Quantstamp Security Assessment Certificate

QUANTSTAMP VERIFIED SECURITY CERTIFICATE

June 1st 2021 — Quantstamp Verified

Harmony BUSD

This smart contract audit was prepared by Quantstamp, the leader in blockchain security.

Executive Summary

ALL ISSUES ADDRESSED

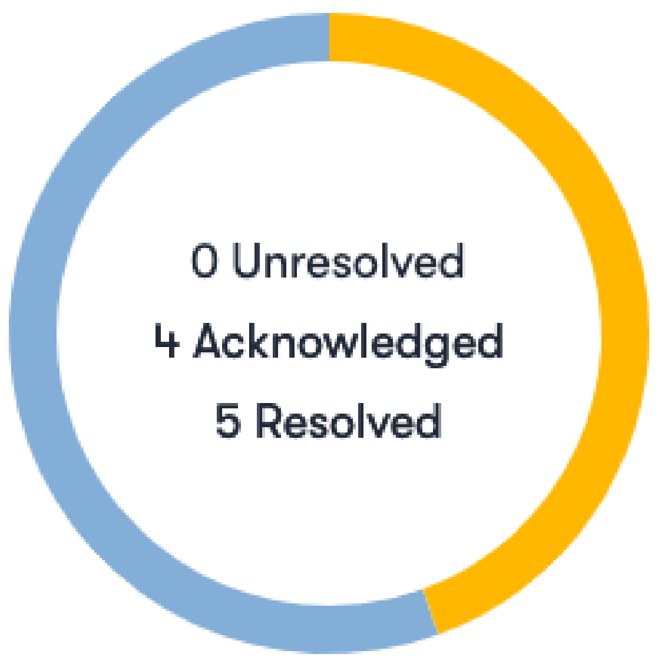
Type	Smart contract, deployment scripts
Auditors	Alex Murashkin, Senior Software Engineer Martin Derka, Senior Research Engineer
Timeline	2020-06-15 through 2020-07-07
EVM	Harmony (blockchain)
Languages	Solidity, Javascript
Methods	Manual Review
Specification	None
Documentation Quality	Undetermined
Test Quality	Undetermined
Source Code	

Repository	Commit
busd-contract	harmony-one (diff only)
busd-contract	cf8206a (diff)

Goals

- To give an assurance that the changes made to the upstream repo will not affect the underlying security of the contract code

Total Issues	9	(5 Resolved)
High Risk Issues	3	(3 Resolved)
Medium Risk Issues	1	(0 Resolved)
Low Risk Issues	3	(0 Resolved)
Informational Risk Issues	2	(2 Resolved)
Undetermined Risk Issues	0	(0 Resolved)



High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.

Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

Upon manual review, we discovered nine potential issues. We classified five of the issues as high-severity, two issues - as medium-severity, and two - as low-severity. Please find the details in the table and the *Assessment* section below.
In addition, we note the following:

1. From the EVM execution perspective, since there are no changes to the Solidity code in the given diff (`https://github.com/paxosglobal/busd-contract/compare/master...harmony-one:master`), under the assumptions that the Ethereum and Harmony blockchain EVMs operate in an identical manner and the smart contracts were initialized correctly, the newly deployed bytecode would preserve the runtime security properties of the original BUSD contract.
2. From the general blockchain security perspective, however, it should be noted that Ethereum and Harmony are two different blockchains with different architectures, governance models, and crypto-economic security properties. Therefore, smart contacts, in general, may not have exactly the same levels of security on both blockchains: e.g., certain types of blockchain-level attacks may be more feasible on one blockchain than another. Assessment of general, blockchain-level attacks is not in the scope of this audit.

Update: the Harmony team has addressed all the findings. The severities of QSP-3, QSP-5, and QSP-7 were lowered after discussing them with the team.

ID	Description	Severity	Status
QSP-1	Potentially incorrect (stale) ownership account information in migration scripts	︽ High	Fixed
QSP-2	Potentially sensitive information stored in environment configuration	︽ High	Fixed
QSP-3	Potential storage of real keys in the simulated keystore	○ Informational	Fixed
QSP-4	Domain separator not initialized by default	︽ High	Fixed
QSP-5	Potentially sensitive account information in code comments	○ Informational	Fixed
QSP-6	Recommended use of hardware wallets	︿ Medium	Acknowledged
QSP-7	Insecure approach to key management in the server code	﹀ Low	Acknowledged
QSP-8	Non-atomic deployment/initialization of the contracts	﹀ Low	Acknowledged
QSP-9	Production unreadiness of the server app	﹀ Low	Acknowledged

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the provided diff `https://github.com/paxosglobal/busd-contract/compare/master...harmony-one:master` for security-related issues.

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Checked if ownership information in deployment scripts is correct
2. Checked if private keys or other sensitive information is stored in the repository
3. Checked if secure deployment practices (such as the use of cold storage wallers) are followed
4. Checked if deployment could be interfered with or taken over by a third-party.

Findings

QSP-1 Potentially incorrect (stale) ownership account information in migration scripts

Severity: High Risk

Status: Fixed

File(s) affected: `migrations/2_deploy_contracts.js`

Description: In `busd-contract/migrations/2_deploy_contracts.js`, `L10`, in the call `await proxy.changeAdmin("0xf0b1eef88956b0a307fa87b5f5671aad6a5d330f");` the account address `0xf0b1eef88956b0a307fa87b5f5671aad6a5d330f` is the same as in the fork's source repo: `https://github.com/paxosglobal/busd-contract/blob/master/migrations/2_deploy_contracts.js#L10`. If the Harmony team does not have a private key for `0xf0b1eef88956b0a307fa87b5f5671aad6a5d330f`, or is not the only entity with access to this key, the deployed proxy would be owned by an undesired account, or by a third party.

Recommendation: Change the address to an address that is exclusively owned by the Harmony team.
Update: the account has been updated as of commit `c09ebc5`.

QSP-2 Potentially sensitive information stored in environment configuration

Severity: High Risk

Status: Fixed

File(s) affected: `(Multiple files)`

Description: It is not considered to be a good practice to keep `.env` files into the repository. The `.env` file may contain sensitive information, and if any of it refers to the actual accounts that are used for deployment or administration of the BUSD contract, the security of the contract may also be compromised.

Recommendation: It is recommended to:

1. Ensure that none of the potentially sensitive information pieces refer to actual accounts on the main network.
2. Include environment configuration files such as `.env` in `.gitignore`, to help avoid accidentally committing sensitive information after manual editing.

Update: the sensitive file has been updated as of commit `c09ebc5`.

QSP-3 Potential storage of real keys in the simulated keystore

Severity: *Informational*

Status: Fixed

File(s) affected: `server/simulated-keystore.js`

Description: The simulated keystore file (`server/simulated-keystore.js`) may contain sensitive information, and if any of it refers to the actual accounts that are used for deployment or administration of the BUSD contract, the security of the contract may also be compromised. While it is well-understood that this is an example code, it is not unlikely for a developer to accidentally store a real key in such a file and commit it into the public repo.

Recommendation: It is recommended to:

1. Ensure that none of the potentially sensitive information pieces refer to actual accounts on the main network.
2. Include certain files in `gitignore`, to help avoid accidentally committing sensitive information after manual editing.

Update: the simulated keystone file has been removed as of commit `c09ebc5`.

QSP-4 Domain separator not initialized by default

Severity: *High Risk*

Status: Fixed

File(s) affected: `contracts/BUSDImplementation.sol`

Description: A note that in `contracts/BUSDImplementation.sol`, the `initializeDomainSeparator();` method is not called by the `initialize()` method by default. It implies that the `EIP712_DOMAIN_HASH` state variable remains unset in the context of the proxy contract until `initializeDomainSeparator();` is called. `EIP712_DOMAIN_HASH` is used in the `betaDelegatedTransferBatch(...)` functionality and if it remains unset, it could imply that the hash calculation in `contracts/BUSDImplementation.sol, L538` (`bytes32 hash = keccak256(abi.encodePacked(EIP191_HEADER, EIP712_DOMAIN_HASH, delegatedTransferHash));`) is incorrect.

Recommendation: Double-check if `EIP712_DOMAIN_HASH` needs to be initialized. If it is, suggesting adding moving `initializeDomainSeparator();` into the `initialize()` method, or adding a command to the deployment scripts to call `initializeDomainSeparator();` separately, as long as it is done prior to use of the delegate transfer functionality.
Update: the initialization step has been added as of commit `c09ebc5`.

QSP-5 Potentially sensitive account information in code comments

Severity: *Informational*

Status: Fixed

File(s) affected: `server/app.js`

Description: In `server/app.js, L36-41`: it is not recommended to keep the account ids in code comments. I could be high-risk if the account ids refer to the real account ids.

Recommendation: Remove the account information from the comments.
Update: the Harmony team has provided an explanation stating that the account ids are public. We lowered the severity and marked this as Resolved.

QSP-6 Recommended use of hardware wallets

Severity: *Medium Risk*

Status: Acknowledged

Description: For Ethereum, it is a good practice to use cold storage (hardware) wallets for deployment or administration of accounts on the main network. The same practice could be applied to the Harmony contracts: it is recommended to not use private keys that could accidentally leak into the public or be committed into the GitHub repository.

Recommendation: To ensure maximum security of the BUSD contract, consider the use of cold storage wallets, either for deployment, administration (i.e., transferring ownership to a cold storage account right after the deployment), or both.
Update: the Harmony team has provided an explanation: "We confirm that we understand this recommendation and use hardware wallet for BUSD deployment in mainnet for better security."

QSP-7 Insecure approach to key management in the server code

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `server/*`

Description: It is not recommended to store private key information as a single, atomic piece of information. When a private key string is leaked (e.g., could be accidentally copied and pasted online), it compromises the account right away.

Recommendation: Looking into using Ethereum keystore files (`https://medium.com/@julien.maffre/what-is-an-ethereum-keystore-file-86c8c5917b97`). Typically, it is a JSON file that contains encrypted private key information, and the encryption key is provided as an environment variable. This way, there are two pieces of information, and if they are stored separately, the risk of both the keystore JSON and the encryption key being leaked at the same time is lower.
Update: the Harmony team has provided an explanation stating that the server code is example code and will not be used for production.

QSP-8 Non-atomic deployment/initialization of the contracts

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `migrations/2_deploy_contracts.js`

Description: Deployment/initialization of the BUSD contacts is not atomic. According to `migrations/2_deploy_contracts.js`, first, the BUSD implementation contract is deployed. Next, the proxy is deployed and linked (atomically) with the implementation contract. Next, admin transfer is performed. And finally, proxy initialization happens.

In theory, any of these steps could fail, be unintentionally delayed, or be attempted to be interfered with. For example, a malicious third-party could send the `initialize(...)` transaction before the intended owner does, or attempt to temporarily DOS the network to attempt to prevent the intended ownership transfer.

Recommendation: While we do not see any issues with racing the transactions, there is a risk of the contract not being deployed fully. It is recommended to manually check the success of execution each step: confirm that the ownership was indeed transferred and that the initialization has, in fact, happened.
Update: the Harmony team has provided a recommendation in the README.

QSP-9 Production unreadiness of the server app

Severity: *Low Risk*

Status: Acknowledged

Description: While it is written in the README that the server app is not production-ready, we would like to highlight some other aspects of the app, such as hard-coded has limits and gas prices, lack of request throttling, transactions potentially ending up on uncle blocks, etc.

Recommendation: It is recommended to keep the `README` note stating the `server` app remains to be an example and is not production-ready.
Update: the Harmony team has provided an explanation stating that the server code is example code and will not be used for production.

Changelog

- 2020-06-16 - Initial report
- 2020-07-07 - Diff report (`cf8206a`)

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp, Inc. (Quantstamp). Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on the website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.