Backdoor Message Execution: Trusted Bridge can verify and execute unsent message on destination chain.

Submitted about 10 hours ago by @abhi3700 (Whitehat) for LayerZero (The World's Largest Bounty)

Details

Report ID

30670

Target

`https://etherscan.io/address/0x1a44076050125825900e736c501f859c50fe728c#code`

Smart Contract

Impact(s)

- All above impacts for OApp, OFT & ONFT related contracts
- Exploits resulting in the permanent locking or theft of user funds
- Permanent DoS attacks (excluding volumetric attacks)
- Any governance voting result manipulation
- Griefing (e.g. no profit motive for an attacker, but damage to the users or the protocol)
- Backdoor Message Execution: Executes unsent message by a trusted DVN/Executor (Bridge) on the destination chain

Status

Closed

The bug report is invalid and will not be paid.

Severity

Critical

Participants

Whitehat

Request Help

LayerZero (The World's Largest Bounty)

Wallet Addresses

EVM Primary

0x30209DA217CDCcBB21FF3c8702Fbd9881dBb6EdE

Description

Brief/Intro

A trusted bridge between 2 OApp/OFT/ONFT contracts (on 2 different chains) could verify & execute an encoded message by just increasing the nonce, without it actually been sent from the source chain, given the OApp chose the set of malicious DVNs (1 or more). This can lead to unlimited minting of tokens or send fake messages. For instance, Alice (from Nova) didn't send wTSSC to Bob (on Sepolia), but the receiver (Bob) received because of Bridge's verification and execution.

Vulnerability Details

I have 2 token contracts deployed on 2 EVM chains (Subspace Nova and ETH Sepolia testnet). Also, deployed the LayerZero suite of contracts, libraries to support Subspace Nova & Sepolia testnet. And then I created a Bridge (written in TypeScript) with 1 DVN security stack configuration for my OFT as PoC. Now, what I found during sending tokens to & fro that, I could send unlimited tokens to my destination chain without it actually been sent from source chain & viceversa. All the steps are added in PoC section below.

Impact Details

- As of now with the scattered/unorganized ecosystem of public DVNs, any OApp/OFT/ONFT contract owner could select a set of malicious DVNs pretending to be a genuine network (by showing the messages proper verification and execution). And then one of the DVNs become executor to exploit by executing verified message (by their pretentious DVNs) with minting as many tokens as they could.
- The encoded message could be decoded and then change the receiver address to preferred one by the malicious executor. And then they could just mint unlimited amount of tokens And then it would compromise so much so that the next message with legit nonce would not be verified as well as executed. Hence the token value/economics is crashed.

- Imagine like in my case, it was a protocol's own token which was deposited into a Wrapped token (that inherits from OFT). So, in this case, the whole Subspace protocol's token is compromised due to LZ's unorganized set of DVNs roaming out there. Out of which some could be malicious ones showing that they are processing packets properly to & fro.
- The impact is not just for fungible & non-fungible tokens. But, also could be for generating info-based messages on destination chain that are actually fake.

References

- [Demo Video that shows the vulnerability live](https://drive.google.com/file/d/1Gq1WrVbY9DQ8ZF-kVlTb8UiAVZsN1ZSA/view?usp=sharing)

Old videos to get more context:

- [Send tokens from Nova to Sepolia](https://drive.google.com/file/d/1UAwgiQ-LQ8dFonEVT6ufAfBjl-2GZlyC/view?usp=sharing).
- [Send tokens from Sepolia to Nova](https://drive.google.com/file/d/1Cy1wunxCSMgXRjmpthHmN1B3pdHyOO9o/view?usp=sharing). [Add-on](https://drive.google.com/file/d/1McCOSttT-CV4XWZiLpAGZmUNOJ9ArIPS/view?usp=sharing)

Github issue (OPEN) in "LayerZero-v2" repo:

- https://github.com/LayerZero-Labs/LayerZero-v2/issues/72

Code:

- [Bridge script (in TypeScript)](https://github.com/subspace/layerzero-playground/blob/234a81f53bd95e733ef0b6fc36df8b7a5a3bf653/demos/src/auto-bridge/dvn.ts): Run the Bridge script to get the latest encoded message packet. Keep it alive to listen to emitted events from either chains (source or destination).
- [Send tokens from one EVM chain to another (in TypeScript)](https://github.com/subspace/layerzero-playground/blob/234a81f53bd95e733ef0b6fc36df8b7a5a3bf653/demos/src/auto-bridge/index.ts): Send 0.01 TSSC (or more) tokens from Nova to Sepolia or viceversa.
- [Partial Bridge script (in Solidity)](https://github.com/subspace/subspace-evm-contracts/blob/00bbdf1dd963719bc60f951ae9199c2358714a2d/script/lz/AutoBridgeDVN.s.sol): Run this with the modified (increase nonce by 1, also could optionally change the receiver address to hacker's) encoded message.

Proof of Concept

Proof of Concept

I followed the steps below (also helpful for you to reproduce):

1. Run Bridge (in TS) Script and get the encoded packet when a message is sent. Let this message gets verified and executed properly through the bridge. The bridge currently has both the verification & execution authority.
2. Now, copy the encoded Packet from terminal (where Bridge is running) E.g. `0x01000000000000001f00077a10000000000000000000000000a66782c958e08275566463cb76a7892e72f2edb100009ce10000000000000000000000008ecc60d2a42747742b9fc67fb25de774677e260e94184cafcf3d40eabcaf7bd82b54c6a1c9c5e1bd45d3d491049d55348e8d0b71000000000000000000000000b751710af8ce68677ab960adb103060f38d097140000000000002710`
3. Paste into the partial Bridge script (written in Solidity) modifying just the nonce of the encoded message by increasing from 0x1f to 0x20 to mint 0.01 WTSSC to receiver. Although optional, you could change the receiver address with that of hacker. E.g. `0x010000000000000002000077a10000000000000000000000000a66782c958e08275566463cb76a7892e72f2edb100009ce10000000000000000000000008ecc60d2a42747742b9fc67fb25de774677e260e94184cafcf3d40eabcaf7bd82b54c6a1c9c5e1bd45d3d491049d55348e8d0b71000000000000000000000000b751710af8ce68677ab960adb103060f38d097140000000000002710`.
4. Run the partial bridge script (in Solidity) to follow the entire steps: DVN verification, Commit Verification, EndpointV2's LzExecute function. And it runs successfully.
5. And then when I actually sent 0.01 WTSSC from source chain as the next message with the next nonce. The message was already executed. I attempted this twice. And it came out to be successful! Strange!

In this PoC, there is one address that is the deployer/delegate of the LZ Suite & also WTsscLz contracts. Although the vulnerability still remains even if there are different admins (of bridge & OFT). Just need this to happen: "Any malicious bridge pretending to be a genuine (showcasing different packets sending from multiple sender contracts) , if somehow could get a potential token contract (with high price/economic value) get their DVNs added into their OApp/OFT/ONFT's DVN Security stack, the project suffers potentially billions of dollars of losses by letting them mint unlimited tokens/messages (could be voting)."

Watch the demo video that shows the whole process of hacking it twice. Link: https://drive.google.com/file/d/1Gq1WrVbY9DQ8ZF-kVlTb8UiAVZsN1ZSA/view?usp=sharing

Also, shared some old video links for starters.

See less ^

Timeline

abhi3700 (Whitehat) created the report May 4, 2024 at 12:13 am

andrew was subscribed to the report by system May 4, 2024 at 12:13 am

All Participants May 4, 2024 at 12:19 am

andrew

Hi,

Thanks for the submission,

Your report is currently being reviewed by Immunefi. We will get back to you once a decision has been made regarding your submission.

Best,

andrew changed the status from Reported to Escalated May 4, 2024 at 12:29 am

All Participants May 4, 2024 at 12:29 am

andrew

Hi,

Immunefi has reviewed this vulnerability report and Escalated it to LayerZero (The World's Largest Bounty) for their technical assessment.

Reasons why Immunefi has escalated:

- claimed impact by the whitehat is in scope for the bug bounty program
- claimed asset by the whitehat is in scope for the bug bounty program
- PoC has been submitted to the project

Since this bug bounty program does not require Immunefi's triaging, note that Immunefi does not:

- check if whitehat's claims are factually correct
- check PoC to understand the validity
- assess the submission's severity

These activities are the project's responsibility.

Remember that, in case of a critical submission, the project has from the escalation day up to 48h to confirm having received the vulnerability and up to

336h for resolving the submission.

If you need help for any reason, please don't hesitate to use the "ask for help" feature to re-subscribe Immunefi to this bug report.

Best

> **3** people were subscribed to the report by andrew May 4, 2024 at 12:29 am

system: Your report has been escalated directly to the project and Immunefi has been unsubscribed. If you need assistance from our team, please request it via the 'Ask Immunefi for help' button on the bottom right of your screen. May 4, 2024 at 12:29 am

andrew unsubscribed from the report May 4, 2024 at 12:29 am

All Participants May 4, 2024 at 1:07 am

abhi3700 (Whitehat)

Thanks for letting me know.

All Participants May 4, 2024 at 1:39 am

abhi3700 (Whitehat)

Currently, set of malicious DVNs could form a group/entity as a trap to gain trust of potential projects by showing them secure flow of verified messages getting executed to & fro. And one fine day, when they find the potential project/contract(s) with good economic benefits have added their DVNs into OApp/OFT/ONFT’s security stack config, their job is done. Now, they just need to wait for the 1st message or may be not at all thereby would mint tokens/assets/transfer fake messages. And they won’t just impact 1 contract pair, but the entire set of crypto projects using parallel execution of messages on destination chain that are actually not sent from source chain. Just imagine! That sort of catastrophe it is.

Luke0x (LayerZero (The World's Largest Bounty)) changed the status from Escalated to Closed May 4, 2024 at 8:44 am

All Participants

May 4, 2024 at 8:44 am

Luke0x (LayerZero (The World's Largest Bounty))

Hey @abhi3700, thank you for your report! The behaviour described in the report is something we would consider as out of scope for the purpose of this bug bounty program, someone doing something like you have described here would be no different to simply deploying a malicious contract and draining user funds. It is up to the UA owner to select their own security properties and to select a DVN set which they trust and this is more of a hypothetical issue than something that is a live vulnerability.

Closed

HIDE DETAILS ^

This report has been closed.

LayerZero (The World's Largest Bounty) has determined that the report is invalid or out-of-scope.

WHAT'S NEXT?

If you disagree with this decision, you can click the “Request Help” button to request assistance from Immunefi. You may also visit our help center.

How was your experience?

Let us know how your experience was within this report so that we can improve Immunefi. All feedback is kept private within Immunefi.

Write a review

Need help?

If you need help with this report, you can request help using the floating button in the bottom right corner.

If your request is not related to a report, you can send us an email at support@immunefi.com