

USDT0 · SECURITY REVIEW

USDT0

Polygon Deployment and Peer Verification.

A security review of Polygon Deployment and Peer Verification.

CLIENT	PROJECT	AUDITORS	AUDIT WINDOW	FINAL REPORT
USDT0	Polygon Deployment and Peer Verification	Osman Ozdemir, Felipe Gomez, Andreas Zottl	August 5 – August 7, 2025	August 7, 2025

§ 00

Contents.

19 pages: overview, confidence ranking, scope, 11 findings, disclaimer, and about Guardian.

§ 01	Overview	P. 03
§ 02	Confidence Ranking	P. 04
§ 03	Scope & Classifications	P. 05
§ 04	Findings	P. 06
	Main Review · 11 findings	P. 07
§ 05	Disclaimer	P. 18
§ 06	About Guardian	P. 19

§ 01

Overview.

Guardian reviewed Polygon Deployment and Peer Verification across a single review round. The final report includes 4 Low, 7 Informational findings.

AUDIT FIRM	REPOSITORY
Guardian	https://github.com/Everdawn-Labs/pos-portal , https://github.com/maticnetwork/pos-portal
CLIENT	METHODOLOGY
USDT0	Smart contract · Manual · Main Review
PROJECT	
Polygon Deployment and Peer Verification	
PREPARED BY	
Osman Ozdemir, Felipe Gomez, Andreas Zottl	
AUDIT WINDOW	
August 5 – August 7, 2025	
FINAL REPORT	
August 7, 2025	

1.1 Confidence Ranking

5 / 5

Very High Confidence

Guardian assigns a Confidence Ranking of 5 out of 5, combining finding severity, remediation outcome, and the reviewer’s assessment of latent risk. The full scale is on the next page.

1	2	3	4	5
VERY LOW CONFIDENCE	LOW CONFIDENCE	MODERATE CONFIDENCE	HIGH CONFIDENCE	VERY HIGH CONFIDENCE

1.2 Findings at a Glance

SEVERITY	TOTAL	PENDING	ACK	PARTIAL	RESOLVED
Critical	0	0	0	0	0
High	0	0	0	0	0
Medium	0	0	0	0	0
Low	4	3	1	0	0
Informational	7	5	1	0	1
All severities	11	8	2	0	1

§ 02

Confidence Ranking.

Guardian assigns a project-level confidence score using the scale below. The ranking combines finding severity, remediation outcome, and the reviewer's assessment of latent risk.

CONFIDENCE RANKING	DEFINITION AND RECOMMENDATION	RISK PROFILE
5 Very High Confidence ASSIGNED	Codebase is mature, clean, and secure. No High or Critical vulnerabilities were found. Follows modern best practices with high test coverage and thoughtful design. Recommendation: code is highly secure at time of audit.	0 High/Critical findings and few Low/Medium severity findings.
4 High Confidence	Code is clean, well-structured, and adheres to best practices. Design patterns are sound and test coverage is strong. Recommendation: suitable for deployment after remediations; consider periodic review with changes.	0-1 High/Critical findings per engagement week and little to no Medium severity findings.
3 Moderate Confidence	Code is functional but shows gaps in defensive programming or test coverage. Recommendation: remediate the findings and re-review the changes before deployment.	Some High/Critical findings, with various Medium severity findings.
2 Low Confidence	Recurring issues suggest systemic gaps rather than isolated mistakes. Recommendation: remediate, then commission a follow-up review at a frozen commit.	Over 6 High and Critical findings across the engagement.
1 Very Low Confidence	The volume and severity of findings indicate the codebase is not ready for production. Recommendation: substantial rework, then a full re-review.	Over 12 High and Critical findings across the engagement.

§ 03

Scope & Classifications.

Guardian reviewed Polygon Deployment and Peer Verification in

<https://github.com/Everdawn-Labs/pos-portal>,

<https://github.com/maticnetwork/pos-portal>.

IN SCOPE

Transactions:

https://docs.google.com/spreadsheets/d/1yS43NI9ncC6XhnFpu2rqODYx_1cpeDvepVzngWWVf5k/edit?gid=0#gid=0

USDT0 - Proxy: 0xc2132D05D31c914a87C6611C10748AEb04B58e8F

USDT0 - Implementation: 0x90040487a6c9f949c4f07cadcfb0f3b8eeab4229

USDT0 - Proxy Owner: 0x4DFF9b5b0143E642a3F63a5bcf2d1C328e600bf8

USDT0-OFT - OFT: 0x6BA10300f0DC58B7a1e4c0e41f5daBb7D7829e13

USDT0-OFT - Implementation: 0x8412d6553a2bd1b547bc48b86c3c5d897122ae47

USDT0-OFT - Proxy Admin: 0xa882c21C9df00958A958cde96f2B2Ae8FB4315B1

3.1 Severity Matrix

	IMP · HIGH	IMP · MED	IMP · LOW
LIK · HIGH	Critical	High	Medium
LIK · MED	High	Medium	Low
LIK · LOW	Medium	Low	Low

IMPACT · HIGH

Significant loss of assets, harm to a group of users, or core protocol functionality is disrupted.

IMPACT · MEDIUM

Small amount of funds can be lost, or ancillary functionality is affected. Users may experience reduced or delayed receipt.

IMPACT · LOW

Unexpected behavior in some functionality that is notable but does not meet the criteria for higher severity.

LIKELIHOOD · HIGH

Possible with reasonable assumptions that mimic on-chain conditions; cost is low compared to amount gained.

LIKELIHOOD · MEDIUM

Possible only in uncommon

§ 04

Findings.

Grouped by round. Visible IDs reset by severity within each round, so a finding is named the same way in the report as it was in the channel it was written in. Status is where each finding ended up; its own page says how.

ID	TITLE	CATEGORY	SEVERITY	STATUS	PAGE
MAIN REVIEW					
L-01	Incorrect Pre-generated Calldata	Logical Error	LOW	PENDING	p. 7
L-02	Avoid Using Private Keys	Best Practices	LOW	ACKNOWLEDGED	p. 8
L-03	Missing Block Validation	Validation	LOW	PENDING	p. 9
L-04	Incorrect Role Revoked In Rollback	Logical Error	LOW	PENDING	p. 10
I-01	Double Check JSON Inputs	Configuration	INFORMATIONAL	PENDING	p. 11
I-02	Warning About Using Same Script	Informational	INFORMATIONAL	PENDING	p. 12
I-03	Manual Script Steps Prone To Errors	Informational	INFORMATIONAL	PENDING	p. 13
I-04	Insolvency Risk For OFT Users	Logical Error	INFORMATIONAL	RESOLVED	p. 14
I-05	Mismatch Between Readme And Code	Informational	INFORMATIONAL	PENDING	p. 15
I-06	Missing Migration Event	Events	INFORMATIONAL	ACKNOWLEDGED	p. 16
I-07	Manager Actions Allowed After Migration	Informational	INFORMATIONAL	PENDING	p. 17

LOW **L-01**

PENDING

Incorrect Pre-generated Calldata

CATEGORY	Logical Error
LOCATION	https://github.com/GuardianOrg/pos-portal-usdt0-polygon-team1/blob/e6438f46f791381befb5d7d16528bed270cf43a4/playbook-usdt0.md#L78-L83
REVIEW	Main Review

DESCRIPTION

There is a rollback plan that aims to revert the proxies to the old implementation in case of a bug. This rollback plan uses pre-generated calldata.

While the pre-generated calldata for the `ERC20Predicate` uses the `updateImplementation` method to schedule and execute the rollback, the calldata for the `RootChainManager` uses the `updateAndCall` method with additional data to perform a `grantRole` operation after the rollback. However, the `grantRole` function cannot be executed due to access control restrictions, and as a result, the current pre-generated calldata fails to deliver the rollback action in case a bug occurs.

RECOMMENDATION

The pre-generated rollback calldata for `RootChainManager` should be updated to call only the `updateImplementation` method.

RESOLUTION

Status: Pending.

LOW **L-02**

ACKNOWLEDGED

Avoid Using Private Keys

CATEGORY Best Practices

LOCATION <https://github.com/maticnetwork/pos-portal/blob/feat/disable-token-bridging/scripts/forge/UpdateImplementation.s.sol#L120>

REVIEW Main Review

— DESCRIPTION

The current preparation scripts optionally deploy new implementation contracts for `RootChainManager`, `ERC20Predicate`, and `UChildUSDT`. To deploy these contracts, a `PRIVATE_KEY` must be specified in the `.env` file or via inline environment variable assignment. This approach risks exposing and compromising private keys.

— RECOMMENDATION

Use a hardware wallet or encrypted keystore file.

— RESOLUTION

Status: Acknowledged.

LOW **L-03**

PENDING

Missing Block Validation

CATEGORY Validation

LOCATION <https://github.com/maticnetwork/pos-portal/blob/feat/disable-token-bridging/contracts/root/RootChainManager/RootChainManager.sol#L439>

REVIEW Main Review

DESCRIPTION

The `updateTokenMigrationStatus` will disable deposits and/or exits for a specific root token. Although the `UpdateTokenMigrationStatus` script validates non zero `lastExitBlockNumber` when disabling exits, this is not the case for the on chain validation.

RECOMMENDATION

Consider validating `lastExitBlockNumber > 0` in `updateTokenMigrationStatus` function.

RESOLUTION

Status: Pending.

LOW **L-04**

PENDING

Incorrect Role Revoked In Rollback

CATEGORY	Logical Error
LOCATION	https://github.com/GuardianOrg/pos-portal-usdt0-polygon-team1/blob/e6438f46f791381befb5d7d16528bed270cf43a4/playbook-usdt0.md#L64-L65
REVIEW	Main Review

DESCRIPTION

The USDT0 playbook includes rollback provisions intended for emergency scenarios. However, as per what documented, the `MIGRATION_MANAGER_ROLE` access is revoked from the zero address (0x0) rather than the actual address to which permissions were granted. This oversight leaves the originally authorized address with unintended and unclear permissions.

RECOMMENDATION

Consider revoking permissions from the address that was originally granted `MIGRATION_MANAGER_ROLE` access.

RESOLUTION

Status: Pending.

INFORMATIONAL I-01

PENDING

Double Check JSON Inputs

CATEGORY Configuration

LOCATION <https://github.com/maticnetwork/pos-portal/blob/feat/disable-token-bridging/scripts/forge/inputs.json.example>

REVIEW Main Review

— DESCRIPTION

The preparation and migration steps involve using different foundry scripts that heavily rely on a JSON file. Although the `inputs.json.example` is not the final JSON that will be used during migration, it already contains wrong addresses.

— RECOMMENDATION

Due to the fact that scripts will heavily rely on this JSON file, extra care should be taken to verify that each value is correct.

— RESOLUTION

Status: Pending.

INFORMATIONAL I-02

PENDING

Warning About Using Same Script

CATEGORY	Informational
LOCATION	https://github.com/maticnetwork/pos-portal/blob/2b73157a2ba548ef1da82ff291f16898e99e6d5f/scripts/forge/UpdateImplementation.s.sol#L96-L101
REVIEW	Main Review

— DESCRIPTION

According to the upgrade `README`, both the `RootChainManager` and `ERC20Predicate` implementation upgrades will be performed using the same `UpdateImplementation.s.sol` script. The script can be run either by providing an input string or without any input. In the former case, it reads the necessary data from the provided input string; in the latter, it reads data from a previously prepared `inputs.json` file in the `scripts/forge` folder.

Since the `inputs.json` file can only contain one `upgradeImplementation` key, and both upgrades use the same script, at least one of the upgrades must be performed using an input string, while the other can be read from the `inputs.json` file.

The same case applies for `UpdateTokenMigrationStatus` script as well.

— RECOMMENDATION

Consider using separate scripts for each upgrade so that all inputs can be prepared beforehand in the `inputs.json` file. Alternatively, modify `_readInputs` to read different keys from `inputs.json` based on whether the upgrade is for `RootChainManager` or `ERC20Predicate`.

— RESOLUTION

Status: Pending.

INFORMATIONAL I-03

PENDING

Manual Script Steps Prone To Errors

CATEGORY Informational

LOCATION <https://github.com/maticnetwork/pos-portal/tree/feat/disable-token-bridging/scripts/forge>

REVIEW Main Review

— DESCRIPTION

The current playbook describes the migration steps for USDT0. These include generating calldata for creating multisig transactions, reading from a JSON file, copy and pasting results from console, testing bridge deposits and exits.

The amount of manual steps may lead to human errors, due to the amount of scripts to run, pasting the incorrect value in the JSON file or into the multisig UI for building the transaction.

— RECOMMENDATION

Consider reducing the manual inputs as much as possible. Additionally, reduce the amount of scripts to run and/or use the forge script to write the output into the JSON file.

If possible, create the multisig transaction directly in the script, using the generated calldata. This avoids any UI phishing as you will be interacting with the smart contract directly.

— RESOLUTION

Status: Pending.

INFORMATIONAL I-04

RESOLVED

Insolvency Risk For OFT Users

CATEGORY	Logical Error
LOCATION	https://github.com/maticnetwork/pos-portal/blob/feat/disable-token-bridging/contracts/root/TokenPredicates/ERC20Predicate.sol#L113
REVIEW	Main Review

DESCRIPTION

The `migrateBridgeFunds` will transfer out USDT from the `ERC20Predicate` according to the collected USDT supply value on Polygon during the upgrade.

However, these funds will be sent to a `Multisig` address according to the playbook, instead of the the USDT `OAdapter`. Additionally, the playbook does not contain any extra steps regarding a subsequent deposit of USDT to another contract.

This will create an insolvency case, as there will be more USDT0 circulating in LayerZero chains than the amount of USDT available in the `OAdapter` to be redeemed.

RECOMMENDATION

After migrating USDT out of the `ERC20Predicate`, transfer them to the `OAdapter` contract, or set it as receiver in the encoded transfer call

RESOLUTION

Status: Resolved.

INFORMATIONAL I-05

PENDING

Mismatch Between Readme And Code

CATEGORY	Informational
LOCATION	https://github.com/maticnetwork/pos-portal/blob/2b73157a2ba548ef1da82ff291f16898e99e6d5f/contracts/root/RootChainManager/RootChainManager.sol#L387-L388
REVIEW	Main Review

DESCRIPTION

Step 8 of the playbook README mentions that exits should be blocked from the upgrade block: "New plan is to just run the upgrade in block x, block exits from block x on (so no tx from x is valid exit) and manually check for any valid exits before the upgrade".

However, the code allows exits in the upgrade block and considers them valid [Reference]

(<https://github.com/maticnetwork/pos-portal/blob/2b73157a2ba548ef1da82ff291f16898e99e6d5f/contracts/root/RootChainManager/RootChainManager.sol#L387-L388>).

```
``plain text
if (migrationStatus[rootToken].isExitDisabled &&
blockNumber > migrationStatus[rootToken].lastExitBlockNumber) { // @note: if the
lastExitBlockNumber is the same as the current block number, exits are still allowed
revert("RootChainManager: EXIT_DISABLED");
}
``
```

RECOMMENDATION

Update the code accordingly if the README is correct and the intention is to disallow exits in the upgrade block. However, note that this change would cause exits within the same block to fail and require manual intervention.

Otherwise, update the README to reflect the current behavior.

RESOLUTION

Status: Pending.

INFORMATIONAL I-06

ACKNOWLEDGED

Missing Migration Event

CATEGORY Events

LOCATION <https://github.com/maticnetwork/pos-portal/blob/feat/disable-token-bridging/contracts/root/RootChainManager/RootChainManager.sol#L468>

REVIEW Main Review

DESCRIPTION

During funds migration flow, neither `RootChainManager.migrateBridgeFunds` or `ERC20Predicate.migrateTokens` emit an event.

This is a major state change that should emit relevant data, like it's currently done in the `updateTokenMigrationStatus` function.

Offchain services will likely need to inspect the migration transaction in order to read the USDT token receiver and amount withdrawn.

RECOMMENDATION

Consider emitting an event either in `RootChainManager` or `ERC20Predicate`, with the token receiver and amount.

RESOLUTION

Status: Acknowledged.

INFORMATIONAL I-07

PENDING

Manager Actions Allowed After Migration

CATEGORY Informational

LOCATION <https://github.com/maticnetwork/pos-portal/blob/feat/disable-token-bridging/contracts/root/RootChainManager/RootChainManager.sol#L473>

REVIEW Main Review

DESCRIPTION

The `MIGRATION_MANAGER_ROLE` can block deposit/exits and later `migrateBridgeFunds` to withdraw tokens from `ERC20Predicate`. Blocking deposits/exits does not have any validation, and migration is only allowed if both deposits and exits are blocked.

This means that manager may:

- call `migrateBridgeFunds` multiple times
- block deposits and exits to later migrate tokens, but re enable deposits or exits in the future again.

This might be unexpected, as migration should be a one time action.

RECOMMENDATION

Consider setting a state mapping that tracks migrated tokens, preventing manager to re-migrate or re-enable deposits/exits after migration.

If this is the expected behavior, document it in the `RootChainManager` contract.

RESOLUTION

Status: Pending.

§ 05

Disclaimer.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Guardian to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Guardian's position is that each company and individual are responsible for their own due diligence and continuous security. Guardian's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by Guardian is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis.

Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract's safety and security. Therefore, Guardian does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

§ 06

About Guardian.

Guardian is a smart contract security firm. Every engagement is worked by two independent teams competing on the same scope, each with its own tooling and methods, so no single reviewer's blind spot survives.

\$45B+DIGITAL ASSETS UNDER
REVIEW**200+**

ENGAGEMENTS COMPLETED

500+CRITICAL & HIGH FINDINGS
REPORTED

Reach us at audits@guardianaudits.com.