Crytic - Continuous Assurance for Smart Contracts

We are proud to announce our new Smart contract security producthttps://crytic.io/ . Crytic provides continuous assurance for smart contracts. The platform reports build status on every commit and runs a suite of security analyses for immediate feedback.

The beta access will be open soon. Follow uson twitter to be notified and benefit from the service as soon as possible! The first three months are free.

How Crytic will secure your smart contracts¶

Once connected to your GitHub repository, Crytic will:

- Run our static analyzerSlither
- , which detects the most common smart contracts vulnerabilities and will save you from critical mistakes.
- Run your Truffle tests continuously to ensure that no bug is added while developing your project.

Slither will analyze your codebase for more than 60 security flaws, including reentrancy, integer overflows, race conditions, and many others. Half of these flaw-detectors are private and were not available to the public. They can detect flaws for which public knowledge is limited and that no other tool can find. The recentGridLock bug would have been detected ahead of time using Crytic!

We built this platform for developers, so we integrated it with GitHub. It will watch every commit and branch to ensure that bugs are not added during development. In addition, Crytic will run the checks on every PR to facilitate your code review.

For every security issue found, Crytic will:

- Show you a detailed report on the bug, including source-code highlighting.
- Allow you to create a GitHub issue to keep track of the fixes easily.
- Let you triage the results, so you can decide what needs to be fixed.

Quick Walkthrough¶

Adding Crytic to your system is straightforward: you just need to connect to your GitHub repository. We have first-class support for Truffle; it works out of the box! We also support most of the other smart contract platforms, including Embark, Dapp, and Etherlime. After adding your repository, the dashboard (Figure 1) will show you a summary of the project, like thiscrytic-demo :

Figure 1: Crytic Dashboard From now on, you will benefit from continuous security analyses.

Issue reports¶

Finding an issue is only the first part. Crytic will provide you with detailed information you need about the bug to fix it:

Figure 2: Report A careful reader will notice the vulnerability here:function constuctor creates a public function (with a typo!) that is callable by anyone instead of being run only at initialization. Crytic will detect these types of critical mistakes instantaneously.

Triaging issues¶

Once a bug has been found, the user can decide to:

- create a GitHub issue, to easily keep track of the fix, or
- discard the issue.

Figure 3: GitHub Issue Generated Crytic follows the modifications to your code and reports only new bugs that are introduced. Each new PR will be analyzed automatically:

Figure 4: Pull Request Integration

What’s next for Crytic¶

We are constantly improving Crytic. Expect to see new bug detectors and new features in the future. We are planning to add:

- Echidna
- andManticore

- integration: to ensure your code is checked for custom security properties.
- Automatic bug repair: Crytic will propose patches to fix the issues it finds.
- Slither printer integration
- : to help visualize the underlying details of your code.
- Delegatecall proxychecker
- : to prevent you from making critical—and all too common—mistakes in yourupgradeability process
- .

Questions? Bring them to TruffleCon, and pose them to us at our booth or at our Friday workshopon automated vulnerability detection tools !

Whether or not you can make it to TruffleCon, join ourslack channel (#crytic) for support, and watch@CryticCI to find out as soon as our beta is open.