Function Fault Injector

A tool to test your exception handlers

8 Weeks Summer Training
At Microsoft IDC Hyderabad.

Atishay Jain
10783017
3-T5
Index

• About Fault Injection.
• About C# and Visual Studio IDE as well as WinCE and the Platform Builder.
• Application Verifier and the FFI & their working.
• The FFI features.
• DFD
• Working, Coding & Testing.
• Results & Scope of Improvement.
About Fault Injection
“Deliberate insertion of upsets (faults or errors) in computer systems to evaluate its behavior in the presence of faults or validate specific fault tolerance mechanisms in computers.”
What is being tested?

Fault → Error → Failure

Fault tolerant mechanisms
Development Environment
the language...

- Object Oriented.
- Windows Friendly.
- Powerful.
- Integrated into the Visual Studio Suite.
- Popular.
- The language of the reusable components.
Visual Studio

- User Friendly Development IDE.
- Intellisense.
- Debugging features like breakpoint and walk into code.
- Support for direct connection into Windows CE devices.
- Code Generation.
• Target OS.
• Required for Application testing.
• Directly testing of code via the use of the mobile emulator available with the Visual Studio.
• Windows programming on reduced ARMv4 architecture.
Platform Builder

- Makes OS for windows mobile device with embedded software.
- Contained the code to be tested.
- The target software development environment as well as the environment to connect to the actual device.
Application Verifier & FFI
Application Verifier

- Freeware.
- Has both x86 as well as Arm versions.
- A lot of help available online.
- Can easily alter the code and disrupt calls.
- Already used as fault injector for many type of faults.
Working-Appverifier

User → Dll, Function List → ShimGenUI

User → Code & Compile

Code & Compile → .rc, .def, and .c file For making the dll

Dll for the AppVerifier

Applied by the User/3rd Party

Application Verifier

Actual dll

Result
Working-FFI

User/3rd Party

Function Fault Injector

function list, probabilities, return values

User

Compile

Dll for the AppVerifier

Application Verifier

Result

Function Fault Injector

Shimgen

sources, makefile and others for compilation

.bat file and .txt file for instructions

.rc .def and .c file for making the dll
Features
1. Intercept the program flow (via the application verifier) and pass the original function to the shimmed dll where the following can be applied:
   a. Give alternate Return Value.
   b. Fill in a stub function, to replace the original one.
   c. Change the passed parameters to the original function return the result produced henceforth.
   d. Wait for the original function to complete and modify the output or the return value after the function has ended its flow.

2. Have probabilities associated with each type of return and also with the original function.

3. Modify the probabilities dynamically, through the windows registry.

4. No injection into the original dll.
5. Can remove the shim from application via just one command.
6. Save and modify the project any time and apply the modified shim by just replacing the old one.
7. Full flexibility to write any C code and include custom headers.
8. Can also be used for API testing through the modify passed values or parameters option.
9. System dlls can be shimmed to produce other types of faults by say restricting the memory available.
10. A lot of time is saved as the tool automatically generates the code and the support files which in normal shimming process takes much more time.
11. Free form many leaks and flaws present in the original shimgen and hence a good alternative to produce new shims for other purposes.
12. Leaves the C files uncompiled for any modification as the user wants.
DFDs
Context Diagram

User

DLL/Fault Details/Saved File

Function Fault Injector (FFI)

C File to be compiled and applied
Working, Coding & Testing
Function Selection UI
API Filter
Generated Files
| BUILD: [00:0000000153:PROGC | Saving D:\wince5\private\test\tools\Build.dat. | Files | Warnings | Errors |
| BUILD: [00:0000000155:PROGC | Done. | 0 | 0 | 0 |
| BUILD: [00:0000000156:PROGC | Midl | 0 | 0 | 0 |
| BUILD: [00:0000000158:PROGC | Message | 0 | 0 | 0 |
| BUILD: [00:0000000159:PROGC | Precomp Header | 0 | 0 | 0 |
| BUILD: [00:0000000160:PROGC | Resource | 3 | 0 | 0 |
| BUILD: [00:0000000161:PROGC | MASM | 0 | 0 | 0 |
| BUILD: [00:0000000162:PROGC | SHASM | 0 | 0 | 0 |
| BUILD: [00:0000000163:PROGC | ARMSASM | 0 | 0 | 0 |
| BUILD: [00:0000000164:PROGC | MIPSASM | 0 | 0 | 0 |
| BUILD: [00:0000000165:PROGC | C++ | 0 | 0 | 0 |
| BUILD: [00:0000000166:PROGC | C | 7 | 0 | 0 |
| BUILD: [00:0000000167:PROGC | Static Libraries | 0 | 0 | 0 |
| BUILD: [00:0000000168:PROGC | Exe’s | 0 | 0 | 0 |
| BUILD: [00:0000000169:PROGC | Dll’s | 2 | 0 | 0 |
| BUILD: [00:0000000170:PROGC | Preprocess deffile | 2 | 0 | 0 |
| BUILD: [00:0000000171:PROGC | Resx | 0 | 0 | 0 |
| BUILD: [00:0000000172:PROGC | CSharp Compile | 0 | 0 | 0 |
| BUILD: [00:0000000173:PROGC | Other | 0 | 0 | 0 |
| BUILD: [00:0000000174:PROGC | Total | 14 | 0 | 0 |
Execution & Testing

Windows CE>appverif -m calldemo.exe -s shim_heap.dll
Verifier loader: SUCCESS
Windows CE>appverif -m calldemo.exe -s shim_calldemo.dll -opt
Verifier loader: SUCCESS
Windows CE>s calldemo.exe
Windows CE>
Execution & Testing
Results
Results and Future scope

• Accepted and Presently under use.
• Embedded as a part of the WinCe 6 testing tools.
• Future scope includes independent compilation of the WinCE code and hence possibly a release as a component of the next version of the Application Verifier.
Thank You