Sayfer Blockchain Audit

XDC2.0

A Few Of Our Clients

Who Are We?

We are a tier-one Blockchain cybersecurity company with extensive experience auditing blockchains such as BNB Chain, Polkadot, Tezos, Starkware, and Aleo.

We believe that the team of auditors is the most important aspect of a quality audit. That's why we have assembled a team with a unique skill set that we believe is ideally suited for your project.

Most of our team members are crypto-analysts from the Israeli top-secret cyber army and national security units, specializing in blockchain cryptography. In addition, we have highly experienced auditors with expertise in web2 and web3 domains. This diversity of skills makes our team one of the most comprehensive in the industry.

Scope of Work

We believe auditing requires a customized approach rather than a one-size-fits-all solution. Below is a list of tests we will perform, but to provide a detailed audit scope, we must ask more questions about your project and fully understand your specific needs.

We have reviewed the whitepaper and code, but as experts in the field, it is our responsibility to gain a comprehensive understanding of your systems and ensure that all the documents you provide are all there is to audit. We attempted to contact you on Discord. Please contact our CEO, Nir, on Telegram @nir_1337 to discuss this further.

The following are some of the tests we will perform:

Master Node Election

When conducting an audit of the node election process, it is crucial to examine various angles. First, it is important to review the electing code to ensure that it is secure and cannot be forged from a cryptographic standpoint. Additionally, it is essential to strengthen the criteria for nodes to be eligible, by including multiple security features.

It is important to note that the security of the blockchain is dependent on the security of the entire infrastructure and not just individual nodes. Therefore, it is necessary to monitor and track important node infrastructure data to maintain diversity between nodes. This means avoiding situations where all nodes are running on the same cloud infrastructure and using the same 3rd party libraries. Such a situation can be dangerous because one CVE can compromise all nodes.

Finally, to ensure the safety of private keys, we will create a more robust and diverse security key management scheme for nodes.

The HotStuff Protocol

Our priority is to thoroughly test the HotStuff Protocol, which has already proven its reliability. To achieve this, we will begin by using proprietary tools to analyze all changes made by the XDC Protocol Team. If any changes are found, we will specifically examine those code segments to assess their impact on the overall security of the protocol.

Additionally, we are aware of known issues with HotStuff products such as "Reliable Leader" and "Leader Bottleneck," and will investigate how these issues relate to the implementation of the XDC Protocol, including accountability, reliability, and forensics.

Proposing and Voting

Our analysis indicates that there is a high risk of remote code execution in this process. This is due to the complex messaging between nodes and the potential for Go data races, which can lead to one node infecting all others and compromising the protocol. In addition, there is also the risk of business logic attacks on the mechanism, which could impact the safety and liveness of the nodes.

Performance Guarantees

In our testing, we concentrate on examining typical network assaults. One type that we come across frequently is denial of service attacks. The recent incidents with Solana demonstrate how much financial harm can result from such an attack, even without any fraudulent transactions taking place.

We strongly support a proactive approach at this point and suggest that nodes should be equipped with active modules to identify and prevent these network attacks (such as an internal firewall). We can provide assistance with developing feature specifications, but we require additional details.

Reward Mechanism

We will ensure that the reward mechanism's code is clean and secure. Additionally, we can assist in developing a reward monitoring dashboard to prevent any future compromise of rewards. However, we must consider financial feasibility and upgradability before proceeding.

Forensic Monitoring

We are thrilled to witness the progress made at the network level! However, we have identified some potential risks. One of the most common ones is the possibility of nodes manipulating the database. Additionally, there are RPC attack vectors that can be exploited for the aforementioned attacks, or even for remote code execution.

Penalty Mechanism

To ensure the safety and security of the mechanism, we must gain a deeper understanding of all its off-chain components. Our team will conduct a thorough analysis to detect any potential security flaws in the design. It's crucial that we audit the dashboard site to prevent any potential breaches since even a single stored XSS vulnerability could lead to a significant security breach, just like what happened in the BadgerDAO hack.

Audit Methodology

When conducting network audits, it is recommended to adhere to OWASP WSTG 4.2 and SCSVS standards, while also taking into account the previous experience of the auditors. It is important to note that there is no universally applicable methodology for such systems.

Here is a breakdown of our methodology:

Knowledge Transfer

To begin, we'll engage in discussions with the development team and onboard our auditing team. This is a critical step to gaining a deeper understanding of the system's architecture. We'll then prioritize the most important components to begin our work.

Static Analysis

Code Review

We perform a detailed line-by-line examination of the codebase to identify code quality, coding standards, and potential vulnerabilities.

Dependency Analysis

We assess all third-party dependencies and libraries to make sure they do not pose any security risks.

Dynamic Analysis

Functional Testing

We will ensure that all features of XDC2.0 are functioning properly by comparing the expected business logic, documentation, and previous conversions to the test results. This includes evaluating the network's ACL, authentication and authorization, encryption, and peer-to-peer communication.

Algorithm Testing

We will examine the BTF algorithm and confirm that XDPoS1.0 to XDPoS2.0 functions as outlined in the white paper. Additionally, we will address common attacks such as Censorship, replay, and Sybil.

Stress Testing

We simulate extreme conditions to test the network's resilience. This includes surges in transaction volume, simultaneous requests from multiple nodes, and P2P tests such as message flooding and malicious node simulation.

Retesting

Addressing a vulnerability is often a complex process, and occasionally, a new vulnerability may emerge even after the initial fix has been implemented. Therefore, we will conduct another round of testing to ensure that all issues have been thoroughly resolved. Our aim is to provide full support to the development team throughout this process.

Final Reporting

Once all testing phases are finished, we will provide a detailed and actionable audit report.

Hashing Review

Finally, we will ensure that the final binaries deployed match the approved ones from our audit as an end-to-end security measure.

Pricing and Fee Schedule

Our fee for the described audit services will be **$85,000**, with 50% due before commencement and 50% upon report delivery.

This fixed price covers all auditor time and expenses.

Project Timeline

The project will start on an agreed-upon date after the price proposal is finalized and signed. Subsequently, the project will follow the following phases:

1. One week before the agreed-upon starting date of the project, we will schedule a kickoff meeting. The meeting will provide us with knowledge about the platform with a live demo. The meeting will also include a short business risk analysis.
2. We will open a direct message channel through Slack or Telegram to keep you updated in real time about critical information.
3. We expect to work on the project for approximately **8 weeks** starting from the agreed-upon date.
4. After completing our assessment, we will provide the client with a comprehensive report detailing any vulnerabilities that were found. The report will also include explanations on how to eliminate or minimize these vulnerabilities.
5. Once we reach this stage, we move on to the revision phase. During this phase, the developers of the client will address any security issues that have been identified, and we will verify that they have been resolved properly. Typically, this phase will require one or two rounds of revisions, and the length of time needed for each revision will vary based on the client's needs.
6. Upon completing the revision phase, your system will achieve a high level of security and earn the **Sayfer Badge**, which is available in various design options to complement your design materials.

The Sayfer team is always available to answer any questions you may have. You can contact us at:

E- mail: info@sayfer.io

Telegram: https://t.me/SayferSecurity

or schedule a meeting: Calendly

Thank you for your consideration and we look forward to hearing from you.

Sincerely,

The Sayfer Team