Wormhole Attack Analysis

Nikola Valesova

September 2, 2022

1 Introduction

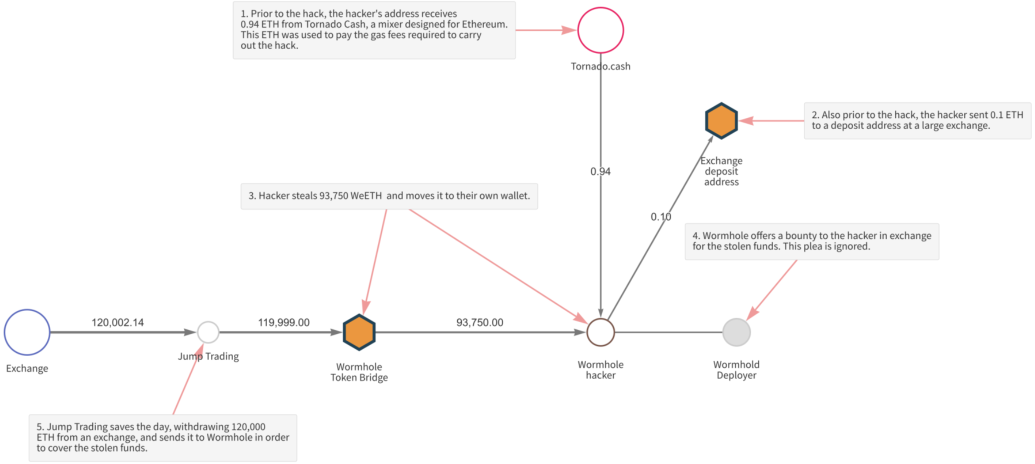
A hack of the Wormhole Network, a popular cross-chain protocol, took place on 2 February 2022 when an unknown hacker exploited a vulnerability in the Wormhole Network. The attack resulted in one of the five largest crypto hacks of all time after the attacker successfully made close to $325 million. [1]

The attacker made use of a security flaw that had been fixed in the latest commit of the project's GitHub repository, the new changes, however, had not yet been deployed to the project itself and thus revealed a possibility to bypass the security measures. [2]

To carry out the attack, the attacker managed to forge a valid signature for a transaction that allowed them to freely mint 120,000 WETH — a "wrapped" Ethereum equivalent on the Solana blockchain, with value equivalent to $325 million at the time of the theft — without first inputting an equivalent amount. This was then exchanged for around $250 million in Ethereum that was sent from Wormhole to the hackers' account, effectively liquidating a large amount of the platform's Ethereum funds that were being held as collateral for transactions on the Solana blockchain. [2]

2 Attack overview

From a high-level perspective, the attack and its preceding and succeeding actions can be described by five transactions. A diagram of the parties and transactions involved can be seen on the Chainalysis Reactor graph 1. The content of this section has been written on the basis of information in an article by ChainAnalysis [1].



1. Prior to the hack, the hacker's address receives 0.94 ETH from Tornado Cash, a mixer designed for Ethereum. This ETH was used to pay the gas fees required to carry out the hack.

2. Also prior to the hack, the hacker sent 0.1 ETH to a deposit address at a large exchange.

3. Hacker steals 93,750 WeETH and moves it to their own wallet.

4. Wormhole offers a bounty to the hacker in exchange for the stolen funds. This plea is ignored.

5. Jump Trading saves the day, withdrawing 120,000 ETH from an exchange, and sends it to Wormhole in order to cover the stolen funds.

Figure 1: Chainalysis Reactor graph of the Wormhole hack

1. Hacker's address receives 0.94 ETH from Tornado Cash

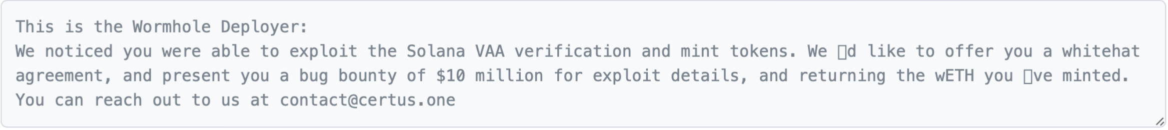
   The first transaction happens in preparation for the attack. The hacker uses Tornado Cash, an Ethereum-based mixer, to receive ETH, which can be later used to pay gas fees for the transactions during and immediately after the attack.

2. Hacker sends 0.1 ETH to a deposit address at a large exchange

3. Hacker steals 93,750 WETH and transfers it into their wallet

4. Wormhole sends an offer to the hacker to give them a bounty of $10 million in exchange for the stolen assets, which remains unaccepted

   The message with the bounty offer was sent from Wormhole to the attacker's wallet address inside a 0 ETH transaction. The transaction can be inspected on Etherscan and the text of the message can be found under input data 2. This attempt to pay the hacker a bounty in return for the stolen funds was ignored.



```
This is the Wormhole Deployer:
We noticed you were able to exploit the Solana VAA verification and mint tokens. We □d like to offer you a whitehat
agreement, and present you a bug bounty of $10 million for exploit details, and returning the wETH you □ve minted.
You can reach out to us at contact@certus.one
```

Figure 2: Chainalysis Reactor graph of the Wormhole hack [3]

5. Jump Trading withdraws 120,000 ETH from an exchange and sends it to Wormhole to cover the stolen funds

   The hack led to a state where $320 million worth of WETH on Solana was unbacked. This imbalance could have resulted in a critical situation as several Solana-based platforms, which accept WETH as collateral, could become insolvent, and users would quickly sell their WETH, both factors accelerating the crash of the value of WETH. The series of events could have resulted in a serious issue for the Solana blockchain and the ecosystem on top of it.

   Luckily, Jump Trading, Wormhole's parent company and a major player in the Solana ecosystem, withdrew and supplied Ether to replace what was stolen. In the end, there was a 13.5% dip in Solana's price shortly after the hack, which may be attribute to concerns around Solana's safety and stability, however, the future of the blockchain and platforms built on top of it remained more or less intact.

3 Vulnerability Details

In this section, we will dive deeper into the vulnerability and how exactly the attacker managed to exploit it.

This part has been written based on information in a series of Tweets by samczsun [4].

1. The attacker created a Solana account 2tHS1...rrZRd, which contained a single serialized instruction corresponding to a call to the Secp256k1 contract.

2. The attacker generated a `SignatureSet` by calling `verify_signatures` and passing in the account created in the previous step as the `Instruction sysvar`.

   Here lies the core vulnerability that was exploited. The `verify_signatures` function is expected to take a set of signatures provided by the guardians and pack it into a `SignatureSet`. However, it doesn't actually do any verification itself and delegates it to the Secp256k1 program instead, which is the root of the issue. The `solana_program::sysvar::instructions` mod should be used with the `Instruction sysvar`. However, the version of `solana_program` that Wormhole used didn't verify the address passed as a parameter. This meant that anyone could create their own account that would hold the same data as the `Instruction sysvar` and substitute that account for the `Instruction sysvar` in the `verify_signatures` execution. This would essentially bypass signature validation entirely. This way the hacker managed to fully bypass signature validation. The corresponding transaction has the ID 25Zu1...o4wVS.

3. Next, the hacker executed transaction with ID 2Soho...n1BCK, in which the `post_vaa` function is called on the main Wormhole bridge. To bypass the signature checks done by this function, the `SignatureSet` from the previous step was used. This way the hacker was able to generate an acceptable VAA (validator action approval).

4. In the last step of the hack, the attacker executed the `complete_wrapped` function with the previously obtained VAA. As a result, the attacker successfully triggered an unauthorized mint and their Solana wallet address received freshly minted 120,000 ETH in transaction 2zCz2...cD6Es.

5. After the hack has been completed, the attacker managed to bridge out 10,000 ETH to their Ethereum wallet address, which can be seen in the transaction with ID 5UaqP...DcAuf on the Solana blockchain and transaction with ID 0x4d520...4a80f on the Ethereum blockchain.

4 Conclusion

We have described and examined the hack of the Wormhole protocol and why it could have had serious consequences. The main lessons we can learn from this attack in order to prevent others in the future are to deploy any significant changes directly after pushing the code to the GitHub repository and to always verify all addresses passed as input.

As of today, the hacker hasn't been identified. Their Ethereum wallet address (0x629e7Da20197a5429d30da36E77d06CdF796b71A) still holds 93,750 ETH and their Solana wallet address (CxegPrfn2ge5dNiQberUrQJkHCcimeR4VXkeawcFBBka) holds slightly over 432,000 SOL.

References

[1] C. Team, "Wormhole hack: Lessons from the wormhole exploit," February 2022. [Online; cited on 1 September 2022].

[2] C. Faife, "Wormhole cryptocurrency platform hacked for $325 million after error on github - the verge," February 2022. [Online; cited on 1 September 2022].

[3] C. Team, "Graph_copy_of_wormhole_hack_2022_02_02_v241024x5332x," 2022. [Online; accessed August 27, 2022].

[4] samczsun, "How did the @wormholecrypto exploit work? i joined forces with @gf_256 and @ret2jazzy to reverse engineer the exploit, and now that it's been patched we can finally share it with you," February 2022. [Online; cited on 2 September 2022]. Tweet.