



# NFTAPE Fault Injector for Win32 Applications

Sandeep Yenugula, Cuong Pham, Daniel Chen, Zbigniew T.Kalbarczyk, Ravi Iyer

## Background

- Reliability is important for all applications.
- Fault injection technique is one way to assess the reliability of applications.
- NFTAPE is a tool developed for performing automated fault injections to evaluate the reliability of applications.

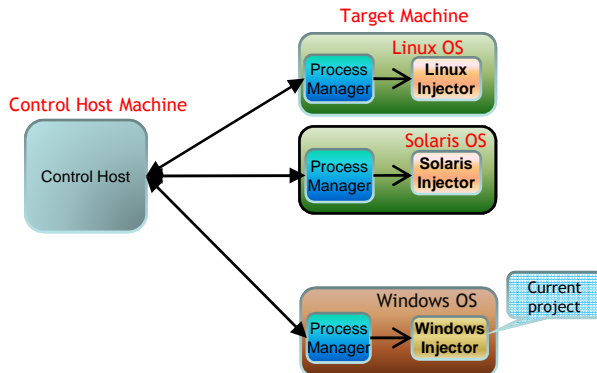


Fig 1:NFTAPE Architecture

- Increased vulnerabilities of Windows Applications indicated the need of reliability assessment and there by taking corresponding measures to improve the reliability of the application.
- Windows Injector can be used for assessing reliability of Windows Applications.

## Goals

- Design a fault injector for Windows Applications which is capable of injecting faults to the text, data, and stack segments of an application.
- Make the injector compatible with the NFTAPE architecture.
- Automate the injection process.
- Collect and log the injection results for analysis.

## Fundamental Questions/Challenges

- How to:
  - ✓ Attach to a target application.
  - ✓ Access the attached application's address space.
  - ✓ Control the injection:
    - When should the injection be done?
    - Where should it be done?
    - What should be injected?
    - How should it be injected?
  - ✓ Automate the fault injection process.

## Research Plan

- Understand the NFTAPE framework and existing Linux and Solaris Injectors.
- Get familiar with the usage of Windows APIs.
- Design a Basic Debugger for Windows Applications.
- Modify the Debugger to function as a fault injector.

## Research Results

➤ Input format of Injector :

Process Name	Breakpoint Address	Destination Address	Injection Type	Mask Function	Mask Size	Mask Data
Stack.exe	0x401291	0x404010	Text/Data/Stack	OverWrite/Add/BitFlip	1byte-4byte	Value

➤ A fault injector for Windows Applications capable of injecting faults to the text, data, and stack segments of the target application.

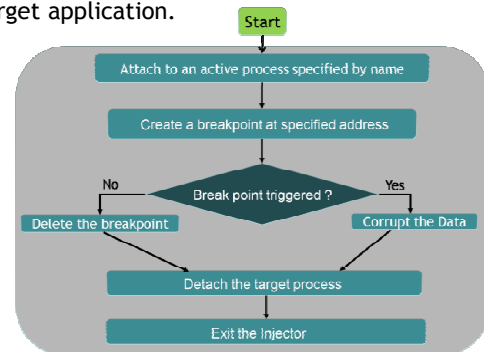


Fig 2 :Win32 Fault Injector

➤ Automated injection process.

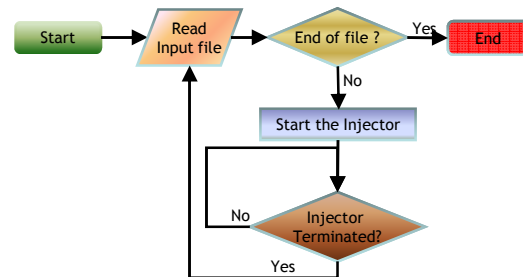


Fig 3: Flow Chart of Automation

## Related Work

- Daniel Chen, Gabriela Jacques-Silva, Zbigniew Kalbarczyk, Ravishankar K. Iyer, Bruce G. Mealey: "Error Behavior Comparison of Multiple Computing Systems: A Case Study Using Linux on Pentium, Solaris on SPARC, and AIX on POWER", Proceedings of the 2008 14th IEEE Pacific Rim International Symposium on Dependable Computing - Volume 00:339-346,2008 .
- Information regarding Windows API's is available in the Microsoft Developer Network site: <http://msdn.microsoft.com/en-us/library/ms>

