May the 25th - Fuzzing project report

Goal (19th May - 25th May)🏆

- **Manual review** (test our script)
- **Test** to fix the Pin problem with the leak input in passing the binary from 64 bits to 32 bits
- **Continue** with **call graph** (avoid **loop**)
- **Continue** to maintain the **GitHub** repository

What I have done ? 👨‍💻

Manual review

Resources

Jonathan Salwan - PinTools : **Loop detection** `test.c` script

Review

Firstly, I had to search an **extension** on VSCode to get easily a **call graph** and see if my **Pin script works well**. After hours of research, I **didn't find any extension** to do that. It took me a lot of time. I **tested with Eclipse** (which are a call graph feature) but I **had a problem** to connect Eclipse with my Docker container.

So, how can we a manual review ?

I'm based on a `test.c` **script** of **Jonathan Salman** for detect looping with PinTools. I have put some changes to test efficiently both **features** (**conditional branches** and **function call**) and in the same time, **avoid function loop**.

I had added some versions of the binary of the `test.c` script to simulate bug and review my Pin script.

Binaries and results:
Binaries for the manual review
Results of the manual review

Test input problem 64 bits to 32 bits

Another problem, I tested to install some packages for 32 bits compilation, compile Pin script, compile the binary... I have a problem during the compilation of the binary (about a library).

I tested also with a 32 bits ubuntu image in Docker. Same problem

Improve the call graph feature

My job was to **avoid function loop**. I counted the number of a **CALL instruction**. If I saw more than one time a **CALL instruction** was executed on an **address**, I could conclude that on this address, we had a **loop** which call a function.

So, to know the **call graph** and so the **depth of the bug**, I subtracted the number of **CALL instruction** with the number of **RET instruction**.

I'm not sure about how I implemented it but, with the manual review, the results seem to be good.

The Pin script:
Pin script : bugdepthevaluation.cpp

Where I stopped ? 🛑

I can try to fix problems that I encountered (64 bits to 32 bits for inputs problem, manual review with real example but how?).

I have to read the paper on "Trin-Trin". Very interesting at first sight.