Case Study – Fuzzing for Server-like Programs

Scantist's Smart Fuzzer is an automated analysis to find unknown vulnerabilities and their root cause in a platform-agnostic manner customized to any operating environment. It has the added advantage of not requiring source-code access to carry out the vulnerability analysis.

Background

In large organizations like our client "BigTelco", there are large amounts of server-like programs which have the following features that render existing fuzzing solutions infeasible.

Firstly, the target program works like a server and will generally not have an 'exit' condition by default. This means that traditional fuzzing workflow, which assumes when the target program starts and ends, cannot be applied directly to test these targets.

Next, only certain specific parts of the target program need to be analysed as testing the entirety of the program would add to computational and complexity overheads.

Third, the source code of the target program may not always be accessible for a variety of reasons like outsourced development to unavailable legacy code.

Lastly, the target program might be built to run on multiple platforms crossing multiple CPU architectures leading to platform and architecture specific limitations for the fuzzing engine.

These challenges were addressed by Scantist's SmartFuzzer with the following unique capabilities:

- **Binary-only –** Scantist's proprietary binary instrumentation platform gives our fuzzer the ability to conduct coverage-guided greybox fuzz testing without requiring the target program's source code.
- **Cross CPU architecture -** For generality, SmartFuzzer has well supported four CPU architectures – intel X86, intel X86-64, ARM32, and ARM64. Limited support is also offered for PPC and MIPS architectures.
- **Segment fuzz -** SmartFuzzer has the ability to customize the start and end of one round of fuzz testing anywhere within the target program. This allows SmartFuzzer to be configured to only fuzz a piece of code area of the target program.
- **New fuzzing workflow -** Scantist redesigned the fuzzing workflow, enabling the SmartFuzzer to handle server-like programs (i.e. http2(apache), DB daemons etc.)

Results

'BigTelco' ran 10 instances of the Scantist SmartFuzzer across 25 internal programs and found thousands of crashes, with 100+ unique crashes. These crashes were further investigated to yield an undisclosed number of exploitable vulnerabilities – offering greater security assurance for the business line of routers and networking equipment sold by 'BigTelco'.

Start Securing with Scantist

Protect yourself from widely known open-source vulnerabilities in 90% of all your code with an automated solution to detect and remedy vulnerabilities, all while increasing the productivity of your valuable developers with no security expertise required.

To find out more, visit **scantist.com**
Drop us a line at **contact@scantist.com** or better yet, try our solutions for free at and get scanning in 15 minutes at **https://scantist.io**
For press related enquiries, email **marketing@scantist.com**