SharkTeam

Move Language Security Analysis and Contract Audit — Manipulate the oracle

Jan 30, 2023

SharkTeam, a leading blockchain security service team, offers smart contract audit services for developers. To satisfy the demands of different clients, thesmart contract audit services provide both manual auditing and automated auditing.

We implement almost 200 auditing contents that cover four aspects: high-level language layer, virtual machine layer, blockchain layer, and business logiclayer, ensuring that smart contracts are completely guaranteed and Safe.

In the previous “Top 10 Smart Contracts Security Threats” series, SharkTeam summarized and analyzed the top 10 vulnerabilities in the smart contract space based on historical smart contract security incidents. These vulnerabilities were usually found in Solidity smart contracts before, so will they be the same for Move smart contracts?

The SharkTeam [Move Language Security Analysis and Contract Audit Essentials] course series will take you step-by-step into the content, including permission vulnerabilities, re-entry vulnerabilities, logical checksum vulnerabilities, function malicious initialization, fallback attacks, proposal attacks, contract escalation vulnerabilities, manipulation of the prophecy machine, sandwich attacks, and replay attacks. The content of this chapter [Manipulate the oracle].

Manipulating oracles is a common attack method in the DeFi field. For the price oracle on the chain, it is easier to be manipulated, especially in a transaction, a large amount of attack funds can be provided through flash loans, so as to realize the manipulation of the price oracle.

1. What is a price oracle

The oracle (Oracle) is a bridge linking smart contracts on the chain and the real environment off the chain. The price oracle machine is a bridge tool for smart contracts on the chain to obtain the price of tokens on or off the chain.

According to the source of price data, there are two types of price oracles: on-chain price oracles and off-chain price oracles.

l On-chain price oracles: Price data comes from on-chain, such as Uniswap trading pairs (AMM) prices

l Off-chain price oracles: Price data comes from off-chain, such as Chainlink

l According to the operators who provide price data, price oracles can be divided into two types: centralized price oracles and decentralized price oracles.

l Centralized price oracle: The price oracle can only be updated and verified by

a single operator, that is, the price of the feedback comes from a single source.

l Decentralized price oracle: The price oracle is updated and verified by multiple operators or users, that is, the price feedback comes from different sources and can be calculated from multiple prices.

2. Manipulating price oracles

Among all the security incidents of blockchain projects, the security incidents caused by manipulating price oracles are the most common one, especially for on-chain price oracles. In addition, flash loans can provide a large amount of funds for manipulating price oracles at extremely low fees, which provides great convenience for operating price oracle attacks.

There are two ways to manipulate price oracles:

(1) Flash loans manipulate the AMM price on the chain: first, the attacker adjusts the price of borrowed tokens upward (or downwardly adjusts the collateral price), and finally liquidates the collateral through the price. This situation is generally in the centralized oracle machine on the chain and the decentralized exchange. The attacker can use the flash loan to manipulate the price of AMM in a transaction, and change the spot price of the token before the lender's smart contract finds the token.

(2) Random attacks caused by oracle machine failure: On June 25, 2019, the price oracle machine relied on by the derivatives platform Synthetix experienced a price feeding failure of the oracle machine, and robots quickly entered and exited the sKRW market, eventually causing a large amount of economic losses.

Of the two approaches, the more frequent of the security incidents is the first, such as the xToken security incident that occurred in May 2021 and the Belt Finance security incident.

3.xToken Security Events

DeFi staking and liquidity strategy platform xToken was attacked by flash loans and price manipulation oracles on May 13, 2021. The xBNTa Bancor pool and

xSNXa Balancer pool were immediately exhausted, causing a loss of about 25 million US dollars. Among them, the xSNXa contract The total value of direct losses above is 416 ETH.

After analyzing the attack transactions, the attack steps are as follows.

(1) The attacker first borrowed a large amount of ETH from dYdX via Lightning Credit, then pledged ETH in AAVE and borrowed SNX. and exchanged ETH for SNX via Sushiswap. thus, the attacker stockpiled a large amount of SNX

(2) All the SNX will be exchanged to ETH through Uniswap, and the price of SNX in the ETH/SNX pair of Uniswap will become very low at this time.

(3) The attacker uses ETH to replace xSNX through the xSNX contract. Since xSNX uses the SNX price on Kyber, and the SNX price on Kyber is referenced to the SNX price of the ETH/SNX pair on Uniswap, the attacker replaces more xSNX at this point.

(4) After replacing the xSNX in the account with ETH and SNX, the attacker returns it to Lightning Loan. The remaining tokens are the profit of the attack.

The attacker manipulates (lowers) the price of SNX through Lightning Lending because the xSNX in the attacked contract uses the SNX price on Kyber, which is referenced to the Uniswap's prophecy machine price. When the SNX price is affected it directly leads to abnormal fluctuation of xSNX price in xToken, causing an attack arbitrage space.

4. Belt Finance Security Incident

The multi-strategy revenue optimization AMM protocol Belt Finance on the Coin Smart Chain (BSC) was attacked by a lightning loan on May 30, 2021, and four pools were affected with a loss of $6.2 million.

After analyzing the attack transactions, the attack process is as follows.

(1) The attacker lent a total of 387,315,994 BUSD (387 million BUSD) from PancakeSwap via lightning loans, in 8 instances.

From 0x58f876857a02d...	To attack contract	For 107,736,995.181428080466763522 ($107,706,823.66)	Binance-Peg ... (BUSD)
From 0x2354ef4df11afa...	To attack contract	For 38,227,899.20026643768020011 ($38,217,193.56)	Binance-Peg ... (BUSD)
From 0x7efaef62fddcca...	To attack contract	For 153,621,552.664648194085106148 ($153,578,531.27)	Binance-Peg ... (BUSD)
From 0x66fdb2eccfb58cf...	To attack contract	For 31,372,406.80087301838491625 ($31,363,621.02)	Binance-Peg ... (BUSD)
From 0x05faf555522fa3f...	To attack contract	For 17,505,135.133349282372649578 ($17,500,232.86)	Binance-Peg ... (BUSD)
From 0x133ee93fe9332...	To attack contract	For 17,294,888.215168936840783967 ($17,290,044.82)	Binance-Peg ... (BUSD)
From 0x7752e1fa9f3a2e...	To attack contract	For 10,828,766.507628240277418295 ($10,825,733.93)	Binance-Peg ... (BUSD)
From 0x804678fa97d91...	To attack contract	For 10,728,353.170906869924155636 ($10,725,348.72)	Binance-Peg ... (BUSD)

（2）Deposit 10 million of these BUSD into the bEllipsisBUSD strategy, Ellipsis is a project licensed to Fork by Curve, the DeFi protocol on Ether.

From attack contract	To MultiStrategy	For 10,000,000 ($9,996,112.35)	Binance-Peg ... (BUSD)
From MultiStrategy	To SingleStrategy	For 10,000,000 ($9,996,112.35)	Binance-Peg ... (BUSD)
From SingleStrategy	To Ellipsis	For 10,000,000 ($9,996,112.35)	Binance-Peg ... (BUSD)
From Ellipsis	To Curve.Fi	For 10,000,000 ($9,996,112.35)	Binance-Peg ... (BUSD)
From 0x0000000000000...	To Ellipsis	For 9,979,523.760129058636855084	Ellipsis.fin... (3EPS)
From 0x0000000000000...	To MultiFeeDistribution	For 40.899668046145336744 ($25.90)	Ellipsis (EPS)
From Ellipsis	To 0xcce949de564fe...	For 9,979,523.760129058636855084	Ellipsis.fin... (3EPS)
From 0x0000000000000...	To MultiStrategy	For 9,827,429.928988095396623737	bEllipsisBUS... (bEllip...)
From 0x0000000000000...	To attack contract	For 9,815,290.724622667074730851	beltBUSD (beltBU...)

(3) Deposit 187 million BUSD into the bVenusBUSD strategy.

From attack contract	To MultiStrategy	For 187,315,996.874269060031993506 ($187,243,175.05)	Binance-Peg ... (BUSD)
From 0x0000000000000...	To MultiStrategy	For 183,330,311.027564342527408894	bVenusBUSD (bVenus...)
From MultiStrategy	To 0x53a53a9e10abe...	For 187,315,996.874269060031993506 ($187,243,175.05)	Binance-Peg ... (BUSD)
From 0x0000000000000...	To attack contract	For 183,917,614.828794108117986198	beltBUSD (beltBU...)

(4) Converting 190 million BUSD to 169 million USDT through the Ellipsis contract.

From attack contract	To Curve.Fi	For 190,000,000 ($189,926,134.73)	Binance-Peg ... (BUSD)
From Curve.Fi	To attack contract	For 169,949,807.019748917627458054 ($169,967,018.18)	Binance-Peg ... (BUSD-T)

At this point, all 387 million BUSD from the Lightning Credit has been put into attack transactions. 187 million of them were deposited into the bVenusBUSD strategy, 10 million into the bEllipsisBUSD strategy, and 190 million were converted into 169 million USDT through Ellipsis.

(5) BUSD withdrawals from the bVenusBUSD strategy.

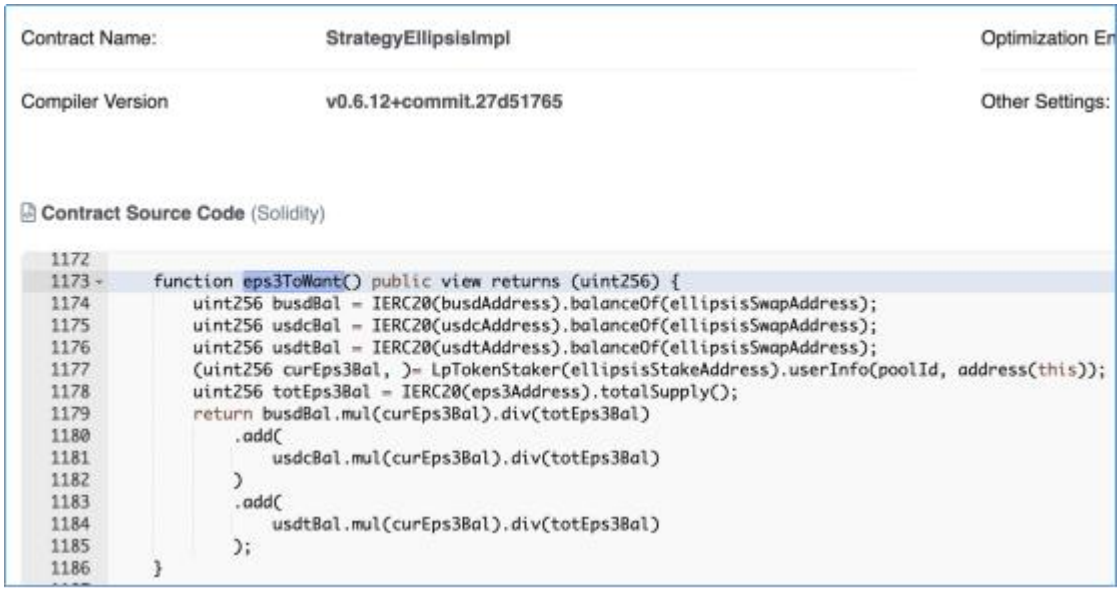
(6) Exchange of 169 million USDT for 189 million BUSD through Ellipsis.

(7) Deposit another 197 million BUSD into the bVenusBUSD strategy.

Repeating steps (5) to (7) several times, the attacker finally extracted more BUSD from the strategy bVenusBUSD by repeatedly buying and selling BUSD, using 169 million BUSDT to exchange 189 million BUSD.

(8) Return the lightning credit and transfer the remaining assets for profit.

To summarize the above steps, this attack, due to a vulnerability in the bEllipsis strategy balance calculation, caused the attacker to manipulate the price of beltBUSD. The analysis of the contract shows that the price of beltBUSD depends on the balance in the machine gun pool. The attacker deposited the BUSD into the bVenusBUSD strategy and then raised it again, and since the number of assets remains the same, even if the operation is repeated, no profit will be made. However, this time the attacker also operated other strategies (bEllipsisBUSD) at the same time to have an impact on the beltBUSD balance, causing the creation of a computational vulnerability.



Contract Name: StrategyEllipsisImpl Optimization Er

Compiler Version v0.6.12+commit.27d51765 Other Settings:

Contract Source Code (Solidity)

```
1172
1173 -    function eps3ToWant() public view returns (uint256) {
1174         uint256 busdBal = IERC20(busdAddress).balanceOf(ellipsisSwapAddress);
1175         uint256 usdcBal = IERC20(usdcAddress).balanceOf(ellipsisSwapAddress);
1176         uint256 usdtBal = IERC20(usdtAddress).balanceOf(ellipsisSwapAddress);
1177         (uint256 curEps3Bal, ) = LpTokenStaker(ellipsisStakeAddress).userInfo(poolId, address(this));
1178         uint256 totEps3Bal = IERC20(eps3Address).totalSupply();
1179         return busdBal.mul(curEps3Bal).div(totEps3Bal)
1180             .add(
1181                 usdcBal.mul(curEps3Bal).div(totEps3Bal)
1182             )
1183             .add(
1184                 usdtBal.mul(curEps3Bal).div(totEps3Bal)
1185             );
1186     }
```

5. Price Oracle in Move

The oracle machine is an essential infrastructure for the public chain. The public chains of the Move ecosystem, including Starcoin, Aptos, and SUI, also need an oracle machine. Therefore, the Move public chain defines a standard Oracle protocol to provide more convenient oracle services for the public chain.

Starcoin defines a set of standard oracle protocols in Stdlib. The standard Oracle protocol is written using the safe and reliable smart contract Move, which not only maintains the decentralization of Starcoin, but also inherits the security of Move.

In the Move ecosystem, in addition to the AMM on the chain that can be used as a price oracle, there is also an off-chain price oracle infrastructure similar to Chainlink in the Solidity ecosystem, which provides certain guarantees for the security of DeFi projects, including providing prices for Aptos Pyth for oracle service, SupraOracles that can provide price oracle service for Aptos and SUI, etc.

6. Move contract manipulates price oracle risk

Manipulating price oracles has nothing to do with smart contract languages and blockchain structures, but only with economic models and types of oracles,

which is a type of economic attack. Therefore, the Move contract may also be at risk of manipulating the price oracle.

Multiple security incidents involving the operation of price oracles in the Solidity ecosystem tell us that flash loans are the most effective auxiliary means for manipulating price oracles. However, in the Move ecosystem, the implementation of flash loans is not as convenient as Solidity, which can reduce the risk of manipulating price oracles to a certain extent, but it cannot fundamentally solve the problem of manipulating price oracles. Moreover, flash loan is a typical characteristic service of DeFi, and it will definitely appear in various Move public chain ecosystems in the future. The real way to avoid price manipulation is to use off-chain oracles similar to the Solidity ecological Chainlink, such as Pyth and SupraOracles.

The experience of the Solidity ecosystem tells us that manipulating price oracles has become a very common security problem in the DeFi field. We believe that the Move project team will take proactive measures during the development process to avoid the attack of manipulating price oracles.

About Us

Our vision is to improve security globally. We believe that by building this security barrier, we can significantly improve lives around the world.SharkTeam composes of members with many years of cyber security experiences and blockchain, team members are based in Suzhou, Beijing, Nanjing and Silicon Valley, proficient in the underlying theories of blockchain and smart contracts, and we provide comprehensive services including threat modeling, smart contract auditing, emergency response, etc. SharkTeam has established strategic and long-term cooperations with key players in many areas of the blockchain ecosystem, such as Huobi Global, OKX, polygon, Polkadot, imToken, ChainIDE, etc

SharkTeam

In Math, We Trust!

https://sharkteam.org

https://t.me/sharkteamorg

https://twitter.com/sharkteamorg