GateSeal

- Source code
- Factory source code
- Deployed contract
- Deployed factory contract
- Deployed blueprint contract

A one-time panic button for pausable contracts.

What is a GateSeal?

A GateSeal is a contract that allows the designated account to instantly put a set of contracts on pause (i.e. seal) for a limited duration. GateSeals are meant to be used as a panic button for crucial contracts in case of an emergency. Each GateSeal is one-time use only and immediately becomes unusable once activated. If the seal is never triggered, the GateSeal will still eventually expire after a set period.

Why use a GateSeal?

To put such crucial components of the Lido protocol asWithdrawalQueue andValidatorsExitBusOracle on hold, the DAO must hold a vote which may take up to several days to pass. GateSeals provide a way to temporarily pause these contracts immediately if the emergency calls for a swifter response. This will give the Lido DAO the time to come up with a solution, hold a vote, implement changes, etc.

Each GateSeal is operated by a committee, essentially a multisig account responsible for pulling the break in case things go awry. However, authorizing a committee to pause/resume the protocol withdrawals would be utterly reckless which is why GateSeals have a number of safeguards in place:

- each GateSeal can only be activated only once and becomes unusable immediately after,
- each GateSeal can only be activated within its expiry period of 1 year maximum and becomes unusable past its expiry timestamp even if it was never triggered,
- the pause duration set at construction time is limited to 14 days.

Thus, the biggest damage a compromised GateSeal multisig can inflict is to pause withdrawals for 14 days, given the DAO does not resume withdrawals sooner via the governance voting.

With all that said, it still is undesirable for a decentralized protocol to rely on a multisig in any capacity. Which is why GateSeals are only a temporary solution; their limited lifespan and one-time use design also act as a kind of "inconvenience bomb", in that once expired, the GateSeal must be replaced and setup anew.

How does it work?

The idea of GateSeals is heavily based aroundPausableUntil contracts which bothWithdrawalQueue andValidatorsExitBusOracle implement. ThesePausableUntil contracts are similar toPausable contracts with one important difference: the paused state is not merely a boolean value, but a timestamp from which the contract is resumed (or unpaused). This allows the user to pause the contract for a certain period, and after this period the contract will resume itself without an explicit call. Thus, the PausableUntil pattern in conjunction with a GateSeal provide a way to pull the break on the protocol in a critical situation.

A GateSeal is set up with an immutable configuration at the time of construction:

- the sealing committee, an account responsible for triggering the seal,
- the seal duration, a period for which the contracts will be sealed,
- the sealables, a list of contracts to be sealed,
- the expiry period, a period after which the GateSeal becomes unusable.

Important to note, that GateSeals do not bypass the access control settings for pausable contracts, which is why GateSeals must be given the appropriate permissions beforehand. If and when an emergency arises, the sealing committee simply calls the seal function and puts the contracts on pause for the set duration.

How are GateSeals created?

GateSeals are created using theGateSealFactory . The factory uses the blueprint pattern whereby new GateSeals are deployed using the initcode (blueprint) stored onchain. The blueprint is essentially a broken GateSeal that can only be used to create new GateSeals.

While Vyper offers other ways to create new contracts, we opted to use the blueprint pattern because it creates a fully autonomous contract without any dependencies. Unlike other contract-creating functions,create_from_blueprint invokes the constructor of the contract, thus, helping avoid the initialization shenanigans.

The blueprint follows theEIP-5202 format, which includes a header that makes the contract uncallable and specifies the version. Edit this page Previous TRP VestingEscrow Next OssifiableProxy