NETITUDE

INTELLIGENT CYBER SECURITY & RISK MANAGEMENT

Penetration Test Management Report

Prepared for The Linux Foundation

Authored by Graham Shaw

07 June 2018

Version 1.0

CREST

CHECK
IT Health Check Service

PCI Security Standards Council ™

NETTITUDE

INTELLIGENT CYBER SECURITY & RISK MANAGEMENT

Penetration Test Management Report

Nettitude provides a wealth of knowledge, expertise and experience in regards to Data Security. We provide comprehensive vulnerability assessment, penetration testing and application assessment services. Our team of dedicated security consultants deliver best in class testing capability as well as strong remediation advice and guidance.

NETTITUDE

REPORT CONTENTS

NETTITUDE

1 DISTRIBUTION LIST

Nettitude

Name	Title
Graham Shaw	Security Consultant
Miles Corn	Account Manager
Iain Wallace	Security Consultant

The Linux Foundation

Name	Title
David Huseby	Security Maven, Hyperledger

Revision History

Version	Issue Date	Issued By	Comments
0.1	25 May 2018	Graham Shaw	Initial Draft
0.2	01 June 2018	Iain Wallace	Quality Assurance
0.3	04 June 2018	Miles Corn	Quality Assurance
1.0	07 June 2018	Graham Shaw	Final

NETTITUDE

2 EXECUTIVE SUMMARY

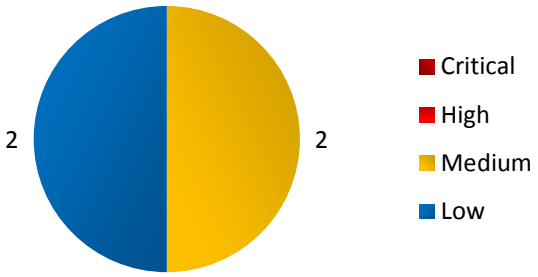
Background

The Linux Foundation engaged with Nettitude in May 2018 in order to assess the overall security posture of their Hyperledger Composer software package.

High Level Assessment

Based on the The Linux Foundation's risk profile, primary security concerns and the vulnerabilities identified at the point of the engagement, Nettitude have found Hyperledger Composer to:

REQUIRE MODERATE ATTENTION



Nettitude were able to…

- Access authentication credentials stored in world-readable files
- Access cards stored in composer-playground from other hosts
- Suggest ways in which the TLS configuration could be hardened
- Find unnecessary information disclosed in error messages

Primary Security Concerns

Nettitude worked with The Linux Foundation, prior to this engagement, to investigate and understand the primary security concerns associated with the systems in scope.

These concerns are not exhaustive, but rather represent a method of helping to gauge the severity of the overall risk presented by the systems in scope.

Concern	Description	Data Category
Web security	Web application and REST API have been secured against common weaknesses	Confidentiality Integrity
Credential storage	Credentials are stored securely on server	Confidentiality
Chaincode generation	Generator does not provide opportunities for code injection or similar vulnerabilities	Integrity

Table 1 – The Linux Foundation Primary Security Concerns

NETTITUDE

3 RISK AND ANALYSIS

Risk Profile

Nettitude present the following high level risk profile for The Linux Foundation in order to help contextualize the reasoning behind each findings severity and the overall system rating of 'requires moderate attention'. This is Nettitude's own assessment, based on their knowledge and understanding of The Linux Foundation, as an organization.

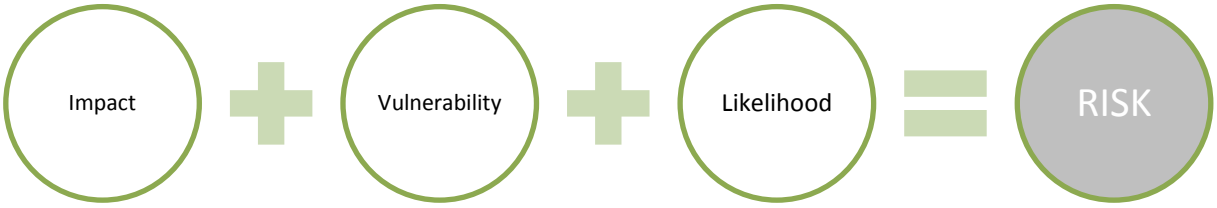
How to understand the values below?

All risks should be run through your own internal risk register and methodology. The aim below is to provide you with a benchmark and a stake in the ground. We have only had a glimpse of the data you hold, and have based the impact on your business on industry equivalents. It's very important that you re-assess and understand these values according to your business and its risk appetite.

How do we calculate risk?

In brief, assets have values which if compromised will have an impact on your business (reputation, ability to function, fines, etc). Weaknesses (or vulnerabilities) allow threats to access/disrupt these assets. The location of the vulnerability will determine the likelihood of the weakness to be exploited.

Risk is a factor of the vulnerability, the impact and the likelihood. Threats need to be considered, but these are outside the scope of this work (See ISO31000 for a detailed methodology).



Risk and Priority Key

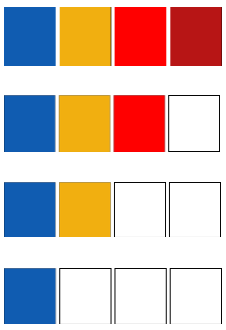
The following key shows how the level of risk and priority will be represented within this report.

Critical

High

Medium

Low



NETTITUDE

The Linux Foundation Risk Details

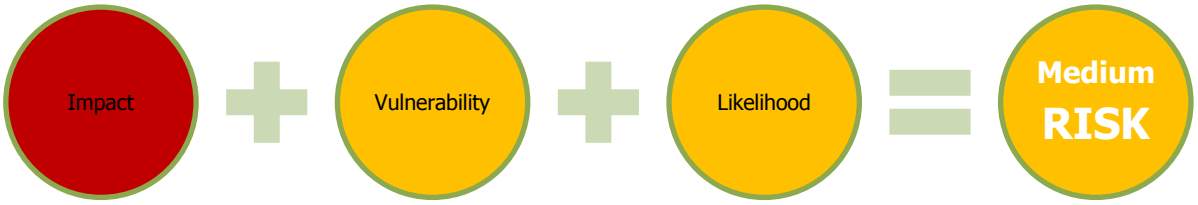
The table below shows the values calculated for this environment.

Risk Factor	Grade	Reasoning
Impact		Hyperledger Composer is a development environment for creating Hyperledger Fabric blockchain applications. In the worst case, a vulnerability could allow an attacker to modify the business logic of blockchains developed using Composer, in a manner which in turn compromises their security. Since these blockchain applications could be used for any purpose, including high-value financial transactions, the impact is potentially very high.
Vulnerability		When credentials are stored on the server, some are accessible to other users due to being world readable. Data stored in composer-playground, including credentials for accessing Fabric networks, can be accessed by other machines on the local network. The TLS configuration of the REST server is not as secure as it could be.
Likelihood		A prudent system administrator would avoid deploying Composer on a machine for which untrusted users had accounts. However world-readable files do pose some risk, because it is not uncommon for an attack against a machine to yield unprivileged access in the first instance. The TLS weaknesses found would not be easily exploitable under real-world conditions. Accessibility from other machines is easily exploitable, however is only a problem if the user is unaware of this behaviour or has failed to take appropriate countermeasures.

Table 2 – The Linux Foundation Risk Breakdown

Overall Risk Status

The overall risk for the environment under review for The Linux Foundation is shown below:



The Linux Foundation may perceive their risk profile to differ from what is presented in this section, in which case Nettitude would be happy to engage and discuss.

NETTITUDE

4 SYSTEM ANALYSIS

Nettitude assessed the security of Hyperledger Composer using a combination of penetration testing, fuzzing, and code review. It appears for the most part to be well-written and documented, and its use of third-party frameworks such as Angular, Loopback and Yeoman greatly reduces the risk of introducing some types of security vulnerability. Advantages of this approach include less new code written, coding of a more declarative, high-level nature, and specific countermeasures built into some of these libraries.

Some time was spent looking at whether the modelling or query languages could be exploited, particularly in light of its superficial similarity between the latter and SQL. However, while this could conceivably create code injection opportunities downstream of Composer (in the same way that SQL can), no obvious methods of exploitation were found within the scope of Composer itself.

The weaknesses that have been found are of a technically unsophisticated nature, and at most medium severity. The one that is perhaps most likely to result in a security breach in practice is the accessibility of the composer-playground interface to all hosts on a local area network (or in the worst case, the public Internet).

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp6       0      0 ::1:6010                :::*                    LISTEN      -
tcp6       0      0 :::8080                 :::*                    LISTEN      26463/node
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

Figure 1: Access allowed from other hosts to port 8080

Best practice is to default to access from the same machine only, but with the option to expand this to other machines if required. This has two beneficial effects:

- Access from other machines is blocked unless it is needed (in accordance with the principle of least privilege).
- If the server is configured to allow wider access, the user should at least be aware that this is the case as a result of having explicitly requested that configuration.

Nettitude was asked to check whether credentials are stored securely on the server, but it was found that they are not: some are world-readable, meaning that a malicious user with an unprivileged account on the same server would be able to access them. Modern administrative practices make it unlikely that an attacker would have a legitimate account on the same machine, however an exploit will often yield unprivileged access in the first instance, therefore there is significant risk from the files being world-readable.

```
malice@composer:/home/gdshaw/.composer/client-data/PeerAdmin@hlfv1$ ls -l
total 12
-rw-rw-r-- 1 gdshaw gdshaw  246 May 15 09:41 114aab0e76bf0c78308f89efc4b8c9423e31568da0c340ca187a9b17aa9a4457-priv
-rw-rw-r-- 1 gdshaw gdshaw  182 May 15 09:45 114aab0e76bf0c78308f89efc4b8c9423e31568da0c340ca187a9b17aa9a4457-pub
-rw-rw-r-- 1 gdshaw gdshaw 1024 May 15 09:41 PeerAdmin
malice@composer:/home/gdshaw/.composer/client-data/PeerAdmin@hlfv1$ cat 114aab0e76bf0c78308f89efc4b8c9423e31568da0c34
0ca187a9b17aa9a4457-priv
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQg00IwLLBKoi/9ikb6
ZOAV0S1XeNGWllvlFDeczRKQn2uhRANCAARrvCsQUNRpMUkzFaC7+zV4mClo+beg
4VkUyQR5y5Fle5UVH2GigChWnUoouTO2e2acA/DUuyLDHT0emeBMhoMC
-----END PRIVATE KEY-----
malice@composer:/home/gdshaw/.composer/client-data/PeerAdmin@hlfv1$
```

Figure 2: World-readable credentials

The TLS configuration is not as secure as it could be, since it allows connections using older protocol versions and cipher suites which fall significantly short of security best practice. However, none of these are egregiously poor, and the preferred configuration is a good one.

```
  Supported Server Cipher(s):
Preferred TLSv1.2  128 bits  ECDHE-RSA-AES128-GCM-SHA256   Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-GCM-SHA384   Curve P-256 DHE 256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA256       Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA384       Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA          Curve P-256 DHE 256
Accepted  TLSv1.2  256 bits  AES256-GCM-SHA384
Accepted  TLSv1.2  256 bits  AES256-SHA256
Accepted  TLSv1.2  256 bits  AES256-SHA
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA          Curve P-256 DHE 256
Accepted  TLSv1.2  128 bits  AES128-GCM-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA
Preferred TLSv1.1  256 bits  ECDHE-RSA-AES256-SHA          Curve P-256 DHE 256
Accepted  TLSv1.1  256 bits  AES256-SHA
Accepted  TLSv1.1  128 bits  ECDHE-RSA-AES128-SHA          Curve P-256 DHE 256
Accepted  TLSv1.1  128 bits  AES128-SHA
Preferred TLSv1.0  256 bits  ECDHE-RSA-AES256-SHA          Curve P-256 DHE 256
Accepted  TLSv1.0  256 bits  AES256-SHA
Accepted  TLSv1.0  128 bits  ECDHE-RSA-AES128-SHA          Curve P-256 DHE 256
Accepted  TLSv1.0  128 bits  AES128-SHA
```

Figure 3: Supported TLS cipher suites and protocol versions

The most secure course of action is to disable all but the most secure cipher suites and protocol versions, however there is a trade-off to be made between security and compatibility. Disablement of TLS v1.0 is likely justified given the type of application which Hyperledger Composer will be used for. For the other potential improvements, it will depend how far you wish to go in terms of prioritizing security over other considerations.

NETTITUDE

5 NEXT STEPS

Post Engagement Actions

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	Server binding	Arrange for the playground server at least to bind to the loopback address by default	
3	File permissions	Remove user and group access from files containing sensitive information.	
4	TLS configuration	Consider disablement of less secure TLS cipher suites and protocol versions, including in particular TLS version 1.0.	

Table 3 – Post Engagement Activities

Nettitude recommend that the contents of this report are fully understood prior to progressing onto the technical report, which provides further information on the individual vulnerabilities identified, including how to fix them.