The BREW project at Sandia

Louis Kruger

Feb 14, 2005

Background

- Sandia does a lot of work with binary analysis.
  - Security audits of high consequence information systems
  - Adding protections and logging to existing code
- Each job often requires a custom ad-hoc solution
- They are interested in tools to add sophistication, reduce costs.

Sandia's interest

- Use BREW analysis and disassembly to provide better view of system behavior that is possible using standard disassembly tools.
- Especially interested in embedded BREW scenarios:
  - Analysis done once, rewriting done many times
  - Embedded environment has more limited resources.

Embedded BREW

- Pre-analyze code in the lab and save analysis information.
- In embedded environment, use stored analysis data to run rewriting clients.
- Save precious resources in embedded environment, which is not as powerful as desktop PC or server.

Projects

- Transfer knowledge about BREW
- New BREW features:
  - Portable assembly language generation (e.g. nasm)
  - Better separation of components
- Demonstrate sample applications:
  - Buffer overflow protection
  - Add functional logging for later audit

NASM Codegenerator

- Problem: BREW relies on Borland compiler and linker
  - Borland tools do things in a nonstandard way
  - Rewritten binary has portability issues
- Solution: target another assembler
  - NASM chosen because it uses familiar syntax and is open-source
  - Produces MS and GNU Linker compatible object files
  - Code Generator modified to produce NASM compatible .asm files
- Status: now integrated into BREW distribution

Embedded BREW

- Problem: All of BREW runs as IDA Pro plugin
  - IDA Pro analysis, connector analysis, rewriting, and codegeneration all happen sequentially in one program module.
  - Not practical for on-line rewriting in an embedded system
- Solution: separate rewriting from analysis
  - Need to persist analysis information to disk
  - When BREW is run as a standalone program, load analysis information, then perform rewriting and codegeneration
- Status: functional proof of concept.

Embedded BREW

- Developed C++ object persistence tool
  - Supports all necessary C++ features used by x86fe data structures
    - pointers, references, fixed and variable size arrays, STL types, virtual subclasses (with or without RTTI), templates
  - Reads type descriptions, generates C++ code to perform serialization and deserialization
  - Object graph written to customizable binary format
  - Serialization of notepad.exe analysis data is ~5.7 megs. (~1.7 megs with gzip)
    - notepad.exe is ~50k and contains ~100 CFGs

Future Work

- Support rewriting DLLs (not just .exe files)
- Eliminate assemble/link step entirely – produce binary output from BREW
- Improve analysis accuracy with VSA, etc.
- Reduced footprint