Overview of the WiSA Infrastructure and Applications

Somesh Jha, Bart Miller, Tom Reps
{jha,bart,reps}@cs.wisc.edu
Computer Sciences Department
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706-1685
Dynamic Buffer Overflow Detection

- Detect buffer overflows that alter return addresses
- Rewrite binary programs to detect and correct programming flaws
  - Return address verification
  - Safe string operations
- Jonathon Giffin, Hao Wang
Dynamic Buffer Overflow Detection

Binary

IDA Pro
- Parse Binary
  - Build CFGs
  
Connector
- Value-Set Analysis
  - BREW
    - Rewrite
  
Generate Code

Codesurfer
- Build SDG
  - Browse

Clients
- Detect Malicious Code
  - Detect Buffer Overrun
  - Build Program Model

Generated Binary
Malicious Code Detection

• Detect viruses and other malware mutated using obfuscation transformations

• Use static analysis to locate malicious code fragments embedded in a program
  - Deobfuscate program code
  - Identify malicious code sequences split across procedure calls

• Mihai Christodorescu, Somesh Jha
Static Buffer Overflow Detection

• Incorporate buffer overrun detection for C source code in a program understanding framework

• Flow-insensitive, partly context-sensitive
• Formulate and solve problem as linear program
• Two solvers developed
  - Fast and approximate
  - Mathematically precise

• Vinod Ganapathy, Somesh Jha
Static Buffer Overflow Detection

Binary

IDA Pro
Parse Binary
Build CFGs

Connector
Value-Set Analysis

BREW
Rewrite

Generated Code

Source Code

Codesurfer
Parse C
Build SDG
Browse

Clients
Detect Malicious Code
Detect Buffer Overrun
Build Program Model

Generated Binary
Value-Set Analysis

• Create an Intermediate-Representation (IR) for an x86 binary (for further analysis)

• IR - similar to that of a C compiler
  - CFG, used, killed, may-killed variables, points-to sets, etc.
  - Key challenge - understanding memory operations
    • No symbol-table/debug information
    • Explicit memory addresses
    • Indirect addressing
    • Pointer arithmetic

• Gogul Balakrishnan, Tom Reps
Value-Set Analysis

Binary

IDA Pro
  Parse Binary
  Build CFGs

Connector
  Value-Set Analysis

BREW
  Rewrite
  Generate Code

Codesurfer
  Build SDG
  Browse

Clients
  Detect Malicious Code
  Detect Buffer Overrun
  Build Program Model

Generated Binary
Model-Based Intrusion Detection

• Detect attempts to subvert processes

• Specify constraints upon program behavior
  - Statically constructed execution model
  - Dyck model: efficient & context-sensitive

• At run-time, ensure execution obeys model

• Jonathon Giffin, Somesh Jha, Barton Miller
Model-Based Intrusion Detection

13 November 2003
Future Directions

• Support multiple architectures
• Include new static analyses
• Expand rewriting API
• Develop architecture-independent rewriter interface for code creation & insertion
Sandbox86

Application

EDL/ESL Compiler

Event Interceptor

Policy Enforcer

Trace Analyzer

OS

Application ESL

 syscall A

 syscall B

 syscall A

 syscall B
SandBoxX86

Application ESL

EDL/ESL Compiler

Event Interceptor

Policy Enforcer

Trace Analyzer

Application

综合征A

综合征B

ОС

.call A

.call B

syscall A

syscall B
Referring to the EDL definition

 Primitive data types

 Policy specification

**ESL (segment)**

```
edl "c:\malware\kazaa.edl"
map ipaddr

string blocked = "cydoor|doubleclick|adserver|..."

match gethostbyname(name) -> allow
  << addHash(ipaddr, ret, name)>>

match connect(addr, socket,
  << isHashed(addr) >> ) -> deny

match connect(sockaddr, socket,
  << regex(blocked, gethostbyaddr(sockaddr) >> )
  -> deny
```
Slicing Results (Kazaa)

Slicing from `gethostbyname` on addr: 209.10.17.137

Slicing from `connect` on socket: 544

Slicing from `connect` on socket: 1337

Slicing from `send` on socket: 544

Slicing from `recv` on socket: 1337

Slicing from `closesocket` on socket: 544

Slicing from `closesocket` on socket: 1337

gethostbyname(cydor.com..) = 209.10.17.137

connect(544, 209.10.17.137,..)

send(544,..)

closesocket(544)

connect(1337, 209.10.17.137,..)

recv(1337,..)

closesocket(1337)
## Performance Results

### Runtime Overhead

<table>
<thead>
<tr>
<th></th>
<th>W/O Logging</th>
<th>With Logging</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH Client</td>
<td>21%</td>
<td>97%</td>
</tr>
<tr>
<td>RealOne Player</td>
<td>0.8%</td>
<td>1.7%</td>
</tr>
<tr>
<td>Kazaa</td>
<td>2.7%</td>
<td>5.9%</td>
</tr>
<tr>
<td>Original</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

### Memory Overhead

<table>
<thead>
<tr>
<th></th>
<th>Text</th>
<th>Data</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH Client</td>
<td>94KB</td>
<td>42KB</td>
<td>136KB</td>
</tr>
<tr>
<td>RealOne Player</td>
<td>94KB</td>
<td>42KB</td>
<td>136KB</td>
</tr>
<tr>
<td>Kazaa</td>
<td>94KB</td>
<td>45KB</td>
<td>139KB</td>
</tr>
<tr>
<td>Original</td>
<td>62KB</td>
<td>7KB</td>
<td>69KB</td>
</tr>
</tbody>
</table>
Contact Information

• Prof. S. Jha
  - email: jha@cs.wisc.edu

• Prof. B. Miller
  - email: bart@cs.wisc.edu

• Prof. T. Reps
  - email: reps@cs.wisc.edu

• Computer Sciences Dept.  1210 West Dayton Street Madison, WI 53706

Project home page

http://www.cs.wisc.edu/wisa
Model-Based Intrusion Detection

Binary → IDA Pro
  Parse Binary → Build CFGs → Connector
    Value-Set Analysis → BREW
      Rewrite → Generate Code → Codesurfer
        Build SDG → Browse → Build Program Model
          Detect Malicious Code → Detect Buffer Overrun
            Build Program Model → Program Model
Model-Based Intrusion Detection

Binary

EEL
- Parse Binary
- Build CFGs
- Value-Set Analysis
- Rewrite
- Generate Code

Generated Binary

Clients
- Detect Malicious Code
- Detect Buffer Overrun
- Build SDG
- Build Program Model

Program Model
Model-Based Intrusion Detection

Binary

EEL
- Parse Binary
  - Build CFGs
  - Rewrite
    - Generate Code
    - Generated Binary

Data Dependence Graph
- Build DDG
- Build Program Model
- Program Model

Clients