Nowswap Spot DEX Incident Post-Mortem

Summary

- On 09/15/21, Nowswap's Spot protocol was exploited and about $1M worth of TVL was drained. The cause was a bug in the pair contract that did not catch an invalid K value. The team started working on the incident and related follow-ups.

Timeline

- <Sep-14-2021 03:16:09 PM +UTC> Hacker created the malicious address and transferred 0.14723529 ETH from 0xf66852bc122fd40bfecc63cd48217e88bda12109 which is tagged as Huobi37.
- <Sep-15-2021 07:42:26 AM +UTC> Hacker's contract for Nowswap was created.
- <Sep-15-2021 07:43:11 AM +UTC> The hacker made the attack through the transaction (etherscan.io/tx/0xf3158a) and swapped out ~158 WETH and ~ 535,706 USDT from the Nowswap liquidity pool to the Hacker's address. The transaction was included in block 13229001.
- <Sep-15-2021 07:47:19 AM +UTC> Attacker swapped all USDT into ETH through 1inch in this transaction.
- <Sep-15-2021 07:51:28 AM +UTC> Attacker deposited ETH to Tornado.Cash in the following 4 transactions:
  https://etherscan.io/tx/0x5d424eab
  https://etherscan.io/tx/0xee432610
  https://etherscan.io/tx/0x3028de2b
  https://etherscan.io/tx/0x3db8eadd

Impact

The incident caused the Nowswap USDT/ETH pool to lose ~1M TVL. This affected 5 liquidity providers.

Detection

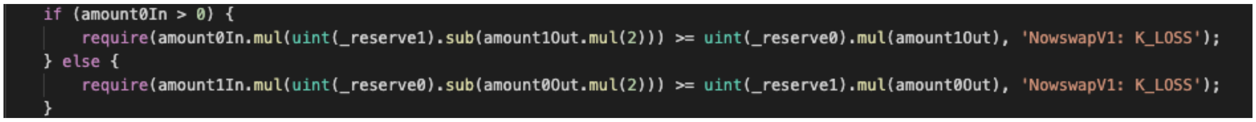
The incident was first identified through @PuPuThrashing's tweet. The team also noticed this incident from the sudden drop in TVL shown in Nowswap's Info Site.

Response

After discovering this attack, the team took immediate action. We first informed major stakeholders and the DeFi community about what happened. Then, we investigated the root cause of the incident with the help of smart contract security experts. The team compared the attack on Nimbus with the attack on Nowswap and found no correlation between the root causes. In addition, the team pieced together the on-chain and off-chain information to identify the hacker.

Root Cause

The root cause of this incident was not due to missing 0 because Nowswap does not have any constant value in the K check formula.
Here is the screenshot of the Nowswap spot check:



```
if (amount0In > 0) {
    require(amount0In.mul(uint(_reserve1).sub(amount1Out.mul(2))) >= uint(_reserve0).mul(amount1Out), 'NowswapV1: K_LOSS');
} else {
    require(amount1In.mul(uint(_reserve0).sub(amount0Out.mul(2))) >= uint(_reserve1).mul(amount0Out), 'NowswapV1: K_LOSS');
}
```

In fact, the root cause was an invalid check on an edge case in the swap function. Normally when a swap happens, (use input Token0 and output Token1 as an example) the *amount0In* and *amount1Out* are both positive, while the *amount0Out* and *amount1In* are both zero.

However, in the attack transaction, the hacker attempted to swap token0 for token0. Therefore, *amount0In* and *amount0Out* were both positive while the *amount1In* and *amount1Out* were both zero. Because the amountIn and amountOut of Token1 was zero, part of the K value check formula also became 0, which then bypassed all the subsequent safety checks. As a consequence, the attacker was able to swap an arbitrary amount of token out.

Lessons Learned

The Nowswap team has taken a painful lesson here. Moving forward, we are going to focus more on improving our test coverage to account for more edge cases. Although the protocol was running for 6 months and the protocol had gone through 2 audits for the current version, we know this still does not guarantee safety. As for further development of Nowswap, the developers will open source early so that we can mitigate future risk. Additionally, we will provide complete documentation along with bug bounties for any white hats.

Corrective Actions

The team has decided to take the Nowswap app site into maintenance to fully check the vulnerability of all features. The team is happy to provide bug bounties for developers providing help. We will continue to work with law enforcement to get the funds back.