Formal Verification of the DSS-Gate

www.deco.tech

Deco Inc.

Summary

Certora Formal Verification audit report prepared by Deco Inc. for Maker. This document describes the specification and verification of dss-gate smart contract protocol using the Certora Prover Tool.

Audited Files

The work was undertaken in Q3, 2022 . The latest commit reviewed and run through the certora Prover is presented as below :

Smart Contract	Repository	Commit Hash
dss-gate	https://github.com/deco-protocol/dss-gate	4ad74a616
CVL Rules	https://github.com/deco-protocol/dss-gate	80611d5b7

The scope of this formal verification in dss-gate included the following contracts :

- **Dss-gate.sol**

And their dependent contracts :

- **Vat.sol**
- **Vow.sol**

The Certora Prover proved the implementation of the Dss-Gate system is correct with respect to the formal specifications written by the Deco team.

Issues Found

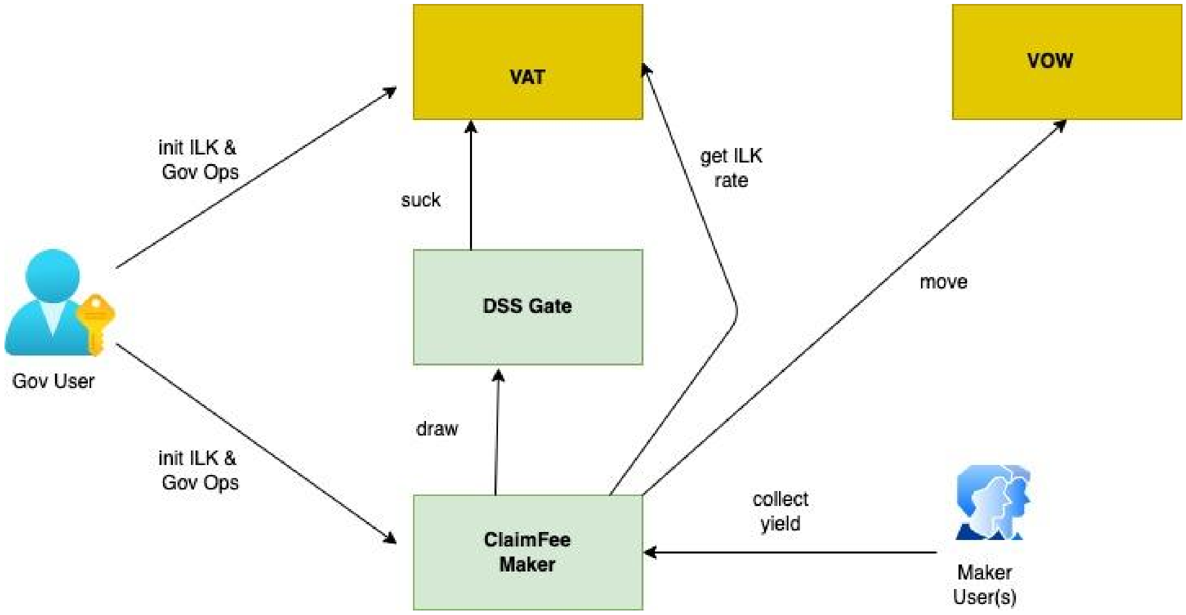
Critical - No Critical issues found.
Major - No Major issues found
Minor - No Minor issues found.

Disclaimer

The Certora Prover takes input a contract and a specification and formally proves that the contract satisfies the specification in all scenarios. The Certora Prover is scoped to the provided specification, and the Certora Prover does not check any cases not covered by the specification. The content of this report should not be construed as a complete guarantee that the contract is secure in all dimensions. In no event shall Deco or any of its employees be liable for any claim, damages or other liability arising, whether in an action of the contract, tort or otherwise, arising from, out of or in connection with the results reported here.

Overview of the verification

Description of the system



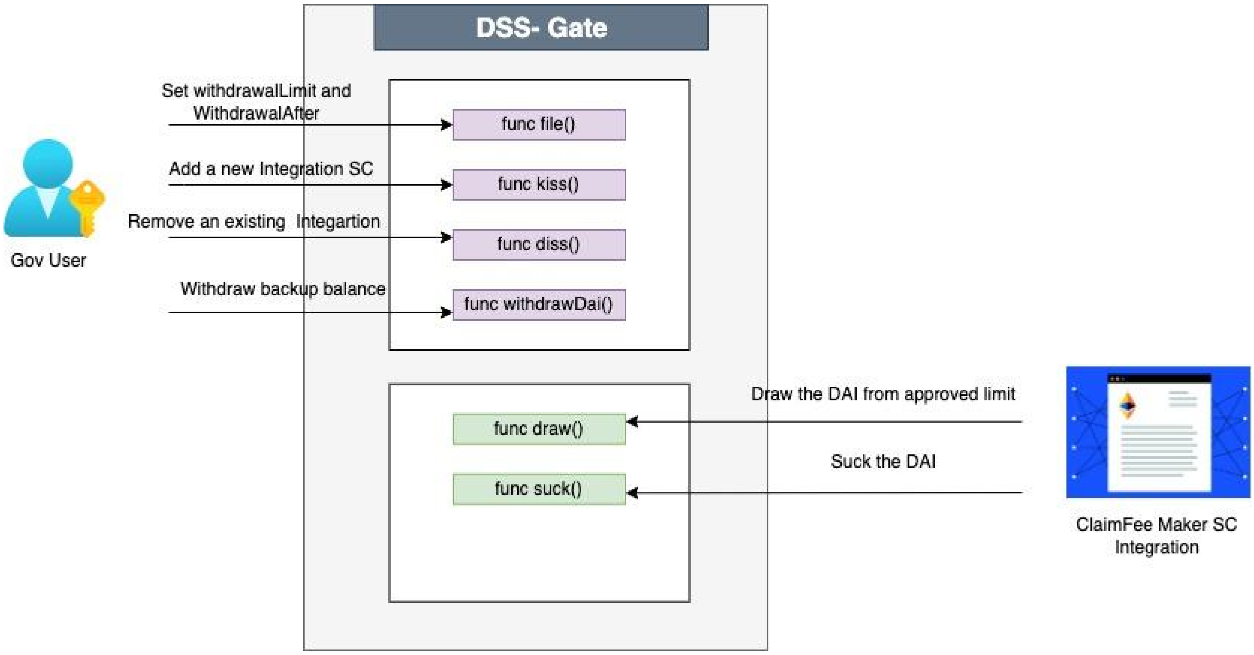
In the above diagram, the highlighted smart contracts (ClaimFeeMaker and DSS Gate) integrate with Core Maker contracts such as VAT and VOW.

`DSS-Gate` is mainly used as a safety lever to protect against the core VAT withdrawal capability. The VAT authorizes a withdrawal limit that it can `suck` DAI from. Upon vat.suck failures, This also holds backup DAI balance to fulfill requests. This approved DAI is further shared upon one or more gate integrations (termed as `buds`). As shown in the diagram, The ClaimFee Maker contract is one among those approved buds which is capable of drawing DAI, and thus transferring to its users.

`ClaimFee Maker` (aka CFM) is an integration of dss-gate module that offers 'Fixed-rate vaults as a Feature' on existing collateral types like ETH-A, WBTC-A etc over a fixed time window. A 'ClaimFee' balance is issued to a user with an existing Maker Vault. This 'Claim Fee' is in turn mapped to a specific ilk (Collateral Type), Issuance time and Maturity, which is coined as the term 'class'. This issued 'ClaimFee' is leveraged to offset the stability fee accrued on the Maker Vault. A user can redeem the 'ClaimFee' to offset the stability fees.

This document mainly focuses on providing a detailed overview of formal verification performed on dss-gate smart contract.

DSS-Gate Contract Overview



The dss-gate (aka Gate1) contract has two important properties :

- an *approval limit (`approvedTotal`)* which indicates the maximum amount of dai that their integrations cumulatively, can withdraw from VAT.
- a *backup balance (`daiBalance`)* that allows integrations to withdraw upon vat.suck failures. Gate will ensure that only a single draw source is leveraged (either vat or backup balance but not both).

Actors

- Ward - A privileged user or a smart contract with administrative authority. These users can set approval limits, withdrawal timestamp, add or remove integrations.
  - Ex : Governance.

- Bud - An integration smart contract which is authorized to draw and suck DAI from the gate. The gate in turn sucks/transfers from VAT.
  - Ex : claim-fee maker

Public Functions

- file() - Used for setting approvedTotal and withdrawalAfter. The approvedTotal is the total DAI that gate integrations can withdraw from vat. The withdrawAfter is a timestamp set to which upon authorized governance addresses can withdraw backup balance.

- kiss() - Add a new integration to gate
- diss() - Remove an existing integration from gate
- rely() - Add a new admin
- deny() - Remove an existing admin
- withdrawDai() - Governance addresses to withdraw backup balance
- draw() - Suck the DAI to approved integration address
- suck() - Suck DAI (for backward compatibility with VAT)
- maxDrawAmount() - Maximum amount that is permissible to withdraw by gate integrations
- daiBalance() - Returns the dai balance held by gate contract.

Private Functions

- accessSuck() - An internal helper function that invokes vat.suck()
- transferDai() - An internal helper function that transfers DAI to destination.

Assumptions and Simplifications

We made the following assumptions during the verification process :

- Several Integrations may be associated with a single Gate contract.

- All the getters of gate contract variables are environment free. The environment free (aka envfree) functions do not use any ethereum environment variables.
- When verifying contracts that make external calls, we assume that those calls can have arbitrary side effects outside of the contracts, but that they do not affect the state of the contract being verified. This means that some reentrancy bugs may not be caught.
- Gate depends on Vat and Vow. The methods in Vow are abstracted.
- All the external, public functions are tested for revert and success paths.

Verification Conditions

Notation

✔ Indicates the rule is formally verified on the latest reviewed commit. Footnotes describe any simplifications or assumptions used while verifying the rules (beyond the general assumptions listed above).

CVL Rules

✔ draw - Verify if the draw methods works as expected

The Gate approved ‘Integrations’ can transfer DAI to the user. Upon sufficient balance, The Vat will be used as the primary source of withdrawal before the Gate’s backup balance is leveraged.

```
{
     _approvedTotal = gate.approvedTotal;
     _gateBalance = vat.dai(gate); // pre gate dai balance

}
     <transaction>
{
    approvedTotal_ = gate.approvedTotal;
    gateBalance_ = vat.dai(gate); // post gate dai balance

}

Properties:
```

```
_approvedTotal > amountToWithdraw ⇒ _approvedTotal - amount

_approvedTotal < amountToWithdraw ⇒ daiBalance(gate) - amount
```

✔ suck - Verify if the suck methods work as expected.

The Gate approved integrations can transfer DAI balance from VAT to the user. Upon sufficient balance, the DAI shall be transferred to the user. The integrations will suck from VAT upto the approved limit as the primary source of funds before the backup balance.

```
{
        _gateBalance = vat.dai(gate)
        _destBalance = vat.dai(destination)
        _approvedTotal = gate.approvedTotal
}

<transaction>

{
      gateBalance_ = vat.dai(gate)
      destBalance_ = vat.dai(destination)
      approvedTotal_ = gate.approvedTotal
}

Properties:

_approvedTotal >= amount ⇒ approvedTotal_ == _approvedTotal -
amount;

_approvedTotal < amount && _gateBalance >= amount ⇒
gateBalance_ == _gateBalance - amount
```

✔ suck_with_revert

Capture and assert all the possible reverts clauses from the suck method.

```
{
     bud(msg.sender) != 1  // sender is not authorized
     vat.wards(gate) != 1  // gate is not authorized
     balance < amount      // Insufficient balance
     vat.live() != 1       // vat is shutdown
     vat.dai() overflow    // ext contract
     vat.sin() overflow    // ext contract
     vat.debt() overflow   // ext contract
     vat.vice() overflow   // ext contract
}
```

✔ withdrawDai

The governance can withdraw the backup balance associated with the Gate contract. The backup balance shall be transferred to the specified user. The amount of withdrawal must be less than backup balance.

```
{
       _gateBalance = vat.dai(gate)
       _destBalance = vat.dai(destination)
}

<transaction>

{
      gateBalance_  = vat.dai(gate)
      destBalance_  = vat.dai(destination)
}

Properties:

_gateBalance >= amount  ⇒  destBalance_ == _destBalance +
amount
```

✔ withdrawDai_with_revert

Capture all the possible revert clauses of the `withdrawDai` function.

```
{
     wards(msg.sender) != 1            // sender is not authorized
     Block.timestamp < withdrawAfter // cannot withdraw before
     msg.value > 0                     // invalid to send ether
     balance < amount                  // Insufficient balance
     vat.dai() overflow                // ext contract
}
```

✔ file

Set the configuration for Gate contract. There are two configurable parameters namely :

1. approvedTotal
2. withdrawAfter

```
<transaction>     // set approvedTotal and withdrawAfter

{
       approvedTotal_ = gate.approvedTotal
       withdrawAfter_ = gate.withdrawAfter

}

Properties:

paramName == “approvedTotal” ⇒ approvedTotal_ == paramValue

paramName == “withdrawAfter” ⇒ withdrawAfter_ == paramValue
```

✔ file_with_revert

Capture all the permissible revert clauses from the `file` function.

```
{
    bud(msg.sender) != 1  // sender is not authorized

     (paramName != “withdrawafter” || paramName != “approvedTotal”)
     // Unknown parameter name

    gate.withdrawAfter < newWithdrawAfter // greater than current
                                                   Value of
Withdrawafter
    msg.value > 0      // cannot send ETH
}
```

✔ maxdraw_amount
✔ maxdraw_amount_revert

The minimum amount that can be withdrawn from Gate by an approved Integration.

```
{
       gateBalance = vat.dai(gate)
       approvedTotal = gate.approvedTotal
}

<transaction>

Properties:

maximum(gateBalance, approvedTotal) // max amount to withdraw
```

✔ kiss

Approve a new integration to Gate

```
<< transaction >>

Properties:

bud(msg.sender) == 1      // A new integration is added to the
gate.
```

✔ kiss_with_revert

Capture all the permissible revert clauses that can be emitted from `kiss` function

```
{
     wards(msg.sender) != 1 // sender is not authorized
     bud(integration) == 1  // integration is already approved
     integration == 0x0  // integration cannot be genesis address
     msg.value > 0      // cannot send ETH
     block.timestamp < withdrawAfter   // cannot add integration 
past
                                      // withdraw after
}
```

✔ diss

Remove an existing integration from the gate.

```
<< transaction >>

Properties:

bud(msg.sender) == 0       // A new integration was removed from the 
gate.
```

✔ diss_with_revert

Capture all the permissible revert clauses that can be emitted from `diss` function

```
{
     wards(msg.sender) != 1 // sender is not authorized
     bud(integration) != 1  // integration was never approved
     msg.value > 0      // cannot send ETH
}
```

✔ rely

Add a new admin(governance) to the Gate.

```
<< transaction >>

Properties:
```

```
wards(user) == 1                 // A new admin is added to the
gate.
```

✔ rely_with_revert

Capture all the permissible revert clauses that can be emitted from `rely` function

```
{
    wards(msg.sender) != 1        // sender is not authorized
}
```

✔ deny

Remove an admin(governance) from the gate

```
<< transaction >>

Properties:

wards(user) == 0       // An existing admin is removed from the
gate.
```

✔ deny_with_revert

Capture all the permissible revert clauses that can be emitted from `deny` function

```
{
  wards(msg.sender) != 1  // sender is not authorized to remove an
                           admin.
}
```

Setup, Installation Instructions

Pre-requisites

- Clone the dss-gate repository

```
git clone git@github.com:deco-protocol/dss-gate.git
```

- Install docker

Follow instructions based on your Operating System

Setup Certora

Step 1 : Pull the Trail-of-bits ETH sec toolbox Container Image

```
docker pull trailofbits/eth-security-toolbox
```

- Change to dss-gate directory

```
cd .../dss-gate
```

Step 2 : Run the Trail-Of-Bits container

```
docker run -it -v "$PWD":/home/training trailofbits/eth-security-toolbox
```

- Install Certora CLI

```
pip3 install certora-cli
```

- Set up the Certora Key

```
export CERTORAKEY=<certora_key> # You can contact Certora Inc., to get a key.
```

- Select the solc version

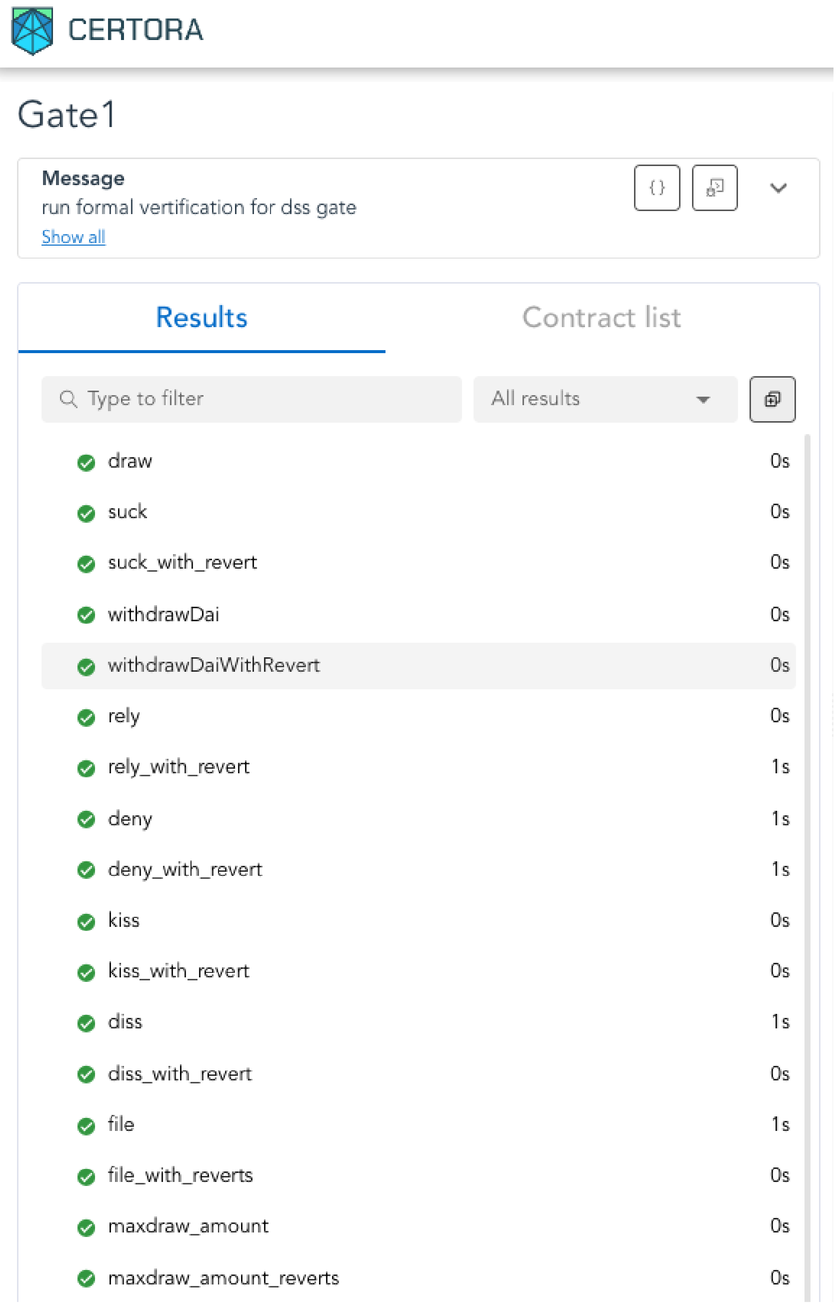
```
solc-select use 0.8.0
```

Note : Make sure you run this command when your pwd is set to the cloned repo folder(i.e dss-gate). This will enable you to access all the files and folders in the current working directory inside the container under `/home/training` folder. This completes the setup.

Step 3 : Run Certora Tests

```
./certora/runCertora.sh "Run dss-gate certora formal verification tests"
```

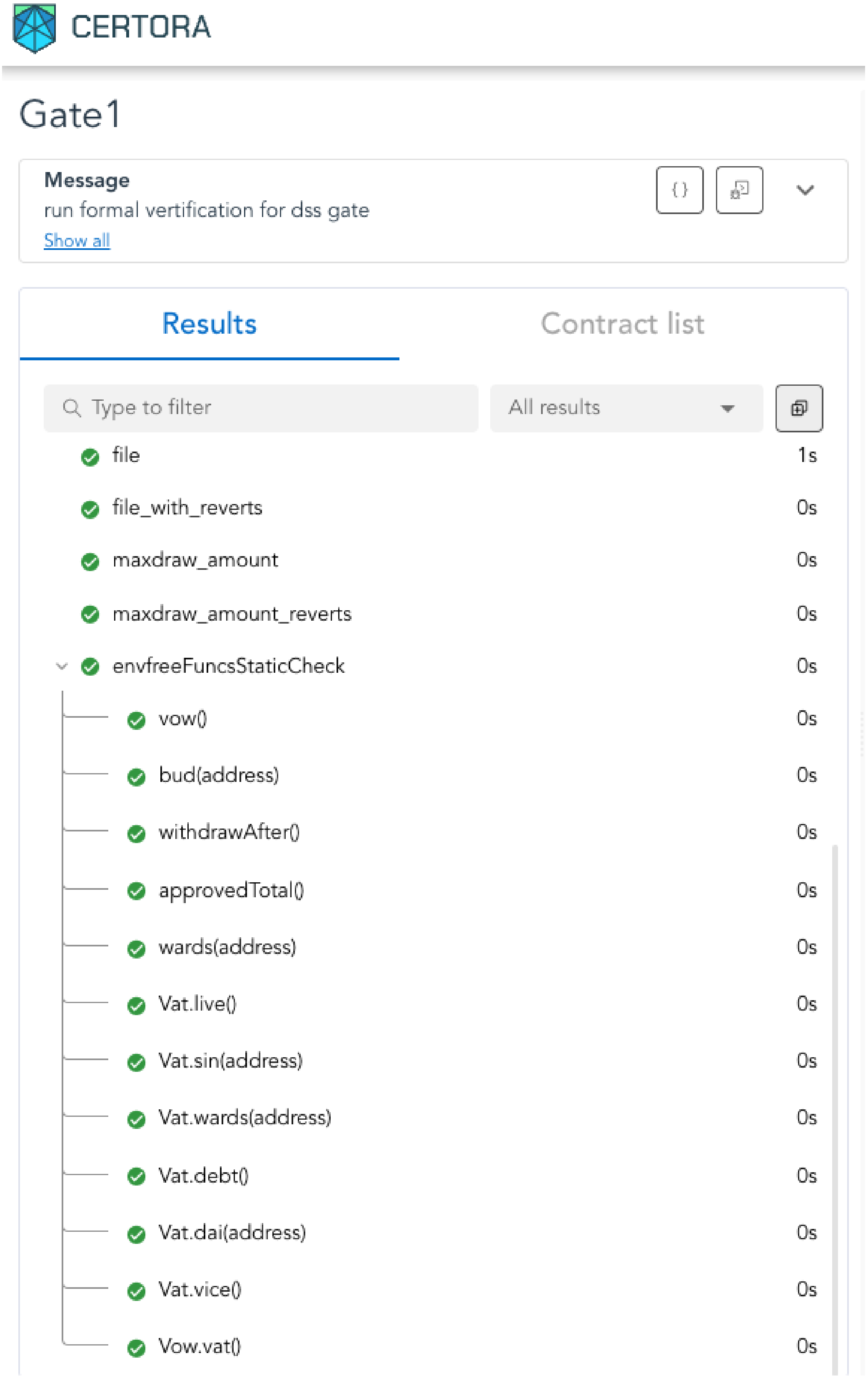
Certora Prover Test Results



Gate1

Message
run formal vertification for dss gate
Show all

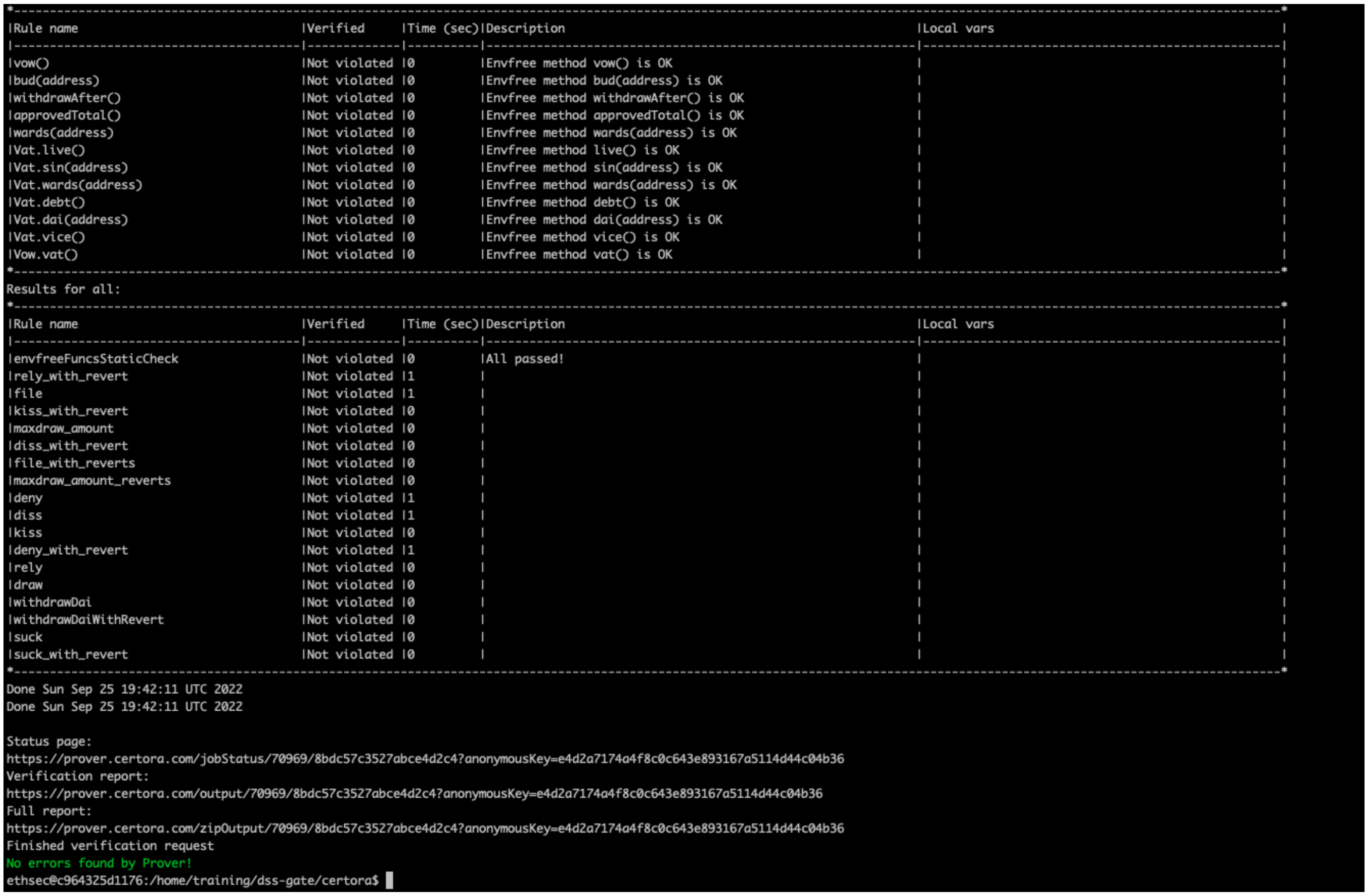
✓	draw	0s
✓	suck	0s
✓	suck_with_revert	0s
✓	withdrawDai	0s
✓	withdrawDaiWithRevert	0s
✓	rely	0s
✓	rely_with_revert	1s
✓	deny	1s
✓	deny_with_revert	1s
✓	kiss	0s
✓	kiss_with_revert	0s
✓	diss	1s
✓	diss_with_revert	0s
✓	file	1s
✓	file_with_reverts	0s
✓	maxdraw_amount	0s
✓	maxdraw_amount_reverts	0s



Gate1

Message
run formal vertification for dss gate
Show all

✓	file	1s
✓	file_with_reverts	0s
✓	maxdraw_amount	0s
✓	maxdraw_amount_reverts	0s
✓	envfreeFuncsStaticCheck	0s
✓	vow()	0s
✓	bud(address)	0s
✓	withdrawAfter()	0s
✓	approvedTotal()	0s
✓	wards(address)	0s
✓	Vat.live()	0s
✓	Vat.sin(address)	0s
✓	Vat.wards(address)	0s
✓	Vat.debt()	0s
✓	Vat.dai(address)	0s
✓	Vat.vice()	0s
✓	Vow.vat()	0s



```
*-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*
|Rule name                               |Verified     |Time (sec)|Description                                                 |Local vars                                        |
|----------------------------------------|-------------|----------|------------------------------------------------------------|--------------------------------------------------|
|vow()                                   |Not violated |0         |Envfree method vow() is OK                                  |                                                  |
|bud(address)                            |Not violated |0         |Envfree method bud(address) is OK                           |                                                  |
|withdrawAfter()                         |Not violated |0         |Envfree method withdrawAfter() is OK                        |                                                  |
|approvedTotal()                         |Not violated |0         |Envfree method approvedTotal() is OK                        |                                                  |
|wards(address)                          |Not violated |0         |Envfree method wards(address) is OK                         |                                                  |
|Vat.live()                              |Not violated |0         |Envfree method live() is OK                                 |                                                  |
|Vat.sin(address)                        |Not violated |0         |Envfree method sin(address) is OK                           |                                                  |
|Vat.wards(address)                      |Not violated |0         |Envfree method wards(address) is OK                         |                                                  |
|Vat.debt()                              |Not violated |0         |Envfree method debt() is OK                                 |                                                  |
|Vat.dai(address)                        |Not violated |0         |Envfree method dai(address) is OK                           |                                                  |
|Vat.vice()                              |Not violated |0         |Envfree method vice() is OK                                 |                                                  |
|Vow.vat()                               |Not violated |0         |Envfree method vat() is OK                                  |                                                  |
*-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*
Results for all:
*-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*
|Rule name                               |Verified     |Time (sec)|Description                                                 |Local vars                                        |
|----------------------------------------|-------------|----------|------------------------------------------------------------|--------------------------------------------------|
|envfreeFuncsStaticCheck                 |Not violated |0         |All passed!                                                 |                                                  |
|rely_with_revert                        |Not violated |1         |                                                            |                                                  |
|file                                    |Not violated |1         |                                                            |                                                  |
|kiss_with_revert                        |Not violated |0         |                                                            |                                                  |
|maxdraw_amount                          |Not violated |0         |                                                            |                                                  |
|diss_with_revert                        |Not violated |0         |                                                            |                                                  |
|file_with_reverts                       |Not violated |0         |                                                            |                                                  |
|maxdraw_amount_reverts                  |Not violated |0         |                                                            |                                                  |
|deny                                    |Not violated |1         |                                                            |                                                  |
|diss                                    |Not violated |1         |                                                            |                                                  |
|kiss                                    |Not violated |0         |                                                            |                                                  |
|deny_with_revert                        |Not violated |1         |                                                            |                                                  |
|rely                                    |Not violated |0         |                                                            |                                                  |
|draw                                    |Not violated |0         |                                                            |                                                  |
|withdrawDai                             |Not violated |0         |                                                            |                                                  |
|withdrawDaiWithRevert                   |Not violated |0         |                                                            |                                                  |
|suck                                    |Not violated |0         |                                                            |                                                  |
|suck_with_revert                        |Not violated |0         |                                                            |                                                  |
*-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*
Done Sun Sep 25 19:42:11 UTC 2022
Done Sun Sep 25 19:42:11 UTC 2022

Status page:
https://prover.certora.com/jobStatus/70969/8bdc57c3527abce4d2c4?anonymousKey=e4d2a7174a4f8c0c643e893167a5114d44c04b36
Verification report:
https://prover.certora.com/output/70969/8bdc57c3527abce4d2c4?anonymousKey=e4d2a7174a4f8c0c643e893167a5114d44c04b36
Full report:
https://prover.certora.com/zipOutput/70969/8bdc57c3527abce4d2c4?anonymousKey=e4d2a7174a4f8c0c643e893167a5114d44c04b36
Finished verification request
No errors found by Prover!
ethsec@c964325d1176:/home/training/dss-gate/certora$ 
```

References

1. Certora Prover - Documentation
2. Certora Tool - Github
3. Ethereum Security Toolbox - Trail of bits