CERT

Function Extraction (FX)

Next-Generation Software Engineering

Dr. Stacy Prowell (sprowell@cert.org)

About SSE
CERT Survivable Systems Engineering

Mission:

To identify and eliminate shortcomings in security and survivability engineering methods.

- Identifying the proper **foundations**
- Developing sound engineering **practices**
- Building **tools** which augment human ability

…to **solve** challenges in constructing systems.

http://www.cert.org/sse/

Creating Next-Generation Systems

Need: Fast and correct development of **ultra-secure**, **ultra-large-scale**, **ultra-high-quality**, and **ultra-secure** systems.

- Can be done, but not with present-day software engineering.
- Complexity and cost limits of technologies evolved over the first fifty years of software engineering have been reached.
- No amount of being careful and trying harder will suffice.

Next-Generation Software Engineering

For future system development, software engineering must be transformed into a **computational discipline**.

- This discipline will be characterized by automated computation of
  - Behavior and security attributes of software
  - Correctness verification of software
  - Composition of components into system architectures
- Other engineering disciplines have made this transformation to computational methods to their everlasting benefit.

CERT

Software Assurance Questions

Past, present, and future:

- Does this foreign-influenced software contain malicious code?
- Does this US-developed software contain code corrupted by insiders?
- Does this acquired software contain errors or vulnerabilities?
- What is this malicious code trying to do?

Getting Answers

With **current** technology:

- Code reading and inspection
  – expensive, fallible, overwhelmed by scale
- Testing
  – exercises only a minor subset of possible behavior
- Model checking
  – explores only properties of models of the code

Bottom line: Can get **some** answers.

Getting Answers

With **next-generation** technology:

- Must understand **everything** the code is doing
  - good, bad, and ugly
- Requires computing the **full behavior** of the code
  - the “all cases of behavior” view
- CERT is developing **Function Extraction (FX)** technology
  - automated computation of full software behavior

Bottom line: Can get **complete** answers.

http://www.cert.org/sse/function_extraction.html

Software Assurance Today

```
public class AccountRecord {
  public int acct_num;
  public double balance;
  public int loan_out;
  public int loan_max;
} // end of AccountRecord

public class AdjustRecord
extends AccountRecord {
  public bool default;
} // end of AdjustRecord

public static AdjustRecord classify_account
(AccountRecord acctRec) {
  AdjustRecord adjustRec = new AdjustRecord();
  adjustRec.acct_num = acctRec.acct_num;
  adjustRec.balance = acctRec.balance;
  adjustRec.loan_out = acctRec.loan_out;
  adjustRec.loan_max = acctRec.loan_max;
    while ((adjustRec.balance < 0.00) &&
      (adjustRec.loan_out + 100) <= adjustRec.loan_max))
  {
    adjustRec.loan_out = adjustRec.loan_out + 100;
    adjustRec.balance = adjustRec.balance + 100.00;
  }
 adjustRec.default = (adjRec.balance < 0.00);
 return adjustRec;
}
```

What does this program do?

- **Read the code** to learn behavior and properties
- 50-year problem: hard, haphazard, error-prone
- **Human time scale** producing suspect information
- Laborious process requiring **significant specialized knowledge**
- ***Change a line…***

CERT

Computing Software Behavior Tomorrow

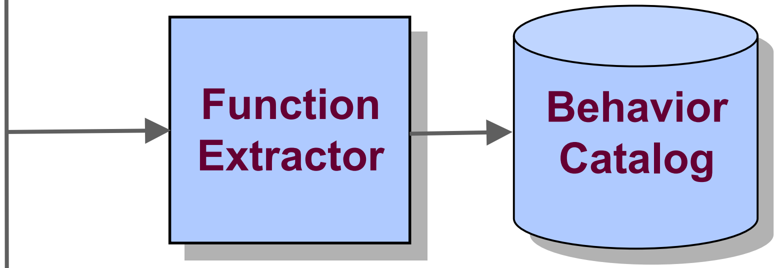
```
public class AccountRecord {
  public int acct_num;
  public double balance;
  public int loan_out;
  public int loan_max;
} // end of AccountRecord

public class AdjustRecord
extends AccountRecord {
  public bool default;
} // end of AdjustRecord

public static AdjustRecord classify_account
(AccountRecord acctRec) {
  AdjustRecord adjustRec = new AdjustRecord();
  adjustRec.acct_num = acctRec.acct_num;
  adjustRec.balance = acctRec.balance;
  adjustRec.loan_out = acctRec.loan_out;
  adjustRec.loan_max = acctRec.loan_max;
    while ((adjustRec.balance < 0.00) &&
      (adjustRec.loan_out + 100) <= adjustRec.loan_max))
  {
    adjustRec.loan_out = adjustRec.loan_out + 100;
    adjustRec.balance = adjustRec.balance + 100.00;
  }
 adjustRec.default = (adjRec.balance < 0.00);
 return adjustRec;
}
```

Function Extractor

- Theoretical foundations of behavior calculation
- Engineering automation



Behavior Catalog

- How does the program transform inputs to outputs in **all cases**?
- The "**as built**" specification of the software, **automatically** calculated.

Function Extraction Prototype Demonstration

Function Extraction Study Results

CERT study on software comprehension and verification showed dramatic improvement with FX:

- **Control group**: traditional reading and inspection
- **Experimental group**: automated FX prototype
- Both given same programs and questions

FX group reduced time to determine program functionality by **three orders of magnitude**.

- FX group 4X better at verifying programs in 1/4 the time
- All achieved with 45 minutes of FX instruction

Report: *The CERT Function Extraction Experiment: Quantifying FX Impact on Software Comprehension and Verification* (CMU/SEI-2005-TN-047)

Back to the Questions with FX

- Foreign-influenced software
  - Behavior can be computed to assure malicious code is not present
- US-developed software
  - Behavior can be computed to assure code has not been corrupted by insiders
- Acquired software
  - Behavior can be computed for analysis to detect errors and vulnerabilities
- Malicious code
  - Behavior of malicious code can be computed for understanding and to develop countermeasures

STAR*Lab
Security Technology Automation Research

STAR*Lab is a new CERT laboratory to create theory-based automated engineering solutions to challenge problems.

Function Extraction for Malicious Code (FX/MC) system development underway in **STAR*Lab**.

- Compute full functional behavior of malicious code in assembly language
- Replace fallible human analysis and timescale with precise computer analysis and timescale
- First capability completed: Transforms spaghetti-logic code into structured form for faster human understanding

STAR*Lab
FX as an Enabling Technology

CERT **STAR*Lab** is exploring FX automation for a variety of applications:

- Code structuring
- Behavior computation
- Security attribute computation (CSA)
- Correctness verification
- Component composition

Our objective is to get these challenge problems off the table once and for all with solid engineering automation.

Thanks!

Contact:

Richard Linger (rlinger@sei.cmu.edu)
Frank Redner (fredner@sei.cmu.edu)
Stacy Prowell (sprowell@cert.org)