 28 July 2019 | Jose Lopes | Quality Assurance |
| 0.3 | 29 July 2019 | Miles Corn | Quality Assurance |
| 1.0 | 8 August 2019 | Graham Shaw | Final |