Torus Node Penetration Test

Completed by: CertiK, LLC
July 11, 2020

Table of Contents

Confidentiality Statement

All information contained in this document is provided in confidence for the sole purpose of adjudication of the document and shall not be published or disclosed wholly or in part to any other party without CertiK's prior permission in writing and shall be held in safe custody. These obligations shall not apply to information that is published or becomes known legitimately from some source other than CertiK.

All transactions are subject to the appropriate CertiK Standard Terms and Conditions.

Certain information given in connection with this proposal is marked "In Commercial Confidence". That information is communicated in confidence, and disclosure of it to any person other than with CertiK's consent will be a breach of confidence actionable on the part of CertiK.

Disclaimer

This document is provided for information purposes only. CertiK accepts no responsibility for any errors or omissions that it may contain.

This document is provided without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall CertiK be liable for any claim, damages or other liability (either direct or indirect or consequential), whether in an action of contract, tort or otherwise, arising from, out of or in connection with this document or the contents thereof.

This document represents our budgetary price proposal for the solution further described in this herein and is provided for information and evaluation purposes only and is not currently a formal offer capable of acceptance.

1. Executive Summary

Torus engaged CertiK to perform a network penetration test and code review for their node and RPC APIs. The main objective of the engagement is to test the overall resiliency of the system to various real-world attacks against the application's controls and functions, and thereby be able to identify its weaknesses and provide recommendations to fix and improve its overall security posture.

CertiK started the test on July 6, 2020 and completed on July 10, 2020.

After a thorough review of the system, CertiK believes that the Torus node is currently at a **Informational** risk level. Given the severity of the vulnerabilities, it is unlikely that the system will be directly compromised.

We found that the backend API code and Node set up are well executed with regards to best practices and security design. Input and error handling while usage of context is well implemented across the codebase.

We suggest that Torus maintain this level of security on future development and leverage our team for a follow-up Penetration Test within 6 months, or immediately after, and major development changes.

1.1 Scope

At the start of the engagement, CertiK worked with Torus to identify the target and set the limits on the scope of the test. A White Box type of testing approach was done where CertiK performed the test with the source code available from the shared GitHub repository.

The following details the target scope of the test.

Target	Torus node
Github Repository	https://github.com/torusresearch/torus-node/
Target URL and IP	node-1.audit.dev.tor.us (18.181.196.195) node-2.audit.dev.tor.us (54.248.188.135) node-3.audit.dev.tor.us (54.178.94.9) node-4.audit.dev.tor.us (52.197.231.161) node-5.audit.dev.tor.us (13.231.208.91)
Environment	development

1.2 Threat Modeling

CertiK performed the threat modeling before starting the penetration test. The attack surface are entry points to the system, and where we would spend time testing. The threat modeling table lists assets in the system, the potential threat and impact if the asset is exploited by the attacker.

1.2.1 Attack surface

Open ports

Port	Service
22	SSH
53	DNS
80/443	RPC APIs
1080	Peer to peer communication
26656	Tendermint peer to peer communication

RPC API available Methods

- PingMethod
- ConnectionDetailsMethod
- ShareRequestMethod
- KeyAssignMethod
- CommitmentRequestMethod
- VerifierLookupRequestMethod
- KeyLookupRequestMethod
- UpdatePublicKeyMethod
- UpdateShareMethod
- UpdateCommitmentMethod

RPC server and Method handler implementation

1. https://github.com/torusresearch/torus-node/blob/011a062be65eb481ee65e12221d44de38d2b0d32/dkgnode/handlers.go
2. https://github.com/torusresearch/torus-node/blob/011a062be65eb481ee65e12221d44de38d2b0d32/dkgnode/jrpc_handlers.go
3. https://github.com/torusresearch/torus-node/blob/011a062be65eb481ee65e12221d44de38d2b0d32/dkgnode/server.go

1.2.2 Threat modeling table

Asset	Threat	Impact	Testing result
SSH access	Unauthorized access	Attacker access to the server	Not vulnerable, password login disabled.
Node(Networking) - Port 80/443 - RPC	Denial of Service	RPC APIs unavailable to users	Not Vulnerable.
Node(Networking) - 26656 tendermint p2p	Denial of Service	Lost tendermint p2p connection	Not vulnerable(with traffic of 50mbps)
Node(Networking) - 1080 p2p	Denial of Service	Lost p2p connection	Not vulnerable(with traffic of 50mbps)
RPC APIs	Requests with malformed and malicious input	Backend APIs malfunction	Not vulnerable

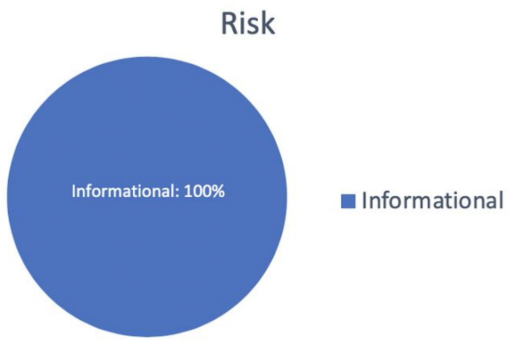
1.3 Limitations

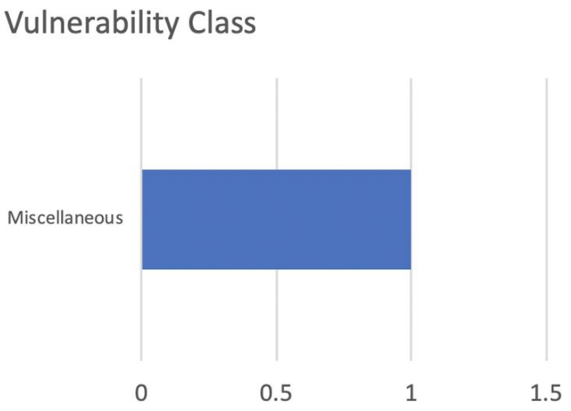
No major limitations were identified during the test.

Testing was performed during regular hours as well as off hours throughout the course of the test.

1.4 Summary of Results

Below summarize the vulnerabilities found on the target system:





ID	SEVERITY	TITLE	Vulnerability Class
2.1	Informational	Error code handling	Miscellaneous

2. Node Penetration Test

CertiK performed a full network penetration test on Torus’s node server. CertiK tested against different vulnerabilities including in the NIST, PTES (Penetration Testing Execution Standard), and OWASP Top Ten.

Listed below are the vulnerabilities that CertiK has found during testing.

Details of the vulnerabilities are provided as well as thorough recommendations on how the vulnerabilities can be fixed.

Findings are classified according to risk level.

2.1 Error code handling

Severity: Informational

Description

The code contains repetitive numbers (`32602, 16...`) usage that is making the code less readable and more prone to error in a future update.

Location

1. https://github.com/torusresearch/torus-node/blob/d9fe26e7d8714fe199a893cd7667c0f46a775620/dkgnode/jrpc_handlers.go#L59
2. https://github.com/torusresearch/torus-node/blob/d9fe26e7d8714fe199a893cd7667c0f46a775620/dkgnode/jrpc_handlers.go#L88

Impact

N/A.

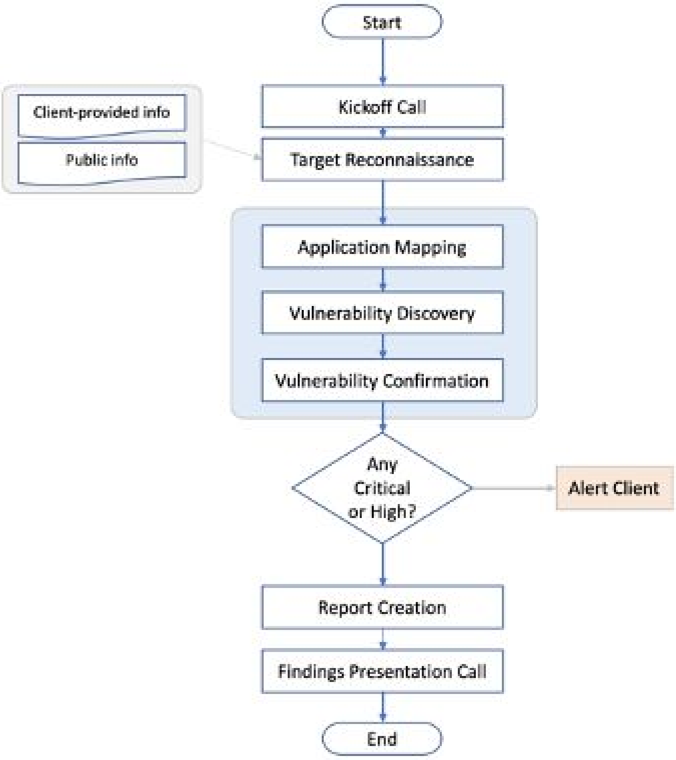
Recommendation

We recommend converting some repetitive numbers to be represented as const to enhance the already very good readability of the codebase and make the code potentially more easy to update.

3. Appendix – Methodology

CertiK uses a comprehensive penetration testing methodology which adheres to industry best practices and standards in security assessments including from **NIST**, **PTES** (Penetration Testing Execution Standard).

Below is a flowchart of our assessment process:



3.1. Coverage and Prioritization

As many components as possible will be tested manually. Priority is generally based on three factors: critical security controls, sensitive data, and the likelihood of vulnerability.

Critical security controls will always receive the top priority in the test. If a vulnerability is discovered in the critical security control, the entire target is likely to be compromised, resulting in a critical-risk to the business. For most targets, critical controls will include the login page, but it could also include major workflows such as the checkout function in an online store.

The Second priority is given to target components that handle sensitive data. This is dependent on business priorities, but common examples include payment card data, financial data, or authentication credentials.

Final priority includes areas of the targets that are most likely to be vulnerable. This is based on CertiK' experience with similar targets developed using the same technology or with other targets that fit the same business role.

3.2. Reconnaissance

CertiK gathers information about the target from various sources depending on the type of test being performed. CertiK obtains whatever information that is possible and appropriate from the client during scoping and supplements it with relevant information that can be gathered from public sources. This helps provide a better overall picture and understanding of the target.

3.3. Vulnerability Discovery

Using the information that is gathered, CertiK comes up with various attack vectors to test against the target. CertiK uses a combination of automated tools and manual techniques to identify vulnerabilities and weaknesses. Industry-recognized testing tools will be used, including Burp Suite, Nikto, Metasploit, and Kali. Furthermore, any controls in place that would inhibit the successful exploitation of a particular system will be noted.

3.4. Vulnerability Confirmation

After discovering vulnerabilities in the target, CertiK validates the vulnerabilities and assesses its overall impact. To validate, CertiK performs a Proof-of-Concept of an attack on the vulnerability, simulating real world scenarios to prove the risk and overall impact of the vulnerability.

Through CertiK' knowledge and experience on attacks and exploitation techniques, CertiK is able to process all weaknesses and examine how they can be combined to compromise the target. CertiK may use different attack chains, leveraging different weaknesses to escalate and gain a more significant compromise.

To minimize any potential negative impact, vulnerability exploitation was only attempted when it would not adversely affect production environment and systems, and then only to confirm the presence of a specific vulnerability. Any attack with the potential to cause system downtime or seriously impact business continuity was not performed. Vulnerabilities were never exploited to delete or modify data; only read-level access was attempted. If it appeared possible to modify data, this was noted in the list of vulnerabilities below.

3.5. Immediate escalation of High or Critical Findings

If critical or high findings are found whereby targets elements are compromised, client's key security contacts will be notified immediately.

3.6. Risk Assessment

The following risk levels categorize the risk level of issues presented in the report:

Risk Level	CVS Score	Impact	Exploitability
Critical	9.0-10.0	Root-level or full-system compromise, large-scale data breach	Trivial and straightforward
High	7.0-8.9	Elevated privilege access, significant data loss or downtime	Easy, vulnerability details or exploit code are publicly available, but may need additional attack vectors (e.g., social engineering)
Medium	4.0-6.9	Limited access but can still cause loss of tangible assets, which may violate, harm, or impede the org's mission, reputation, or interests.	Difficult, requires a skilled attacker, needs additional attack vectors, attacker must reside on the same network, requires user privileges
Low	0.1-3.9	Very little impact on an org's business	Extremely difficult, requires local or physical system access

Informational	0.0	Discloses information that may be of interest to an attacker.	Not exploitable but rather is a weakness that may be useful to an attacker should a higher risk issue be found that allows for a system exploit

4. Why CertiK

CertiK is a blockchain cybersecurity company with a global headquarter in New York City and presence in Beijing, Seattle, Seoul and Tokyo, pioneering the use of cutting-edge technologies, including static/dynamic analysis, Formal Verification, and Penetration Testing. CertiK has received grants from IBM, the Qtum Foundation, and the Ethereum Foundation to support its research of improving security across the blockchain industry. CertiK also contributes to the technical communities and ecosystems by providing guidance, research, and advisory about blockchain and smart contract best practices.

Our Penetration Testing service envisions to empower individuals and businesses to thrive in the new digital security age, especially in the blockchain space.