Stonewall
Smart Contract Security

DTreon Platform Security Review

Introduction

A time-boxed security review of the **DTreon Platform** contracts was conducted by **Stonewall**, with a focus on the security aspects of the smart contract implementation.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource and expertise bound effort where we try to find as many vulnerabilities as possible. We can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts. Subsequent security reviews, bug bounty programs and on-chain monitoring are strongly recommended.

About Stonewall

Stonewall is an independent smart contract security firm delivering immovable protection for Web3 protocols. Our team brings deep expertise in DeFi security, having reviewed DEXs, yield farming protocols, gaming contracts, and complex financial systems.

About DTreon

DTreon is a Web3 creator economy platform similar to Patreon, featuring:

- **Subscription Platform**: Tiered subscriptions with multiple payment tokens
- **PPV Content**: Pay-per-view content unlocking
- **DM System**: Paid direct messages between users and creators
- **Tips**: Direct tipping functionality
- **Profit Sharing Staking**: 25% of platform fees distributed to stakers
- **Platform Token**: Fixed supply ERC20 token

Privileged Roles & Actors

Role	Description
Owner	Can pause, set fees, manage accepted tokens
Creator	Creates tiers, sets DM fees, manages content
Platform Wallet	Receives 10% platform fees
Reward Distributors	Can add profit share to staking
Authorized Stakers	Can stake on behalf (airdrops)

Observations

- Multi-token payment support (BNB + ERC20 stablecoins)
- Creator can disable specific payment tokens
- Flexible subscription periods
- Lock-based staking multipliers (up to 3x for 1 year)

Risk Classification

	High Impact	Medium Impact	Low Impact
High Likelihood	Critical	High	Medium
Medium Likelihood	High	Medium	Low
Low Likelihood	Medium	Low	Low

Security Assessment Summary

Review Details	
Protocol Name	DTreon Platform
Repository	Private

Commit	`8ba1b4e382bcc8a669825b9843fe2e071343aefe`
Review Date	January 2026
Methods	Manual review, static analysis
Network	BNB Chain

Project Links

Platform	Link
Website	dtreon.com
Twitter	@Dtreon_Official
Discord	Join Discord
Instagram	@dtreon.official
YouTube	@Dtreon
Email	support@dtreon.com

Scope

Contract	SLOC
`main.sol` (SubscriptionPlatform)	~1260
`staking.sol` (ProfitSharingStaking)	~610
`token.sol` (PlatformToken)	~90
`jobs.sol`	~200

Findings Summary

ID	Title	Severity	Status
[H-01]	Payment processing is commented out - funds not transferred	High	Open

[M-01]	Mint function has no access control	Medium	Open
[M-02]	Stakers array unbounded leading to DoS	Medium	Open
[M-03]	Emergency withdraw bypasses early withdrawal penalty	Medium	Open
[L-01]	convertUSDToWei uses hardcoded BNB price	Low	Open
[L-02]	Tier deletion loop can fail with many tiers	Low	Open
[L-03]	No minimum subscription period enforced	Low	Open

Findings

[H-01] Payment processing is commented out - funds not transferred

Severity: High

Impact: Critical - Users pay but funds are never transferred to creators or platform.

Likelihood: High - Affects all payment flows.

Location: `main.sol:903-918` , `main.sol:819-830`

Description:

Multiple payment processing functions have the actual transfer logic commented out:

```
function _processPayment(
    address _creator,
    uint256 _totalAmount,
    address _paymentToken
) internal {
    uint256 platformFee = (_totalAmount * SUBSCRIPTION_PLATFORM_FEE) / BASIS_POINTS;
    uint256 creatorAmount = _totalAmount - platformFee;

    // if (_paymentToken == address(0)) {
    //     if (platformFee > 0) {
    //         payable(platformWallet).transfer(platformFee);
    //     }
    //     payable(_creator).transfer(creatorAmount);
    // } else {
    //     IERC20 token = IERC20(_paymentToken);
    //     token.safeTransferFrom(msg.sender, _creator, creatorAmount);
    //     ...
    // }
```

```
    creators[_creator].totalEarnings += creatorAmount;  // Stats updated but no transfe
}
```

Similarly in `_processPPVUnlock`:

```
// if (_token == address(0)) {
//     if (platformFee > 0) {
//         payable(platformWallet).transfer(platformFee);
//     }
//     payable(content.creator).transfer(creatorAmount);
// } else {
//     ...
// }
```

Impact:

- Users' BNB/tokens taken (via msg.value or transferFrom)
- Creators never receive payments
- Platform never receives fees
- Stats show earnings but funds stuck in contract

Recommendation:

Uncomment the transfer logic before deployment:

```
if (_paymentToken == address(0)) {
    if (platformFee > 0) {
        payable(platformWallet).transfer(platformFee);
    }
    payable(_creator).transfer(creatorAmount);
} else {
    IERC20 token = IERC20(_paymentToken);
    token.safeTransferFrom(msg.sender, _creator, creatorAmount);
    if (platformFee > 0) {
        token.safeTransferFrom(msg.sender, platformWallet, platformFee);
    }
}
```

[M-01] Mint function has no access control

Severity: Medium

Impact: High - Anyone can mint up to max supply.

Likelihood: Low - Max supply cap limits damage.

Location: `token.sol:28-35`

Description:

```
function mint(address to, uint256 amount) external {  // @audit No onlyOwner!
    require(to != address(0), "Cannot mint to zero address");
    require(totalSupply() + amount <= TOTAL_SUPPLY, "Exceeds maximum supply");
    _mint(to, amount);
}
```

The mint function has no access control. While the total supply is capped, anyone can mint tokens up to that cap.

Recommendation:

```
function mint(address to, uint256 amount) external onlyOwner {
    // ...
}
```

[M-02] Stakers array unbounded leading to DoS

Severity: Medium

Impact: Medium - Gas costs increase, potential transaction failures.

Likelihood: Medium - Grows with each new staker.

Location: `staking.sol:530-544`

Description:

```
function _removeStaker(address _staker) internal {
    // ...
    for (uint256 i = 0; i < stakers.length; i++) {  // @audit Unbounded loop
        if (stakers[i] == _staker) {
            stakers[i] = stakers[stakers.length - 1];
```

```
            stakers.pop();
            break;
        }
    }
}
```

With many stakers, unstaking becomes expensive or impossible.

Recommendation:

Use a mapping to track staker index for O(1) removal:

```
mapping(address => uint256) public stakerIndex;
```

[M-03] Emergency withdraw bypasses early withdrawal penalty

Severity: Medium

Impact: Medium - Users can avoid penalties during emergencies.

Likelihood: Low - Requires emergency situation.

Location: `staking.sol:592-612`

Description:

```
function emergencyWithdraw() external nonReentrant {
    // ...
    stakingToken.safeTransfer(msg.sender, amount);  // Full amount, no penalty
}
```

Users can use emergency withdraw to bypass the 10% early withdrawal penalty.

Recommendation:

Consider if this is intended behavior. If penalties should apply even in emergencies:

```
uint256 penalty = (amount * earlyWithdrawalPenalty) / BASIS_POINTS;
stakingToken.safeTransfer(msg.sender, amount - penalty);
```

[L-01] convertUSDToWei uses hardcoded BNB price

Severity: Low

Location: `main.sol:1036-1041`

Description:

```
function convertUSDToWei(uint256 _usdCents) public view returns (uint256) {
    if (bnbUsdPriceFeed == address(0)) {
        return (_usdCents * 1e18) / 60000;  // Hardcoded $600 BNB
    }
    return (_usdCents * 1e18) / 60000;  // Same hardcoded value even with feed!
}
```

Both branches return the same hardcoded calculation, ignoring the price feed.

Recommendation:

Implement actual Chainlink price feed integration:

```
function convertUSDToWei(uint256 _usdCents) public view returns (uint256) {
    if (bnbUsdPriceFeed == address(0)) {
        return (_usdCents * 1e18) / 60000;  // Fallback
    }
    (, int256 price,,,) = AggregatorV3Interface(bnbUsdPriceFeed).latestRoundData();
    return (_usdCents * 1e18) / uint256(price);
}
```

[L-02] Tier deletion loop can fail with many tiers

Severity: Low

Location: `main.sol:385-393`

Description:

Deleting a tier iterates through all active tiers array. With many tiers, this can become expensive.

[L-03] No minimum subscription period enforced

Severity: Low

Location: `main.sol:413`

Description:

```
require(_subscriptionPeriod > 0, "Invalid subscription period");
```

No minimum is enforced, allowing 1 second subscriptions.

Security Patterns Observed

Positive

- Solidity ^0.8.19 with overflow protection
- ReentrancyGuard on state-changing functions
- Pausable for emergency stops
- SafeERC20 for token transfers
- Role-based access control
- Early withdrawal penalty mechanism

Concerns

- Payment logic commented out (critical)
- Missing access control on mint
- Hardcoded price feeds
- Unbounded arrays

Conclusion

The DTreon Platform provides comprehensive creator economy features. However, the **payment processing logic is completely commented out**, which would result in funds being locked in the contract without transfers to creators or the platform.

The contract is NOT ready for deployment in its current state.

We STRONGLY RECOMMEND uncommenting the payment logic and conducting a follow-up review before any mainnet deployment.

Overall Risk Assessment: HIGH (due to commented payment logic)

This security review was conducted by Stonewall. For questions or clarifications, contact our team.