

Hi everyone!

We would like to present you our new Rollup Improvement Proposal RIP-7614, which suggests implementing a call stack that contains opcodes, addresses and function selectors, and exposing through a precompiled contract interface.

[github.com/ethereum/RIPs](https://github.com/ethereum/RIPs)

### **[RIP-7614: Expose call stack to contracts](#)**

ethereum:master

← canercidam:call-stack-precompile

opened 01:19PM - 07 Feb 24 UTC

[  
canercidam  
(https://github.com/canercidam)

[ +165

-0

](https://github.com/ethereum/RIPs/pull/10/files)

<https://ethereum-magicians.org/t/rip-7614-expose-call-stack-to-contracts/18535>

Here is the original Ethereum Magicians thread in case you would like catch up with our community discussions so far:

[Fellowship of Ethereum Magicians – 7 Feb 24](#)

### **[RIP-7614: Expose call stack to contracts](#)**

EIPs

rip

evm

precompile

Hi Magicians! We would like to propose an improvement to increase visibility in L2 EVM execution. RIP-7614: Expose the proposed changes? We propose a new call stack implementation to record opcodes,...

Reading time: 8 mins

Likes: 47 ♥

- This improvement is rooted in the authors' common goal to prevent smart contract exploits.
- Transaction screening is an emerging and promising exploit prevention approach. Here's an example - a DeFi protocol may screen incoming transactions against a negative reputation list and/or for anomalous activity outside of normal user behavior, and revert transactions deemed malicious.
- Unfortunately, transaction screening techniques are susceptible to evasion. DeFi protocols only have visibility into msg.sender

and tx.origin

fields of a transaction, meaning hackers can hide behind various proxies.

- The call stack precompile levels protocols' transaction visibility with attackers, exposes hidden attack contracts and helps reason better about the patterns.

We believe that this is a critical leap forward in implementing effective screening solutions and protecting protocols from attacks on L2 chains, and for this reason we find Arbitrum's adoption of this proposal very valuable.

We are curious to know what you think of this proposal and discuss!