AstraSec

Radpie

Security Audit Report

May 3, 2024

Contents

1 Introduction

1.1 About Radpie

Developed by Magpie, Radpie is a DeFi platform developed atop Radiant Capital, dedicated to delivering optimized yields and efficient governance services. Radpie offers a solution that empowers Radiant to maximize its long-term value. Radpie allows dLP holders to earn a share of the platform's revenue with no lock-up period required. It also provides Radiant voters with cost-effective voting power, and enables liquidity providers to earn RDNT rewards without the necessity to maintain a certain percentage of their deposits as dLP.

1.2 Audit Scope

First Audit Scope

The following source code was reviewed during the audit:

- https://github.com/magpiexyz/radpie_contracts/tree/CrossChain/CCIP/contracts/crosschain
- Commit ID: e0e891f

And this is the final version representing all fixes implemented for the issues identified in the audit:

- Commit ID: c0e5072

Second Audit Scope

The following source code was reviewed during the audit:

- https://github.com/magpiexyz/radpie_contracts/pull/112
- Commit ID: e0e891f

And this is the final version representing all fixes implemented for the issues identified in the audit:

- Commit ID: 3714383

Third Audit Scope

The following source code was reviewed during the audit:

- https://github.com/magpiexyz/radpie_contracts/pull/124
- Commit ID: 442db88

And this is the final version representing all fixes implemented for the issues identified in the audit:

- Commit ID: 442db88

Forth Audit Scope

The following source code was reviewed during the audit:

- https://github.com/magpiexyz/radpie_contracts/pull/129
- Commit ID: 0cf5607

And this is the final version representing all fixes implemented for the issues identified in the audit:

- Commit ID: 0cf5607

Fifth Audit Scope

The following source code was reviewed during the audit:

- https://github.com/magpiexyz/radpie_contracts/pull/120
- Commit ID: d60a85d

And this is the final version representing all fixes implemented for the issues identified in the audit:

- Commit ID: d60a85d

1.3 Changelog

Version	Date
First Audit	December 21, 2023
Second Audit	March 23, 2024
Third Audit	April 15, 2024
Forth Audit	April 16, 2024
Fifth Audit	May 3, 2024

2 Overall Assessment

This report has been compiled to identify issues and vulnerabilities within the `Radpie` project. Throughout this audit, we identified a total of 3 issues spanning various severity levels. By employing auxiliary tool techniques to supplement our thorough manual code review, we have discovered the following findings.

Severity	Count	Acknowledged	Won't Do	Addressed
Critical	-	-	-	-
High	-	-	-	-
Medium	1	1	-	-
Low	1	-	-	1
Informational	1	-	-	1
Undetermined	-	-	-	-

3 Vulnerability Summary

3.1 Overview

Click on an issue to jump to it, or scroll down to see them all.

M-1 Potential Risks Associated with Centralization

~~L-1~~ Suggested Necessary Initialization in esRDNT::__RadpieCC_init()

~~I-1~~ Meaningful Events for Key Operations

3.2 Security Level Reference

In web3 smart contract audits, vulnerabilities are typically classified into different severity levels based on the potential impact they can have on the security and functionality of the contract. Here are the definitions for critical-severity, high-severity, medium-severity, and low-severity vulnerabilities:

Severity	Description
C-X (Critical)	A severe security flaw with immediate and significant negative consequences. It poses high risks, such as unauthorized access, financial losses, or complete disruption of functionality. Requires immediate attention and remediation.
H-X (High)	Significant security issues that can lead to substantial risks. Although not as severe as critical vulnerabilities, they can still result in unauthorized access, manipulation of contract state, or financial losses. Prompt remediation is necessary.
M-X (Medium)	Moderately impactful security weaknesses that require attention and remediation. They may lead to limited unauthorized access, minor financial losses, or potential disruptions to functionality.
L-X (Low)	Minor security issues with limited impact. While they may not pose significant risks, it is still recommended to address them to maintain a robust and secure smart contract.
I-X (Informational)	Warnings and things to keep in mind when operating the protocol. No immediate action required.
U-X (Undetermined)	Identified security flaw requiring further investigation. Severity and impact need to be determined. Additional assessment and analysis are necessary.

3.3 Vulnerability Details

[M-1] Potential Risks Associated with Centralization

Target	Category	IMPACT	LIKELIHOOD	STATUS
Multiple Contracts	Security	Medium	Medium	Acknowledged

In the `Radpie` protocol, the existence of a series of privileged accounts introduces centralization risks, as they hold significant control and authority over critical operations governing the protocol. In the following, we show the representative function potentially affected by the privileges associated with the privileged accounts.

Radpie::mint()

```
function __Radpie_init(address deployer, uint256 _initialMint) public initializer {
    __ERC20_init("Radpie ", "RDP");
    __ERC20Burnable_init();
    __Pausable_init();
    __Ownable_init();

    _mint(deployer, _initialMint);
    _grantRole(DEFAULT_ADMIN_ROLE, deployer);
}

function mint(address to, uint256 amount) public onlyRole(MINTER_ROLE) {
    _mint(to, amount);
}
```

Remediation To mitigate the identified issue, it is recommended to introduce multi-sig mechanism to undertake the role of the privileged accounts. Moreover, it is advisable to implement timelocks to govern all modifications to the privileged operations.

Response By Team This issue has been confirmed by the team. The multi-sig mechanism will be used to mitigate this issue.

[L-1] Suggested Necessary Initialization in esRDNT::__RadpieCC_init()

Target	Category	IMPACT	LIKELIHOOD	STATUS
esRDNT.sol/Radpie.sol	Coding Practice	Low	Low	🔗Addressed

The `esRDNT` contract showcases proficient code implementation and organization through the utilization of several reference contracts. Notably, it enhances functionality by inheriting the `ERC20PermitUpgradeable` contract to support `EIP-2612`. However, during the examination of the `__RadpieCC_init()` function, we noticed the absence of a call to `ERC20PermitUpgradeable::__ERC20Permit_init()`, which is essential for the correct operation of `EIP-2612` support. It is recommended to include it within the initialization function to ensure proper functioning of the `permit` feature.

esRDNT::__RadpieCC_init()

```
function __RadpieCC_init(address deployer) public initializer {
    __ERC20_init("esRDNT Token", "esRDNT");
    __ERC20Burnable_init();
    __Pausable_init();
    __Ownable_init();
    _grantRole(DEFAULT_ADMIN_ROLE, deployer);
}
```

Remediation Properly execute `ERC20PermitUpgradeable::__ERC20Permit_init()` within the `esRDNT::__RadpieCC_init()` and `Radpie::__Radpie_init()` functions.

[I-1] Meaningful Events for Key Operations

Target	Category	IMPACT	LIKELIHOOD	STATUS
RadpieCCIPRouter.sol	Coding Practices	N/A	N/A	Addressed

The `event` feature is vital for capturing runtime dynamics in a contract. Upon emission, `events` store transaction arguments in logs, supplying external analytics and reporting tools with crucial information. They play a pivotal role in scenarios like modifying system-wide parameters or handling token operations.

However, in our examination of protocol dynamics, we observed that certain privileged routines lack meaningful events to document their changes. We highlight the representative routines below.

RadpieCCIPRouter

```
function setRouterAddress(address _router) external onlyOwner {
  if (_router == address(0)) revert AddressZero();
  chainlinkRouter = _router;
}

function whitelistChain(uint64 _destinationChainSelector) external onlyOwner {
    whitelistedChains[_destinationChainSelector] = true;
}

function denylistChain(uint64 _destinationChainSelector) external onlyOwner {
    whitelistedChains[_destinationChainSelector] = false;
}
```

Remediation Ensure the proper emission of meaningful events containing accurate information to promptly reflect state changes.

4 Appendix

4.1 About AstraSec

AstraSec is a blockchain security company that serves to provide high-quality auditing services for blockchain-based protocols. With a team of blockchain specialists, AstraSec maintains a strong commitment to excellence and client satisfaction. The audit team members have extensive audit experience for various famous DeFi projects. AstraSec's comprehensive approach and deep blockchain understanding make it a trusted partner for the clients.

4.2 Disclaimer

The information provided in this audit report is for reference only and does not constitute any legal, financial, or investment advice. Any views, suggestions, or conclusions in the audit report are based on the limited information and conditions obtained during the audit process and may be subject to unknown risks and uncertainties. While we make every effort to ensure the accuracy and completeness of the audit report, we are not responsible for any errors or omissions in the report.

We recommend users to carefully consider the information in the audit report based on their own independent judgment and professional advice before making any decisions. We are not responsible for the consequences of the use of the audit report, including but not limited to any losses or damages resulting from reliance on the audit report.

This audit report is for reference only and should not be considered a substitute for legal documents or contracts.

4.3 Contact

Phone	+86 176 2267 4194
Email	contact@astrasec.ai
Twitter	https://twitter.com/AstraSecAI