CyStack

DATA SHEET

Blockchain Protocol Audit

Auditing for the integrity of your blockchain protocol

Security Challenges

Blockchain technology is designed to be secure, but there are still several challenges to ensuring the security of a blockchain. Some of these challenges include:

- **51% attack**: In a 51% attack, a group of attackers gains control of more than 50% of the computational power of a blockchain network, allowing them to manipulate the network and potentially reverse or block legitimate transactions.
- **Sybil attack**: A Sybil attack is when a malicious actor creates multiple identities or nodes in a network in order to gain control of a significant portion of the network's computational power.
- **Double-spending**: In a double-spending attack, an attacker is able to spend the same digital currency or token more than once by creating a copy of the digital asset.
- **Privacy and data leakage**: A privacy and data leakage attack is when an attacker accesses sensitive data stored on the blockchain, such as personal information or financial data.
- **Side-Channel Attack**: A side-channel attack is an attack that is exploiting some side information that is not part of the main communication channel, such as power consumption or electromagnetic radiation.

These are just a few examples of the security challenges that blockchain technology faces. The field is still evolving, and security researchers and developers are constantly working to identify and mitigate new threats.

Customer Benefits

- Identify and address vulnerabilities and potential security risks before the launching
- Ensure that the protocol can interact securely with other systems
- Build trust with stakeholders and customers
- Avoid costly security breaches and downtime in the future
- Improve the performance and overall efficiency of the network

Overview

A blockchain protocol audit is an assessment including reviewing and analyzing its design, implementation, and security, to identify any vulnerabilities or weaknesses that may be exploited by attackers. The audit is typically performed by auditors with expertise in blockchain technology. The audit comprises code and documentation evaluations, testing, and analysis to pinpoint security vulnerabilities, as well as interviews with developers and stakeholders to make sure that the blockchain protocol is properly implemented according to documentation.

A blockchain protocol audit helps developers to:

- Assess the overall security and reliability of the blockchain protocol to identify any vulnerabilities or weaknesses that could be exploited by attackers.
- Evaluate the design and implementation of the blockchain protocol to ensure that it meets established standards and best practices.
- Identify potential threats and attack vectors that could be used to compromise the security of the blockchain protocol.
- Receive recommendations for improving the security and reliability of the blockchain protocol.

Overall, a blockchain protocol audit is an important step in ensuring the security and reliability of a blockchain protocol, which is essential for maintaining the trust and confidence of users and stakeholders.

How CyStack Helps

The CyStack Audit Team is a group of highly skilled security testers who use a goal-oriented approach to testing, refined through years of experience and extensive testing. Our team members have a unique blend of app development and security testing expertise, enabling them to conduct comprehensive security evaluations that uncover potential risks for organizations. Members of this team are also regular speakers at world-known cyber security conferences and also talented bug hunters who discovered many critical vulnerabilities in the products and are acknowledged in the Hall of Fame of global tech giants such as IBM, HP, Microsoft, Alibaba, Sea Group, etc.

We understand the intricacies of blockchain protocols and the potential risks associated with them. We have a proven methodology that includes code review, automated testing, penetration testing, compliance testing, performance testing, and interoperability testing. This approach helps us to identify and address any vulnerabilities or issues in the blockchain protocol, ensuring that it is secure, efficient, and reliable. Performance testing includes scalability, gas consumption, and usability evaluations. This can help businesses to identify and address any issues that may affect the performance of their blockchain protocol.

Key Features

- Comprehensive review
- Thorough code review
- Automated testing powered by SafeChain.org
- Penetration testing
- Governance testing
- Performance testing
- Interoperability testing
- Bug bounty program powered by WhiteHub.net

Methodology

Security auditing a blockchain protocol involves a systematic process to identify vulnerabilities and potential threats. CyStack strictly follows the below steps:

1. **Review the documentation**: Review the documentation of the blockchain protocol to understand its design and functionality. This includes the whitepaper, the codebase, and any other available documentation.
2. **Understand the consensus mechanism**: Understand the consensus mechanism used by the blockchain protocol. This will help to identify any potential weaknesses in the mechanism that could be exploited.
3. **Perform code review**: Perform a thorough code review of the blockchain protocol, looking for potential vulnerabilities, such as buffer overflow, SQL injection, and other common software vulnerabilities.
4. **Test the code**: Use automated testing tools to test the code for potential vulnerabilities. This includes unit testing, integration testing, and regression testing.
5. **Perform penetration testing**: Perform penetration testing to simulate a real-world attack on the blockchain protocol. This includes trying to exploit vulnerabilities identified during the code review and testing.
6. **Evaluate compliance**: Evaluate the blockchain protocol for compliance with relevant regulations and industry standards.
7. **Interoperability testing**: Evaluate the blockchain protocol's compatibility with other blockchain networks and systems, ensuring that the protocol can interact with other systems in the ecosystem.
8. **Performance testing**: Evaluate the blockchain protocol's performance, including its scalability, gas consumption, and usability.
9. **Create a report**: Create a detailed report of the audit findings, including any vulnerabilities or issues identified, and recommendations for addressing them.
10. **Provide remediation support**: Provide assistance with addressing any issues or vulnerabilities identified during the audit, including guidance on how to remediate the issues.

Vulnerability List

During the audit, vulnerabilities in the following table will be tested:

Layer	Component	Vulnerability	Severity	Description
Network Layer	**P2P**	**Sybil Attack**	High	An attacker creates multiple identities and then uses them to control the network or disrupt its normal operation. For example, an attacker may create multiple identities in order to control a majority of the voting power in a decentralized network, allowing them to control the network's decision-making process.
		Eclipse Attack	High	An attacker isolates a specific node or group of nodes from the rest of the network. This is accomplished by redirecting the target network participant's inbound and outbound connections from a specific node or group of nodes to the attacker's nodes.
		Eavesdropping Attack	Low	An attacker intercepts and monitors the communication between two or more parties without their knowledge or consent. This can allow the attacker to gain access to sensitive information such as login credentials, financial data, or personal information.
		Denial of Service Attack	Medium	A Denial of Service (DoS) attack can be used to disrupt the normal operation of the network by overwhelming the resources of one or more nodes. For example, an attacker may use a botnet to flood a node with a large number of requests, causing it to become unresponsive. This can lead to a reduction in the overall performance and availability of the network, and can potentially cause the network to become unavailable to legitimate users.
		BGP Hijack Attack	Low	An attacker intercepts and redirects Internet traffic by manipulating the routing information exchanged between routers. This is accomplished by falsely announcing ownership of IP prefixes that the attackers do not actually own, control, or route to, causing the traffic to be redirected from its intended destination.
		Alien Attack	Low	A type of attack that targets decentralized networks by creating a new chain with the same communication protocol as an existing chain or fork of the existing chain. The attacker then uses this new chain to attack the existing chain, causing them to pollute each other, degrading node communication performance, and eventually causing node blockage.
		Timejacking	High	A type of attack on a blockchain network that attempts to manipulate the time in the system. The attacker does this by broadcasting incorrect time information to the network, causing the nodes in the network to have a different understanding of the current time. This can lead to confusion and disrupt the normal operation of the network, as well as potentially cause the network to become unavailable to legitimate users.

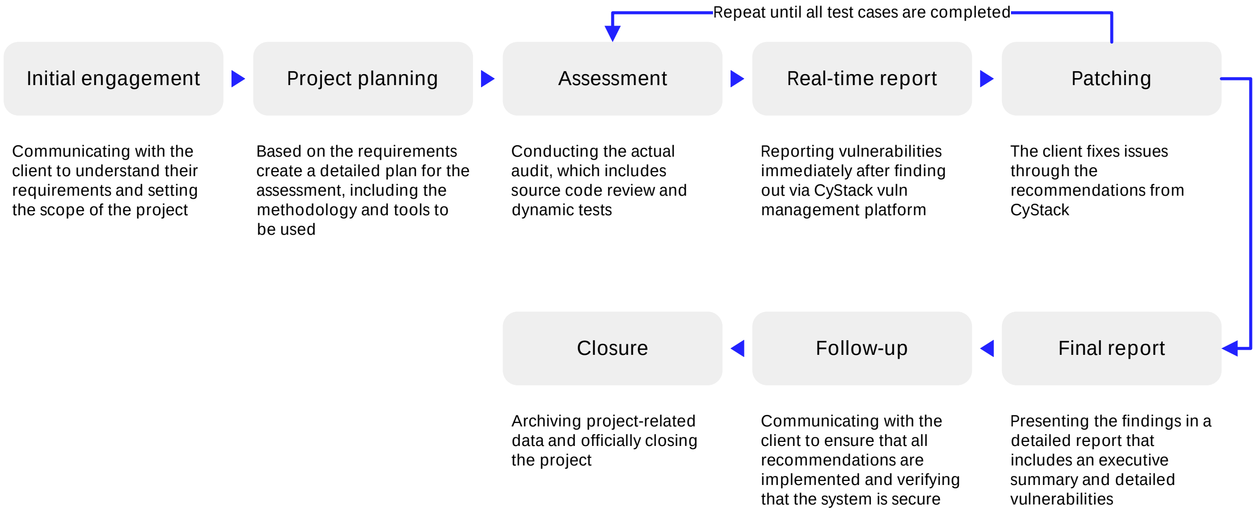
Network Layer	**RPC**	**Eavesdropping Attack**	Low	An attacker intercepts and monitors the communication between two or more parties without their knowledge or consent. This can allow the attacker to gain access to sensitive information such as login credentials, financial data, or personal information.
		Denial of Service Attack	Medium	A DoS attack can be used to disrupt the normal operation of the network by overwhelming the resources of one or more nodes. For example, an attacker may use a botnet to flood a node with a large number of requests, causing it to become unresponsive. This can lead to a reduction in the overall performance and availability of the network, and can potentially cause the network to become unavailable to legitimate users.
		The Ethereum Black Valentine's Day Vulnerability	Low	This vulnerability is exploited by leveraging an authentication flaw of Ethereum node's Geth/Parity RPC API to maliciously steal tokens via eth_sendTransaction.
		HTTP Input Attack	Low	Cross-Site Scripting (XSS) / Template injection / Third-party component vulnerability / HTTP parameter pollution / SQL injection / XXE entity injection / Deserialization vulnerability / SSRF vulnerability / Code injection / Local file contains / Remote file contains / Command execution injection / Buffer overflow / Formatted string
		Cross-Domain Phishing Attack	Low	This attack can be used to steal sensitive information such as private keys, seed phrases, or login credentials by tricking the user into visiting a fake website that looks like a legitimate RPC endpoint.
Ledger Layer	**Consensus**	**Long Range Attack**	High	Long Range Attacks are only relevant for Proof of Stake (PoS) protocols. An attacker creates a new chain that is a fork of an existing chain but with a longer chain history that is partially or completely different than the main chain. The attacker then uses this new chain to overtake the main chain due to the longest chain rule, causing the network to become unavailable to legitimate users, as well as allowing them to manipulate the network in their favor.
		Bribery Attack	High	An attacker attempts to bribe a miner or validator to mine or validate malicious transactions. This can be done by offering a miner or validator a large payment in exchange for mining or validating a specific transaction, or by threatening to withdraw support from a miner or validator if they do not comply.
		Race Attack	High	An attacker may attempt to accomplish this by creating two conflicting transactions, one that sends the digital assets to the attacker's address and the other that sends the digital assets to a legitimate recipient. The attacker then attempts to broadcast the transaction that sends the assets to their address before the other transaction is confirmed by the network.

Ledger Layer	**Consensus**	**Liveness Denial**	High	Liveness Denial is a form of Denial of Service attack in PoS protocols. In this attack, some or all of the validators decide to take action and purposefully block transactions by stopping publishing blocks. By avoiding to perform their validator duties, the blockchain will come to a halt as new blocks would not be able to be validated and published in the blockchain.
		Censorship	High	An attempt by an attacker to prevent certain transactions from being processed or recorded on the blockchain. This can be done by blocking access to certain nodes, manipulating the consensus mechanism, or interfering with the network's communication channels. The goal of this type of attack is to prevent specific transactions from being recorded on the blockchain, and as a result, disrupt the normal operation of the network.
		Finney Attack	High	Finney Attack is a variation of a Double Spend Attack that is possible when one transaction is premined into a block and an identical transaction is created before that premined block is released to the network, thereby invalidating the second identical transaction. The attacker can leverage this type of attack to send coins to themselves, but instead of sending coins to a definite merchant.
		Vector76 Attack	High	Vector76 is a combination of Race Attack and Finney Attack. This type of attack can allow a malicious miner to double spend a transaction by targeting a small subset of nodes, causing harm to the network.
		Alternative Historical Attack	High	An attacker creates an alternative version of the blockchain's history and attempts to convince the network to switch to that version. This can be done by creating a fork of the blockchain and then using the computational power to mine blocks on that fork, potentially causing a network split.
		51% Attack	High	An attacker controls more than 50% of the network's mining power (hash rate), allowing them to manipulate the network in their favour.
		Grinding Attack	High	Grinding Attack, also known as precomputation attack, is an implementation-specific issue and affects PoS systems. By exploiting the lack of randomness in the slot leader election process, a slot leader is capable of manipulating the frequency of them being elected in subsequent blocks.
		Coin Age Accumulation Attack	High	Coin age, which is a central concept in PoS algorithms, refers to the amount of time coins have been inactive. An attacker accumulates a large amount of coin age on a set of coins by holding them unspent for a long period of time, resulting in the possibility to claim the accumulated award within minutes, or even take over the network.

Ledger Layer	**Consensus**	**Selfing Mining**	High	An attacker controls more than 50% of the network's hash rate, allowing them to mine blocks faster than the rest of the network. The attacker then uses this control to perform selfish mining, which is the process of mining blocks on their own blockchain network, without broadcasting them to the rest of the network until the chain reaches their desired length.
		Block Double Production	High	An attacker creates and publishes two different blocks at the same height of the blockchain. This can happen when an attacker controls more than 50% of the network's hash rate, allowing them to mine blocks faster than the rest of the network.
	Encryption	**Cryptographic Attack**	High	Common attack methods: Analytic Attack / Implementation Attack / Statistical Attack / Brute Force / Frequency Analysis and the Ciphertext Only Attack / Known Plaintext / Chosen Ciphertext / Chosen Plaintext / Meet in the Middle / Man in the Middle / Birthday attack / Replay attack / Collision attack
		Private Key Prediction	High	An attacker uses various methods to predict the private key that corresponds to a public address. This allows the attacker to access the funds associated with the public address without the owner's knowledge or consent.
		Length Extension Attack	Low	A type of attack on a cryptographic hash function that allows an attacker to extend the length of a message and find the corresponding hash, without knowing the original message or its hash. This can be done by manipulating the internal state of the hash function, which is typically based on a Merkle-Damgard construction.
		Hash Collision attack	High	A type of attack on a cryptographic hash function that allows an attacker to find two different messages that have the same hash value, also known as a "collision". This can be done by manipulating the internal state of the hash function, or by using mathematical algorithms to find two messages that have the same hash value.
	Transaction	**Double Spend Attack**	High	An attacker attempts to spend the same digital assets more than once. This can be done by creating multiple copies of the same digital assets, or by creating conflicting transactions that spend the same assets in different ways.
		Transaction Malleability Attack	High	A type of attack on a blockchain network that allows an attacker to change the transaction ID of a transaction, without changing the content of the transaction. This can be done by manipulating the digital signature of the transaction, or by modifying the transaction data in a specific way.

Ledger Layer	**Transaction**	**Time-Locked Transaction Attack**	Low	An attacker manipulates the time-lock feature of a transaction to prevent it from being executed or to execute it earlier than intended. This can be done by adjusting the time-lock value in the transaction or by manipulating the system time of the nodes involved in the transaction.
		False Top-Up Attack	High	An attacker manipulates the network to make it appear as if a user's account has been topped up with more funds than it has. This can be done by creating a false transaction record or manipulating the network's records to indicate that an account has more funds than it does.
		Rug Pull Attack	High	A type of attack on decentralized finance (DeFi) protocol where a malicious actor, or group of actors, who controls a significant amount of the liquidity in a specific market or liquidity pool, suddenly withdraws that liquidity, causing the value of the assets in the pool to drop significantly. This can lead to significant financial losses for other market participants.

Flow To Work With Clients



Repeat until all test cases are completed

Initial engagement

Communicating with the client to understand their requirements and setting the scope of the project

Project planning

Based on the requirements create a detailed plan for the assessment, including the methodology and tools to be used

Assessment

Conducting the actual audit, which includes source code review and dynamic tests

Real-time report

Reporting vulnerabilities immediately after finding out via CyStack vuln management platform

Patching

The client fixes issues through the recommendations from CyStack

Final report

Presenting the findings in a detailed report that includes an executive summary and detailed vulnerabilities

Follow-up

Communicating with the client to ensure that all recommendations are implemented and verifying that the system is secure

Closure

Archiving project-related data and officially closing the project

About CyStack

CyStack is an innovative company in the field of cybersecurity in Vietnam. We are a pioneer in building next gen security products for businesses and individuals. Our solutions focus on data protection, cyber attack prevention, and security risk management.

For more information, please call **(+84) 247 109 9656** or send an email to **contact@cystack.net** to speak to CyStack security specialist.
cystack.net