FuncTracker
Discovering Shared Code (to aid malware forensics)

Presenter: Charles LeDoux
University of Louisiana at Lafayette
Shifting Focus of Malware Research

- New focus is on forensics tasks
- Old question: *What?*
- New questions: *Who? Why?*
Relationships: Putting it together

- Single instance → Single piece of the puzzle
- Relationships indicate fitting of pieces
- Key Relationship: Shared Code
Stuxnet, Duqu, … come from the same factory or factories

Stuxnet and Duqu were written on the same platform…by the same group of programmers.

… linked specific portions of code
Key Relationship: Shared Code

Industries:
- Automotive
- Defense
- Financial
- And more...

Linked attacks by similarities in code

Mapped out M.O.
Existing Approaches

- Clustering related malware
- Focus on *whole* binary comparison
  - Would miss single shared function
- Not Scalable
  - $O(n^2)$

**FuncTracker:**
- Small, non-trivial shared code
- Scalable
FuncTracker

- **Granularity: Shared *Functions***
  - Whole binary comparison too coarse
  - Block level too noisy
- **Comparison: Hash Based***
  - Constant time comparison
  - Syntactic and Semantic hashes
- **Exploration: Graph Based***
  - Palantir intelligence platform
Hashes: Heart of FuncTracker

- Represent functions by set of blocks
- Represent each block by single feature
- Sort, concatenate, cryptographic hash

![Diagram showing function blocks and features]

- Block features determine abstraction layer
- BinJuice: Code, GenCode, Semantics, GenSemantics
Blocks: Heart of Hashes

- **Code**
  - Boring ol’ code
  - Fragile against obfuscations

- **GenCode**
  - Abstract out registers and constants
  - Still fragile
    - Instruction reordering
    - Semantically equivalent substitutions

<table>
<thead>
<tr>
<th>Code</th>
<th>GenCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>mov eax, 0x5</td>
<td>mov A, N1</td>
</tr>
<tr>
<td>add ebx, 0x4</td>
<td>add B, N2</td>
</tr>
<tr>
<td>imul eax, ebx</td>
<td>imul A, B</td>
</tr>
</tbody>
</table>
Blocks: Heart of Hashes

- **Semantics**
  - Effect on registers and memory
  - Symbolic interpretation
  - Algebraic simplification
  - Canonical representation

Code

| mov  eax, 0x5 |
| add  ebx, 0x4 |
| imul eax, ebx |

Semantics

| eax = 5 |
| ebx = def(ebx) x 5 + 20 |
Blocks: Heart of Hashes

- **GenSemantics**
  - Analogous to GenCode

```plaintext
Semantics
eax  =  5
ebx  =  def(ebx) * 5 + 20

GenSemantics
A    =  N1
B    =  def(B) * N1 + N2
```
Hashes: Heart of FuncTracker

- Function
  - Code1
  - Code2
  - ...  
  - CodeN

- Function
  - md5(Code1+Code2+...+CodeN)

- Function
  - Block1
  - Block2
  - ...  
  - BlockN

- Function
  - GenSemantics1
  - GenSemantics2
  - ...  
  - GenSemanticsN

- Function
  - md5(GenSemantics1+GenSemantics2+...+GenSemanticsN)
FuncTracker: Exploring Relationships

- Graph representation
- Nodes:
  - Binaries
  - Blocks
  - Functions
- Attributes:
  - Blocks: BinJuice Features
  - Functions: The different hashes
- Edges: “contains” relationship
FuncTracker: Exploring Relationships

- Searches:
  - Traversal
  - Shared attribute
  - Both

- Extensible
  - Time stamp
  - Geographic location
  - Author Information
  - ...

![Diagram of a tree structure with nodes labeled 'Binary', 'Procedure', 'Contains', 'Block']
Example Use Case

- Search for shared behavior
- Start with ground truth
Example Use Case

- Search for shared behavior
- Start with ground truth
- Perform search on shared “GenSemantics”
# Behavior Search Performance

<table>
<thead>
<tr>
<th></th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binaries</td>
<td>17</td>
<td>1</td>
<td>2</td>
<td>90</td>
</tr>
<tr>
<td>Procedures</td>
<td>8</td>
<td>1</td>
<td>18</td>
<td>9889</td>
</tr>
</tbody>
</table>
What’s next?

- Comprehensive evaluation
- Extend Hashing
  - Locality Sensitive Hashing
  - Bloom Filters
Thank You!

Charles LeDoux
charles@charlesledoux.com
University of Louisiana at Lafayette

Arun Lakhotia
arun@louisiana.edu
University of Louisiana at Lafayette

Craig Miles
craig@craigmil.es
University of Louisiana at Lafayette

Vivek Notani
vivek200690@gmail.com
University of Louisiana at Lafayette

Avi Pfeffer
apfeffer@cra.com
Charles River Analytics