Fuzzing

Setting up a fuzzer locally

OpenConsole can be built with a `Fuzzing` configuration. To set up a fuzzer, you'll need an `LLVMFuzzerTestOneInput` function. This serves as a way for the fuzzer to attach itself and inject tests into your fuzz target.

To build the fuzzer locally, build the OpenConsole solution in the `Fuzzing` configuration. This should output an executable that runs the fuzzer on the provided test case. In the case of PR #9604, the desired executable is located at `bin\x64\Fuzzing\OpenConsoleFuzzer.exe` .

Resources

- LibFuzzer Docs
- #9604

Setting up OneFuzz

OneFuzz allows us to run our fuzzers in CI and be alerted of new bugs found in this endeavor.

Installing OneFuzz

You can download the latest OneFuzz CLI on their releases page.

Configuring OneFuzz

To run OneFuzz locally, you'll need to configure its endpoint, client ID, and client secret. Windows has a preset configuration available; this can be found at this tutorial on osgwiki.

`onefuzz config --endpoint $(endpoint) --client_id $(client_id) --authority $(authority) --tenant_domain $(tenant_domain)`

NOTE: Our pipeline is already set up with these variables, so you don't need to worry about this when running this on Azure DevOps.

Running a job on OneFuzz

You should now be able to run a job using the following command:

`onefuzz template libfuzzer basic <project> <name> <build> <pool> --target_exe <exe_path>`

- `project` : the name of the project
- `name` : the name of the test
- `build` : the identifier for the build (i.e. commit SHA1)
- `pool` : the VM pool to run this on
- `exe_path` : the fuzzer executable output from building your project

This should also output more information (i.e. job ID) about the newly created job in a JSON format.

Enabling notifications

NOTE: Our pipeline is already set up with this functionality. However, here is a quick guide on how to get it set up and modify it to our liking.

OneFuzz supports multiple notification systems at once including MS Teams and Azure DevOps. See the resources below to learn more about setting these up.

Our pipeline has been set up to create Azure DevOps work items.

Resources

- OneFuzz GitHub
  - Getting started using OneFuzz
  - Releases Page
- Notifications
  - MS Teams
  - Azure DevOps
- OSG Wiki - OneFuzz