

AdventureWurld Wurldlings Audit Summary

Prepared by Cyfrin

Version 2.0

Head of Security Research

Dacian (Proprietary AI)

Lead Auditors

Farouk

qpzm

November 3, 2026

Contents

1 About Cyfrin **2**

2 About This Document **2**

3 Disclaimer **2**

4 Copyright and License **2**

5 Audit Overview **3**

6 Audit Scope **3**

7 Risk Classification **4**

8 Protocol Summary **4**

8.1 Scoped Components 4

8.2 Mint Lifecycle 5

8.3 Bridge Lifecycle 5

9 Attestation **5**

1 About Cyfrin

Cyfrin is a Web3 security company dedicated to bringing industry-leading protection and education to our partners and their projects. Our goal is to create a safe, reliable, and transparent environment for everyone in Web3 and DeFi. Learn more about us at cyfrin.io.

2 About This Document

This document is a public summary of the security audit Cyfrin performed for AdventureWurld on Wurldlings. It records the repositories and commits reviewed, the timeline, auditors, in-scope files, and the number and final status of findings by severity. It closes with Cyfrin's attestation of the review's scope and final remediation status.

It is deliberately abbreviated: the technical detail of individual findings is not reproduced here. The complete audit report was delivered privately to AdventureWurld, who retain it as the full record of the engagement.

3 Disclaimer

The Cyfrin team makes every effort to find as many vulnerabilities in the code as possible in the given time but holds no responsibility for the findings in this document. A security audit by the team does not endorse the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the in-scope code being audited. Unless the scope explicitly notes otherwise, the review is limited to the application layer and does not extend to the correctness of API/SDK/execution environment implementations, compilers, standard and third-party libraries, underlying operating systems and hardware.

4 Copyright and License

© 2026 Cyfrin Inc. All rights reserved. Material from AdventureWurld or third parties remains the property of its owners.

This summary may be shared only in complete and unmodified form, so that any copy can be checked against the hash in its attestation. Short quotations with attribution to Cyfrin are permitted, provided they do not misrepresent this summary's scope or results. Except as permitted by applicable law, no part of this summary may otherwise be copied, modified, adapted, sold or sublicensed, or used to create derivative works, without Cyfrin's prior written permission.

Cyfrin and the Cyfrin logo are trademarks of Cyfrin Inc. and may not be used to suggest that Cyfrin has reviewed or endorses anything outside this summary's scope, or to misrepresent its results.

5 Audit Overview

Over the course of 12 days, the Cyfrin team conducted an audit on the Wurldlings code provided by AdventureWurld. The findings consist of 1 Medium & 20 Low severity issues with the remainder being informational.

Summary

Project Name	Wurldlings
Repository	REDACTED
Commit	bfe2d844e8ab...
Fix Commit	286d8b6c0386...
Audit Timeline	Oct 12th - Oct 27th, 2026
Methods	Manual Review, Proprietary AI

Issues Found

Severity	Found	Resolved	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	1	1	0
Low Risk	20	20	0
Informational	10	10	0
Gas Optimizations	0	0	0
Total	31	31	0

6 Audit Scope

The audit scope was limited to the following files:

```
contracts/nft/bridge/NftBridge.sol
contracts/nft/items/ItemCatalog.sol
contracts/nft/items/ItemCollection.sol
contracts/nft/items/ItemSale.sol
contracts/nft/wurldling/randomness/vendor/bls-solidity/BLS2.sol
contracts/nft/wurldling/randomness/vendor/bls-solidity/Precompiles.sol
contracts/nft/wurldling/randomness/DrandQuicknetBeacon.sol
contracts/nft/wurldling/Wurldling.sol
contracts/nft/wurldling/WurldlingGeneration.sol
contracts/nft/wurldling/WurldlingGenerationDirect.sol
contracts/nft/wurldling/WurldlingSale.sol
contracts/nft/Aw721Base.sol
contracts/nft/Aw721Transferable.sol
contracts/nft/Bridgeable.sol
```

7 Risk Classification

	Impact: High	**Impact: Medium**	**Impact: Low**
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

8 Protocol Summary

AdventureWurld is a free-to-play web3 gaming platform that runs on its own private EVM chain, a Besu network. The audited contracts are its NFT subsystem: the Wurldling character collection, whose traits are rolled on-chain from drand randomness; a sibling collection of purchasable items; and `NftBridge`, which moves tokens between the private chain and public chains. New tokens are minted only on the AdventureWurld chain. Vaults on Ethereum and Base receive them by bridge, so that holders can trade them on public marketplaces.

8.1 Scoped Components

Component	Role
`Aw721Base`, `Aw721Transferable`, `Bridgeable`	The immutable ERC-721 vault base. Each token belongs to a registered family and carries an opaque payload that the family contract renders into the on-chain `tokenURI` JSON. `Aw721Transferable` adds a non-transferable (ERC-5192 locked) mint path, and `Bridgeable` adds the burn-here, mint-there seam (`bridgeBurn`, `bridgeMint`) that only the configured bridge can call.
`Wurldling`	The Wurldling vault. It adds the roll mint: it verifies the drand signature for a commit's round through the configured beacon, derives randomness from it, has the generation roll the traits, and mints.
`WurldlingSale`	The swappable sale module, holding all funds and sale policy: pricing, whitelists, quantity caps, commit-reveal, refunds and proceeds. A sale fix ships as a new sale wired with `Wurldling::setSale`, with no holder migration.
`WurldlingGeneration`, `WurldlingGenerationDirect`	Family contracts. A generation holds a catalog of tiers, traits and conditional trait groups, sealed before registration, and rolls a trait set from the randomness; the direct family stores manually minted attributes.
`DrandQuicknetBeacon`, `BLS2`, `Precompiles`	Verifies drand quicknet beacons on-chain: BLS12-381 signatures in G1 over the round number, checked with the EIP-2537 precompiles, using a BLS library vendored from upstream with local changes.
`ItemCollection`, `ItemCatalog`, `ItemSale`	The items collection. Each family's catalog defines items as rows that seal at their first mint, and the sale sells them at fixed prices, numbering each item's tokens and limiting non-transferable items to one per wallet.
`NftBridge`	The production bridge, adapted from AdventureWurld's earlier ERC-20 bridge. It burns a token on the source chain and mints it on the destination once an M-of-N set of releasers has attested the burn, with fees, routes, pause, and a queue of parked releases.

8.2 Mint Lifecycle

1. **Commit.** A buyer calls `WurldlingSale::commitMint` and pays for a quantity of a generation. The commit is pinned to a drand round, `roundAt(block.timestamp) + roundOffset`, whose randomness has not been published yet, so nobody, the operator included, can know the outcome when paying.
2. **Reveal.** Once that round is published, anyone can call `WurldlingSale::finalizeMint` with its signature; finalization is permissionless by design, and AdventureWurld runs a finalizer service. The vault verifies the signature, the generation rolls each token's traits from the derived randomness, and the vault mints and emits `MintedRolledAttributes`, which AdventureWurld's off-chain renderer uses to produce the token images.
3. **Refund.** The owner can refund any pending commit through `WurldlingSale::refundCommit`, and the committer can refund its own commit through `WurldlingSale::refundUnfulfillable` once the generation's remaining supply has fallen below the commit's quantity. A refund the recipient cannot receive is held as a balance that anyone can pay out to the recipient through `WurldlingSale::claimRefund`.

8.3 Bridge Lifecycle

1. **Lock.** The token's owner calls `NftBridge::lockToken` with a destination and a recipient, paying the bridge fee. The bridge burns the token on the source chain and emits the payload.
2. **Attest.** Releasers on the destination chain sign the release. The bridge has no light client or proof of the source burn, so its security rests on the honesty and liveness of the releaser set and its threshold, which is operated by an external team.
3. **Release.** Once the threshold is met, the bridge mints the token on the destination through `bridgeMint`. A release that cannot complete at once, for example because its release fee has not been paid, is parked in a per-recipient queue and completes later through `NftBridge::payReleaseFee`, `NftBridge::processPendingReleasesFor`, or the owner's `NftBridge::forceProcessPendingRelease`.

The game servers, the renderer and its image storage, the finalizer service and the releaser infrastructure are off-chain and outside this review, but users depend on them for gameplay, token images, timely finalization and bridge delivery.

9 Attestation

Cyfrin attests that it performed a security review of the Wurldlings code provided by AdventureWurld.

The repositories, commits, review period and findings with their final status are in the Summary and Issues Found tables in Section 5, and the in-scope files in Section 6. This attestation is limited to the documented scope, audited commits and mitigations specifically reviewed by Cyfrin.

Cyfrin publishes this document and its SHA-256 hash at https://github.com/Cyfrin/cyfrin-audit-reports. To check a copy on macOS or Linux, run the following in the folder that contains it; it prints `OK` when the copy matches. If the copy was renamed, set F to its file name. Before running it, check that the URL is Cyfrin's repository.

```
U=https://github.com/Cyfrin/cyfrin-audit-reports/raw/main/reports/sha256
N=2026-11-03-cyfrin-adventurewurld-wurldlings-v2.0.pdf
F=$N
H=$(curl -fsSL $U/$N.sha256 | cut -d' ' -f1)
echo "$H *$F" | shasum -a 256 -c
```

Cyfrin's full audit report with all findings provided to AdventureWurld has the SHA-256 hash and file name below. To check a copy of it, first check this document as above, then run the following in the folder that contains the full report; it prints `OK` when the copy matches. If the copy was renamed, set F to its file name.

```
H=96bc90022e1d7c70728a214c50a66f03802a9e0fddff7c6b0054125a7dec4aee
F=2026-11-03-cyfrin-adventurewurld-wurldlings-v2.0-full.pdf
echo "$H *$F" | shasum -a 256 -c
```

On Linux, `sha256sum -c` can be used in place of `shasum -a 256 -c`. On Windows, run `Get-FileHash <file> -Algorithm SHA256` in PowerShell and compare the result with the hash.