X86 Shell Code , “Secure” Self Modifying Code & X Platform Programming

Praseed Pai K.T.

Practice Manager – Mobile Development

Cabot Solutions

http://www.cabotsolutions.com

Personal Blog @

http://praseedp.blogspot.com

Self Modifying Code

- The Executable image of the program modfies itself to customize or fine tune behavior
- The technique is widely used by Crackers , Viruses, Code Obfuscators and even debuggers
- The issue regarding the CPU cache needs to be taken care of.

How do i write Self modifying code ?

- Write Your C Code
- Generate Object Code
- Disassemble and write the instructions to an array
- Make the page executable
- Flush CPU cache
- Invoke the function
- A Demo of the process using Visual C++ command line compiler.

How do i change data inside the instruction stream on the fly ?

- We need to copy the data into instruction stream
- Set the executable bit for the page where instruction stream lies
- Jump to the code.
- A demo program using Visual C++ command line compiler

A Four Function Calculator

- The Calculator uses Recursive descent to parse arbitary mathematical expressions on the fly and generate code for x86 32 bit machine.
- This will give the flexibility of interpretation and performance of compilers

Virtual Machine Run times

- JVM and Microsoft's CLR are the most widely used run times.
- MONO Project has got a Virtual machine
- A simple expression evaluator in C#/.NET (The code can be run on GNU Linux and MAC OS X using Mono)

Why Should We write Self modifying Code ?

- Anti Tampering – usual “excuse”
- Patching Instruction stream for customization
- Mechanism vs Policy in Software systems
- A Unix approach and the Shell script
- Business Process Parameterization using DSL scripts.
- To avoid performance problem , code generation for the native processor.

SLANG4.net – A Open Source Compiler

- Hosted @ http://slangfordotnet.codeplex.com
- Generates .NET IL code (has got a Tree walking interpreter as well -C#)
- Can be used as a base for writing your own DSL
- Has been used as a base for student projects
- JIT translator inside the .net VM generates x86 code with the added advantage of Security Sandbox
- A book “Art of Compiler Construction” available

Migration to Open Systems

- Open systems are “secure”
- The great platform “Lockin”
- Writing (portable !) cross platform code
- Binary “Porting” using Wine
- Virtualization approach
- DOSBOX , Wine , Harbour and Mono - Dangerous for Proprietory systems

DOSBOX – Run your Windows App anywhere

- Can run 16 bit legacy app under Win32/64 , GNU Linux and MAC OS X.
- Can host Legacy development tools like Turbo C++ , CA-Clipper Compiler (80% of business applications in the MSDOS world are written using Clipper or Foxpro) and Foxpro.
- It is a good platform to rescue your “frozen” business apps which are running well.
- For Line of Business Applications , DOS was a good host!

Harbour Project

- Hosted @ http://harbour-project.org
- A xBase compiler which can compile your Clipper/Foxpro source to Win32,GNU Linux and MAC OS X (64bit) binaries
- Millions of Lines of Clipper code is still running all across india (A great economic opportunity beckons)
- A Case study on how i rescued a project which was undertaken in 1994 to run on GNU Linux

Wine and the WineLib

- WINE – Wine Is Not an Emulator
- Can run Windows 32 bit executable under GNU Linux and MAC OS X
- Wine as a SDK for writing Linux Programs ! Aka “Windows Programming under Linux”
- A simple Clock Demo
- A case study where i ported 50,000+ systems in a weekend to run on GNU Linux

Mono

- A Clean room .net implementation which runs on GNU Linux , MAC OS X and Windows
- Has kept up with the changes in Microsoft's .net implementation
- The concept of “Cross Platform C#” and Platform agnostic code in C#
- A case study where i could port a Finanical Accounting/Inventory package written in C# to run under Fedora 10

Thank You

- Q&A