NETITUDE

excellence as standard

Security Assessment Management Report

Prepared for: **The Linux Foundation**

System: **Fabric**

Type: Security Assessment

Author: Graham Shaw

Date: 19th September 2017

Version: 1.1

Report Contents

High Level Assessment

The Linux Foundation engaged with Nettitude in August 2017 in order to assess the overall security posture of their Fabric software product.

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

Nettitude were able to:

- Write malicious chaincode capable of performing an nmap scan
- Connect to a command-and-control server from within malicious chaincode
- Fabricate log entries

Overall Security Posture

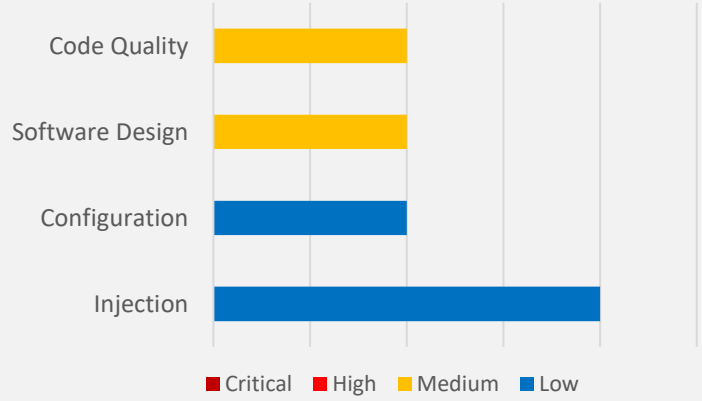
STRONG

MODERATE ATTENTION

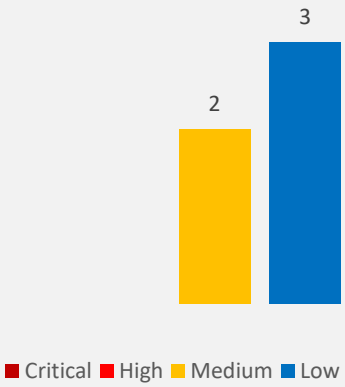
ATTENTION REQUIRED

IMIDIATE ATTENTION

Vulnerability category breakdown



Severity clasification



Limitations

Some limitations and constraints were encountered during the engagement. Please refer to the technical report for more details.

System Analysis

Fabric runs chaincode in a Docker container in order to securely isolate it from the rest of the system. It is good that it does this, and the measures in place would be effective in preventing many types of malicious activity, however they are insufficient to prevent malicious chaincode from being written.

The main concerns are that the chaincode has access to networking, can very easily download and install further software packages (including security tools), and can run for long periods of time. By bringing these capabilities together it would be possible to write a type of malware known as a Remote Access Trojan (RAT), the purpose of which is to act as a foothold onto a corporate network in order to allow other systems to be scanned and attached.

To demonstrate this, Nettitude wrote chaincode to perform a security scan of a machine attached to an internal network which is not directly reachable from the public Internet. The result was then exfiltrated to another machine acting as a command-and-control server:

```
Starting Nmap 7.01 ( https://nmap.org ) at 2017-08-24 07:54 UTC
Nmap scan report for giles.hellmouth.org.uk (192.168.0.129)
Host is up (0.0012s latency).
Not shown: 994 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
2049/tcp open  nfs

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

Figure 1: Port scan results collected and exfiltrated by malicious chaincode

Installation of the RAT would not in itself have any direct business impact, however it would act as an excellent base from which a threat actor could undertake a more comprehensive attack. For example, if any vulnerable network services were found during the port scan, the RAT could be used to exploit them and pivot to other systems.

Nettitude recognises that installation of malicious chaincode would be a non-trivial exercise for most threat actors given the level of access required, however there are some plausible scenarios:

- A threat actor could create a new ledger with associated malicious chaincode, and persuade others to participate.
- A threat actor could infiltrate an organisation responsible for developing and maintaining the chaincode for an existing ledger, then publish an update.

Fabric is also vulnerable to a method of attack known as log injection, which is made possible when unvalidated inputs are written verbatim to a log. This would similarly have no direct business impact,

however as part of a larger attack it might be used to fabricate log entries to mislead incident response efforts, or corrupt the log to prevent it from being processed by automated monitoring systems.

One function was found which was potentially vulnerable to a technique known as code injection, but not in a way that is exploitable as the program as a whole is currently written. This is a symptom of a larger concern, which is that the subcomponents of the system (functions and data structures) do not have detailed interface specifications which would allow a code audit to efficiently determine:

- Whether a function body correctly implements the required behaviour, and
- Whether calls to that function elsewhere in the program are using it correctly and appropriately.

Such interface specifications would certainly have improved the efficiency of the code review performed by Nettitude, and would have allowed greater depth and coverage to be achieved. It would also reduce the risk of misunderstandings when future changes are made by the code. The alternative is to engage in what amounts to reverse engineering of the existing code, which is both time-consuming and error-prone. The ideal is to provide a sufficiently-detailed specification such that the function could be:

- Safely re-written without reference to the locations from which it is called, and
- Safely used without reference to the current implementation.

Finally, Nettitude has some concerns about the use of remote imports in chaincode. Attempts to exploit this were unsuccessful due to the mitigations in place, however there is potential for chaincode to be written with a very large attack surface spanning multiple organisations. Whitelisting of third-party repositories is recommended.

Attempts were made to fuzz the HTTP- and gRPC-based network services, without success. This was done at both the presentation layer and the application layer, the latter making use of the supplied SDK to send well-formed messages which would exercise deeper parts of the attack surface.

Attempts were made to find weaknesses in the software by penetration testing. This covered issues such as TLS configuration, certificate pinning, authentication/session management, use of HTTP headers, path traversal and argument validation. No weaknesses were found.

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	Chaincode sandboxing	Improve sandboxing of chaincode to limit network access and persistence, and to execute from an unprivileged user account.	+++
3	Log injection	Sanitise untrusted strings before inserting them into log messages.	++
4	Comment headers	Provide comment headers which provide a detailed specification of the behaviour of each function.	+

Distribution List

Nettitude	Name	Title
	Graham Shaw	Senior Research Analyst
	Patrick Matthews	Security Consultant
	Jose Lopes	Security Consultant
	Kristopher Vasilik	Key Account Manager

The Linux Foundation	Name	Title
	David Huseby	Security Maven, Hyperledger

Revision History

Version	Issue Date	Issue By	Comments
0.1	1st September 2017	Graham Shaw	Initial Draft
0.2	4th September 2017	Jose Lopes	Quality Assurance
0.3	6th September 2017	Kristopher Vasilik	Quality Assurance
1.0	8th September 2017	Graham Shaw	Final
1.1	19th September 2017	Graham Shaw	Added pentest details