Forta Scan Node Assessment - Audit Report

April 7th, 2022

Provided to:

Forta

by:

dédalo

www.dedalo.io - info@dedalo.io

Table of Contents

OVERVIEW

Forta engaged Dédalo's team to conduct a white-box security audit of Forta Node and its source code. This process consisted of identifying possible threats and security risks in the components of Forta Node source code and its infrastructure. Dédalo performed source code reviews, manual and dynamic tests with multiple tools and techniques to identify security weaknesses. To properly test Forta Scan Node components, we performed tests locally and also set up a live Forta Scan Node instance running in a development environment configured to scan the Ethereum mainnet network.

Every finding is described in this document, including details that we consider necessary to understand the dangers of their possible impact, along with proper recommendations.

SCOPE

This assessment was done on the source code of the “Forta Scan Node” application provided by Forta through a private GitHub repository.

The following components were delivered to Dédalo and have been audited:

- GitHub repositories:
  - **`[REDACTED]`** (Language: Go)

- Live Forta Scan Node instance in development environment:
  - **`[REDACTED]`**

EXECUTIVE SUMMARY

For this audit, we analyzed all the components of Forta Node specified in the scope, and we couldn't identify security vulnerabilities that negatively affect Forta's business and users' data. This security assessment was conducted between March 14, 2022, and April 4, 2022, and the following days were used for findings triaging and report elaboration.

We performed a complete code review of the in-scope repository focusing on user-provided inputs, data and secrets management, Forta Agents communication mechanisms, Docker-based logic for running Agents, authorization, and authentication mechanisms. Along with code review, we performed multiple dynamic tests to find application security issues.

As there currently is almost no restriction on what an Agent can do, default configurations may allow Agents to be used to abuse the platform to perform malicious network activity while keeping this activity anonymous.

General recommendations

We analyzed Forta Node container-based logic for running Agent in order to verify the efficiency of isolation mechanisms for code running in Forta Agents. We found agents run in a proper isolated way. Regardless of whether containers are correctly isolated, we recommend restricting or reducing sharing of the Docker daemon socket file (*`docker.sock`* currently used by “*forta-json-rpc*” and “*forta-supervisor*” containers).

Given the lack of control of the actions Forta Agents can perform, we recommend implementing automated mechanisms to block Agents when performing abusive or malicious actions and better logging mechanisms to have broader visibility on how Agents are behaving.

WORK TEAM

The work team assigned to this project consisted of the following qualified roles:

- 1 Project Leader
- 2 Security Analysts

The Project Leader acted as the direct communication link between Dédalo and the client, resolving all doubts and providing the client with information and updates on the project, according to the times required by it. Simultaneously, he coordinated and provided support to the Security Analysts, which were responsible for carrying out the operative tasks that the security assessment process involved.

METHODOLOGY

Our security analysis methodology is based on the industry standards OWASP Testing Guide v4, OWASP Docker Security rules (for container-based systems), and OSSTMM (Open Source Security Testing Methodology Manual). For this project, we used a hybrid approach of penetration testing techniques against live instances of Forta Node for dynamic testing combined with manual source code review on the provided Forta Node repository.

COVERAGE

The main goal of this assessment was to evaluate if Forta Agents were able to execute operations bypassing the container-based sandboxed environment to the host OS. To properly evaluate this condition, and to cover other security issues in Forta Node, we performed several testing approaches for each component of Forta Node.

- Manual code review and dynamic tests on *forta-scanner* focusing on the following:
  - Agent running mechanisms.
  - Inter-agent communication.
  - Agent-to-host communication.
  - Agent alerts mechanism.
- Manual code review of *forta-json-rpc* and its JSON-RPC interface implementation.
- Manual code review and dynamic tests of *forta-supervisor* for services management.
- Manual code review of *forta-updater* for updating mechanisms.
- Manual code and configuration review of containers implementation and Docker infrastructure.

For dynamic tests, we set up a debugging version of forta-node using a “docker-compose” file and modified the source code (to fasten the warmup of the environment) to debug critical parts of the code in order to find flaws in the application and its interactions between the agent environment and its interactions with the other components

TECHNICAL REPORT

SECURITY ISSUES

We found the following issues classified by severity level:

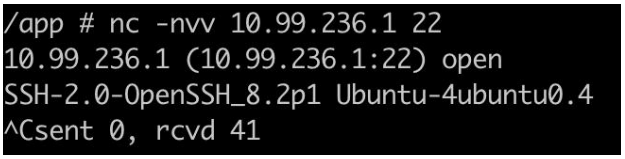
#01 - Unrestricted network access to Forta Node host from Agents

Severity: Low

Description

Agents running in containers have unrestricted network access to Forta Node host. This exposes local services which could not be accessed directly via Internet, but it is to Forta Agents. This could allow malicious users to scan or exploit local Forta Node host services.

Evidence



```
/app # nc -nvv 10.99.236.1 22
10.99.236.1 (10.99.236.1:22) open
SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
^Csent 0, rcvd 41
```

Agents have network access to Forta Node host SSH service.

Recommendations

Restrict access from Docker containers to Docker host. This can be done through firewall implementation (e.g. iptables rules).