智能合约安全审计报告

1. 概要

慢雾安全团队于 2021 年 03 月 02 日，收到 SupremeX 团队对 SupremeX 系统安全审计的申请，根据项目特点慢雾安全团队制定如下审计方案。

慢雾安全团队将采用“白盒为主，黑灰为辅”的策略，以最贴近真实攻击的方式，对项目进行安全审计。

慢雾科技 DeFi 项目测试方法：

黑盒测试	站在外部从攻击者角度进行安全测试。
灰盒测试	通过脚本工具对代码模块进行安全测试，观察内部运行状态，挖掘弱点。
白盒测试	基于项目的源代码，进行脆弱性分析和漏洞挖掘。

慢雾科技 DeFi 漏洞风险等级：

严重漏洞	严重漏洞会对项目的安全造成重大影响，强烈建议修复严重漏洞。
高危漏洞	高危漏洞会影响项目的正常运行，强烈建议修复高危漏洞。
中危漏洞	中危漏洞会影响项目的运行，建议修复中危漏洞。
低危漏洞	低危漏洞可能在特定场景中会影响项目的业务操作，建议项目方自行评估和考虑这些问题是否需要修复。
弱点	理论上存在安全隐患，但工程上极难复现。
增强建议	编码或架构存在更好的实践方法。

2. 审计方法

慢雾安全团队智能合约安全审计流程包含两个步骤:

- 使用开源或内部自动化分析的工具对合约代码中常见的安全漏洞进行扫描和测试。
- 人工审计代码的安全问题，通过人工分析合约代码，发现代码中潜在的安全问题。

如下是合约代码审计过程中我们会重点审查的漏洞列表:

（其他未知安全漏洞不包含在本次审计责任范围）

- 重入攻击
- 重放攻击
- 重排攻击
- 短地址攻击
- 拒绝服务攻击
- 交易顺序依赖
- 条件竞争攻击
- 权限控制攻击
- 整数上溢/下溢攻击
- 时间戳依赖攻击
- Gas 使用，Gas 限制和循环
- 冗余的回调函数
- 不安全的接口使用
- 函数状态变量的显式可见性
- 逻辑缺陷
- 未声明的存储指针
- 算术精度误差
- tx.origin 身份验证
- 假充值漏洞
- 变量覆盖

3. 项目背景

3.1 项目介绍

SupremeX (SXC) 是一个在 OKExChain 上通过智能合约搭建的首个去中心化金融借贷平台。

SupremeX 为去中心化金融(DeFi)生态系统引入了简单易用的加密资产借贷解决方案，使用户能够直接以主流代币高速借贷，同时只需要较少的手续费。此外，SXC 允许用户在借贷过程中获取 SXC 代币，用于 SXC 社区的治理，可以用来投票，增加新的抵押品类型、改变合约参数和借贷协议改进。

审计合约文件：

项目源代码

审计初始版本：

Venus.zip: 29821b9e8a45257901a56231df7e70415942b4f5251300fd715d9bd13e623351

审计最终版本：

venus.zip: 159487a7ae02ec1d75012ec6fe63f9176b2ef7a279fc28b92ba5f679eb11059a

3.2 审计合约结构

```
├── BEP20Interface.sol
├── CarefulMath.sol
├── Comptroller.sol
├── ComptrollerG1.sol
├── ComptrollerG2.sol
├── ComptrollerInterface.sol
├── ComptrollerStorage.sol
├── Context.sol
├── DAIInterestRateModelV2.sol
├── EIP20Interface.sol
├── EIP20NonStandardInterface.sol
├── ERC20.sol
```

```
├── ErrorReporter.sol
├── Exponential.sol
├── Governance
│   ├── GovernorAlpha.sol
│   ├── SXP.sol
│   └── XVS.sol
├── InterestRateModel.sol
├── JumpRateModel.sol
├── Lens
│   └── VenusLens.sol
├── Maximillion.sol
├── Migrations.sol
├── MintableERC20.sol
├── Ownable.sol
├── PriceOracle.sol
├── PriceOracleProxy.sol
├── Reservoir.sol
├── SafeMath.sol
├── SimplePriceOracle.sol
├── Timelock.sol
├── Unitroller.sol
├── VAI
│   ├── VAI.sol
│   └── lib.sol
├── VAIController.sol
├── VAIControllerInterface.sol
├── VAIControllerStorage.sol
├── VAIUnitroller.sol
├── VBNB.sol
├── VBep20.sol
├── VBep20Delegate.sol
├── VBep20Delegator.sol
├── VBep20Immutable.sol
├── VDaiDelegate.sol
├── VToken.sol
├── VTokenInterfaces.sol
├── VenusPriceOracle.sol
└── WhitePaperInterestRateModel.sol
```

4. 代码概述

4.1 主要合约地址

合约暂未主网进行部署。

4.2 主要合约函数可见性分析

在审计过程中，慢雾安全团队对核心合约的可见性进行分析，结果如下:

Comptroller			
Function Name	Visibility	Mutability	Modifiers
getAssetsIn	External	-	-
checkMembership	External	-	-
enterMarkets	External	Can modify state	-
addToMarketInternal	Internal	Can modify state	-
exitMarket	External	Can modify state	-
mintAllowed	External	Can modify state	onlyProtocolAllowed
mintVerify	External	Can modify state	-
redeemAllowed	External	Can modify state	onlyProtocolAllowed
redeemAllowedInternal	Internal	-	-
redeemVerify	External	Can modify state	-
borrowAllowed	External	Can modify state	onlyProtocolAllowed
borrowVerify	External	Can modify state	-
repayBorrowAllowed	External	Can modify state	onlyProtocolAllowed
repayBorrowVerify	External	Can modify state	-
liquidateBorrowAllowed	External	Can modify state	onlyProtocolAllowed
liquidateBorrowVerify	External	Can modify state	-
seizeAllowed	External	Can modify state	onlyProtocolAllowed
seizeVerify	External	Can modify state	-
transferAllowed	External	Can modify state	onlyProtocolAllowed
transferVerify	External	Can modify state	-
getAccountLiquidity	Public	-	-

getHypotheticalAccountLiquidity	Public	-	-
getHypotheticalAccountLiquidityInternal	Internal	-	-
liquidateCalculateSeizeTokens	External	-	-
_setPriceOracle	Public	Can modify state	-
_setCloseFactor	External	Can modify state	-
_setCollateralFactor	External	Can modify state	-
_setMaxAssets	External	Can modify state	-
_setLiquidationIncentive	External	Can modify state	-
_supportMarket	External	Can modify state	-
_addMarketInternal	Internal	Can modify state	-
_setMarketBorrowCaps	External	Can modify state	-
_setBorrowCapGuardian	External	Can modify state	-
_setProtocolPaused	Public	Can modify state	onlyAdmin
_setVAIController	External	Can modify state	-
_setVAIMintRate	External	Can modify state	-
_become	Public	Can modify state	-
adminOrInitializing	Internal	-	-
setVenusSpeedInternal	Internal	Can modify state	-
updateVenusSupplyIndex	Internal	Can modify state	-
updateVenusBorrowIndex	Internal	Can modify state	-
distributeSupplierVenus	Internal	Can modify state	-
distributeBorrowerVenus	Internal	Can modify state	-
distributeVAIMinterVenus	Public	Can modify state	-
claimVenus	Public	Can modify state	-
claimVenus	Public	Can modify state	-
claimVenus	Public	Can modify state	-
grantXVSInternal	Internal	Can modify state	-
_setVenusVAIRate	Public	Can modify state	-
_setVenusVAIVaultRate	Public	Can modify state	-
_setVAIVaultInfo	Public	Can modify state	-
_setVenusSpeed	Public	Can modify state	-
getAllMarkets	Public	-	-
getBlockNumber	Public	-	-
getXVSAddress	Public	-	-
setXVSAddress	Public	Can modify state	onlyAdmin
setMintedVAIOf	External	Can modify state	onlyProtocolAllowed
releaseToVault	Public	Can modify state	-

VToken			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can modify state	-
transferTokens	Internal	Can modify state	-
transfer	External	Can modify state	nonReentrant
transferFrom	External	Can modify state	nonReentrant
approve	External	Can modify state	-
allowance	External	-	-
balanceOf	External	-	-
balanceOfUnderlying	External	Can modify state	-
getAccountSnapshot	External	-	-
getBlockNumber	Internal	-	-
borrowRatePerBlock	External	-	-
supplyRatePerBlock	External	-	-
totalBorrowsCurrent	External	Can modify state	nonReentrant
borrowBalanceCurrent	External	Can modify state	nonReentrant
borrowBalanceStored	Public	-	-
borrowBalanceStoredInternal	Internal	-	-
exchangeRateCurrent	Public	Can modify state	nonReentrant
exchangeRateStored	Public	-	-
exchangeRateStoredInternal	Internal	-	-
getCash	External	-	-
accrueInterest	Public	Can modify state	-
mintInternal	Internal	Can modify state	nonReentrant
mintFresh	Internal	Can modify state	-
redeemInternal	Internal	Can modify state	nonReentrant
redeemUnderlyingInternal	Internal	Can modify state	nonReentrant
redeemFresh	Internal	Can modify state	-
borrowInternal	Internal	Can modify state	nonReentrant
borrowFresh	Internal	Can modify state	-
repayBorrowInternal	Internal	Can modify state	nonReentrant
repayBorrowBehalfInternal	Internal	Can modify state	nonReentrant
repayBorrowFresh	Internal	Can modify state	-
liquidateBorrowInternal	Internal	Can modify state	nonReentrant
liquidateBorrowFresh	Internal	Can modify state	-
seize	External	Can modify state	nonReentrant
seizeInternal	Internal	Can modify state	-

_setPendingAdmin	External	Can modify state	-
_acceptAdmin	External	Can modify state	-
_setComptroller	Public	Can modify state	-
_setReserveFactor	External	Can modify state	nonReentrant
_setReserveFactorFresh	Internal	Can modify state	-
_addReservesInternal	Internal	Can modify state	nonReentrant
_addReservesFresh	Internal	Can modify state	-
_reduceReserves	External	Can modify state	nonReentrant
_reduceReservesFresh	Internal	Can modify state	-
_setInterestRateModel	Public	Can modify state	-
_setInterestRateModelFresh	Internal	Can modify state	-
getCashPrior	Internal	-	-
doTransferIn	Internal	Can modify state	-
doTransferOut	Internal	Can modify state	-

InterestRateModel			
Function Name	Visibility	Mutability	Modifiers
getBorrowRate	External	-	-
getSupplyRate	External	-	-

Unitroller			
Function Name	Visibility	Mutability	Modifiers
_setPendingImplementation	Public	Can modify state	-
_acceptImplementation	Public	Can modify state	-
_setPendingAdmin	Public	Can modify state	-
_acceptAdmin	Public	Can modify state	-
Fallback	External	Payable	-

XVS			
Function Name	Visibility	Mutability	Modifiers
allowance	External	-	-
approve	External	Can modify state	validLock
balanceOf	External	-	-
transfer	External	Can modify state	validLock
transferFrom	External	Can modify state	validLock

delegate	Public	Can modify state	validLock
delegateBySig	Public	Can modify state	validLock
getCurrentVotes	External	-	-
getPriorVotes	Public	-	-
_delegate	Internal	Can modify state	-
_transferTokens	Internal	Can modify state	-
_moveDelegates	Internal	Can modify state	-
_writeCheckpoint	Internal	Can modify state	-
safe32	Internal	-	-
safe96	Internal	-	-
add96	Internal	-	-
sub96	Internal	-	-
getChainId	Internal	-	-

VAI			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can modify state	note auth
deny	External	Can modify state	note auth
add	Internal	-	-
sub	Internal	-	-
transfer	External	Can modify state	-
transferFrom	Public	Can modify state	-
mint	External	Can modify state	auth
burn	External	Can modify state	-
approve	External	Can modify state	-
push	External	Can modify state	-
pull	External	Can modify state	-
move	External	Can modify state	-
permit	External	Can modify state	-

VBNB			
Function Name	Visibility	Mutability	Modifiers
mint	External	Payable	-
redeem	External	Can modify state	-
redeemUnderlying	External	Can modify state	-
borrow	External	Can modify state	-

repayBorrow	External	Payable	-
repayBorrowBehalf	External	Payable	-
liquidateBorrow	External	Payable	-
Fallback	External	Payable	-
getCashPrior	Internal	-	-
doTransferIn	Internal	Can modify state	-
doTransferOut	Internal	Can modify state	-
requireNoError	Internal	-	-

4.3 代码审计详情

4.3.1 中危漏洞

4.3.1.1 权限过大风险

在 VToken、VAIController、VenusPriceOracle 等合约中存在管理员角色，管理员可以对项目的利率模型、暂停机制等敏感参数进行任意修改.在 VAI 合约中，auth 角色可以通过 mint 函数进行无限制的铸造代币，且 auth 可通过 rely 函数为任意用户添加 auth 权限。这将导致管理员权限过大的风险。

修复建议：建议将管理员角色交与社区治理控制，以避免权限过大的风险。

修复状态：项目暂未在主网上线，暂未修复。

4.3.2 增强建议

4.3.2.1 签名重放问题

delegateBySig 函数 nonce 是由用户自己传入的参数进行签名的，当用户传了一个较大的 nonce 时，当前交易无法通过校验但是相关的签名数据仍会留在链上，导致此签名可能在未来某个时间段可用。(XVS.sol 中同样如此)

修复建议：建议参考 EIP-2612 进行修复。

代码位置： Governance/SXP.sol, VAI/VAI.sol

```
    function delegateBySig(address delegatee, uint256 nonce, uint256 expiry, uint8 v, bytes32 r, bytes32 s) public validLock permissionCheck {
        bytes32 domainSeparator = keccak256(abi.encode(DOMAIN_TYPEHASH, keccak256(bytes(_name)), getChainId(), address(this)));
        bytes32 structHash = keccak256(abi.encode(DELEGATION_TYPEHASH, delegatee, nonce, expiry));
        bytes32 digest = keccak256(abi.encodePacked("\x19\x01", domainSeparator, structHash));
        address signatory = ecrecover(digest, v, r, s);
        require(signatory != address(0), "Invalid signature");
        require(nonce == nonces[signatory]++, "Invalid nonce");
        require(now <= expiry, "The signature expired");
        return _delegate(signatory, delegatee);
    }
```

```
    function permit(address holder, address spender, uint256 nonce, uint256 expiry,
                    bool allowed, uint8 v, bytes32 r, bytes32 s) external
    {
        bytes32 digest = keccak256(abi.encodePacked(
                "\x19\x01",
                DOMAIN_SEPARATOR,
                keccak256(abi.encode(PERMIT_TYPEHASH,
                                     holder,
                                     spender,
                                     nonce,
                                     expiry,
                                     allowed))
        ));

        require(holder != address(0), "VAI/invalid-address-0");
        require(holder == ecrecover(digest, v, r, s), "VAI/invalid-permit");
        require(expiry == 0 || now <= expiry, "VAI/permit-expired");
        require(nonce == nonces[holder]++, "VAI/invalid-nonce");
        uint wad = allowed ? uint(-1) : 0;
        allowance[holder][spender] = wad;
        emit Approval(holder, spender, wad);
    }
```

修复状态： 未修复。

5. 审计结果

5.1 总结

审计结论：存在风险

审计编号：0X002103230002

审计时间：2021 年 03 月 23 日

审计团队：慢雾安全团队

审计总结：慢雾安全团队采用人工结合内部工具对代码进行分析。审计期间发现了 2 个问题。其中包含 1 个中危漏洞，并提出了 1 点增强建议。由于目前项目暂未在主网部署，权限仍未移交社区治理，因此项目仍存在权限过大的风险。

6. 声明

慢雾仅就本报告出具前已经发生或存在的事实出具本报告，并就此承担相应责任。对于出具以后发生或存在的事实，慢雾无法判断其智能合约安全状况，亦不对此承担责任。本报告所作的安全审计分析及其他内容，仅基于信息提供者截至本报告出具时向慢雾提供的文件和资料(简称“已提供资料”)。慢雾假设：已提供资料不存在缺失、被篡改、删减或隐瞒的情形。如已提供资料信息缺失、被篡改、删减、隐瞒或反映的情况与实际情况不符的，慢雾对由此而导致的损失和不利影响不承担任何责任。

慢雾科技
slow mist

官方网址

www.slowmist.com

电子邮箱

team@slowmist.com

微信公众号