

# EtherTrust User Guide

EtherTrust is a static analysis tool that implements a sound analysis of Ethereum Smart contracts. In particular, it is capable of showing whether a contract has a reentrancy flow that can lead to loss of the money stored in a contract, hence it is of interest to Smart contract developers and users.

## Preliminaries

EtherTrust does not need an installation and can be used via its online interface.

**ETHERTRUST**

[HOME](#) [PUBLICATIONS](#) [SOURCE](#) [TRY ETHERTRUST](#)

### **ETHERTRUST** **A STATIC ANALYSIS TOOL!**

WE PRESENT THE FIRST SOUND AND AUTOMATED STATIC ANALYSIS FOR EVM BYTECODE, WHICH IS BASED ON AN ABSTRACTION OF THE EVM BYTECODE SEMANTICS BASED ON HORN CLAUSES. IN PARTICULAR, OUR STATIC ANALYSIS SUPPORTS REACHABILITY PROPERTIES, WHICH WE SHOW TO BE SUFFICIENT FOR CAPTURING INTERESTING SECURITY PROPERTIES FOR SMART CONTRACTS (E.G., SINGLE-ENTRANCY) AS WELL AS CONTRACT-SPECIFIC FUNCTIONAL PROPERTIES.

All publication materials about the theoretical foundations driving EtherTrust are also available online

## PUBLICATIONS

Foundations and tools for the static analysis of Ethereum smart contracts  
Last Modified: 06/08/2019

A Semantic for the Security Analysis of Ethereum smart contracts  
Last Modified: 07/08/2019

# Analyzing contracts

Currently, the web interface accepts contracts with size  $\leq 100\text{kb}$ . Users are free to submit their contracts saved in a text file. Contracts are expected to be submitted in a form they are stored in a blockchain, i.e., bytecodes.

# ANALYZE YOUR ETHEREUM SMART CONTRACT

Upload Smart Contract

Choose File

1e.txt

Start!

Example Contracts

Report Id

Submitted Smart Contract

Status

In case a user does not have a contract to analyze, still would like to try the service, we provide two example contracts, one (0xd2e16A20dd7B1ae54fB0312209784478D069c7B0) which exhibits a reentrancy flow and a safe contract with respect to reentrancy property (0xAC1Ff2CDb6e88138d78F0Ed39A4bFa1DaD9d27a).

# ANALYZE YOUR ETHEREUM SMART CONTRACT

Upload Smart Contract

Choose File

1e.txt

Start!

Example Contracts

**The DAO at 0xd2e16A20dd7B1ae54fB0312209784478D069c7B0**

```
606060405236156100405760e0
60020a60003504630221038a81
1461004d57806318bdc79a1461
00aa5780638da5cb5b146100be
```

**Random contract at 0xAC1Ff2CDb6e88138d78F0Ed39A4bFa1DaD9d27a**

```
606060405260e060020a600035
04634665096d8114602e578063
be040fb0146036578063dd4670
6414607d575b005b60b8600154
```

Report Id

Submitted Smart Contract

Status

After pressing a **Start!** button analysis is executed. Each analysis task gets assigned a Report Id - a unique hash value. This value can be used if a user leaves website, but still would like to check the results later.

ETHERTRUST

HOME   PUBLICATIONS   SOURCE   TRY ETHERTRUST

ANALYZE YOUR ETHEREUM SMART CONTRACT

Upload Smart Contract

Choose File

1e.txt

Start!

Example Contracts

Report Id	Submitted Smart Contract	Status
5a9a5229-99c4-43a1-a7c7-83d87130190b	1e.txt	In Progress

When analysis is done the webpage changes and a user is welcomed to see the results. Our website fetches the results of analysis automatically, there is no need to refresh the page to observe the behavior. And happens it to be the case that the webpage was closed, analysis results still can be retrieved using the Report Id.

ETHERTRUST

HOME   PUBLICATIONS   SOURCE   TRY ETHERTRUST

ANALYZE YOUR ETHEREUM SMART CONTRACT

Upload Smart Contract

Choose File

1e.txt

Start!

Example Contracts

Report Id	Submitted Smart Contract	Status
5a9a5229-99c4-43a1-a7c7-83d87130190b	1e.txt	Click to View

# Results

For each analyzed contract there can be multiple results, each of them corresponds to a possible reentrancy flow in a contract.

EtherTrust can establish two kinds of result: SATISFIABLE (sat) and UNSATISFIABLE (unsat).

Sat means that there is a satisfying assignment found that corresponds to the reentrancy flow. In other words, if sat result is derived it might be the case that the analyzed contract might have a reentrancy bug.

Unsat means that EtherTrust managed to show the impossibility of having a reentrancy bug for a particular contract. In other words, it means that the analyzed contract is guaranteed to be safe with respect to the reentrancy flow.

**5A9A5229-99C4-43A1-A7C7-83D87130190B**

reentrancyCall\_0\_142

SATISFIABLE

This work is licensed under the Creative Commons Attribution 3.0 Austria License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/at/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.