Who Wrote This Code?

Identifying the Authors of Program Binaries

Nathan Rosenblum, Xiaojin Zhu, Bart Miller
September 13, 2011

nater@maginatics.com
http://pages.cs.wisc.edu/~nater/
Program Provenance

```
01110101
10101010
10111010
11101010
01101101
```

PROGRAM
Program Provenance
Program Provenance
Program Provenance

stuff happens

C++

Program Provenance
Program Provenance
Authorship Attribution
The Problem
The Problem

PROGRAM
The Problem
The Problem

PROGRAM

mapping
for(int i=0; i<sz;++i)
{
    // etc
}

std::vector<int>::iterator it = foo.begin();
while(it != foo.end())
{
    // etc
}
for(int i=0; i<sz;++i)
{
    // etc

    std::vector<int>::iterator it = foo.begin();
    while(it != foo.end()) {
        // etc
    }
}
Approach

TRAINING DATA

LEARNING FRAMEWORK

provenance model
Approach

TRAINING DATA

LEARNING FRAMEWORK

provenance model
Approach

TRAINING DATA

LEARNING FRAMEWORK

PROGRAM

LEARNING FRAMEWORK

provenance model
code!
program binary
program binary

... 55 89 e5 83 ec 2c 57 56 53 8b 45 0c 8b 00 a3 90 a3 05 08 85 c0 74 2b 83 c4 ...
... 55 89 e5 83 ec 2c 57 56 53 8b 45 0c 8b 00 a3 90 a3 05 08 85 c0 74 2b 83 c4 ...

... push %ebp mov %esp, %ebp sub $0x2c,%esp push %edi push %esi ...

program binary
... 55 89 e5 83 ec 2c 57 56 53 8b 45 0c 8b 00 a3 90 a3 05 08 85 c0 74 2b 83 c4 ...

... push %ebp mov %esp, %ebp sub $0x2c,%esp push %edi push %esi ...

Control Flow Graph

program binary
... 55 89 e5 83 ec 2c 57 56 53 8b 45 0c 8b 00 a3 90 a3 05 08 85 c0 74 2b 83 c4 ...

... push %ebp  mov %esp, %ebp  sub $0x2c,%esp  push %edi  push %esi ...
... push %ebp  mov %esp, %ebp  sub $0x2c,%esp  push %edi  push %esi ...

Control Flow Graph
Call Graph

layout, block contents

program binary

... 55 89 e5 83 ec 2c 57 56 53 8b 45 0c 8b 00 a3 90 a3 05 08 85 c0 74 2b 83 c4 ...
... 55 89 e5 83 ec 2c 57 56 53 8b 45 0c 8b 00 a3 90 a3 05 08 85 c0 74 2b 83 c4 ...

... push %ebp  mov %esp, %ebp  sub $0x2c,%esp  push %edi  push %esi ...

Control Flow Graph
Call Graph
External Libraries
Idioms

<push EBP ; * ; mov ESP, EBP>

<mov [IMM], RAX ; sub [IMM], RAX>
Idioms

single-instruction wildcard

<push EBP ; * ; mov ESP, EBP>

<mov [IMM], RAX ; sub [IMM], RAX>

opcode class abstraction

hidden immediates
Graphlets
Graphlets

code element nodes (e.g. basic blocks)

node colors

typed edges (branch, call, etc.)
Graphlets

code element nodes (e.g. basic blocks)

typed edges (branch, call, etc.)

node colors

Ex: instruction summary graphlets

Color bit field

[Kruegel 2005]

arithmetic

privileged instruction

[ISSTA 2011]
High-level Features

program binary
Call graphlets

printf    ...    printf

...  ...  ...

exit
Call graphlets

printf

exit
Call graphlets

printf

exit
Call graphlets

printf

...... printf

exit

...... printf

Para
Supergraphlets
Supergraphlets

basic blocks
Supergraphlets

insn graphlets

basic blocks
Supergraphlets

insn graphlets

basic blocks

⇒

merged instruction summaries

supergraphlets
Authorship Models
Authorship Models

Bob

Bob

Dave

Bob

Dave

Bob

Alice

7 x <push ebp, mov esp, ebp>

27 x

Paradyn
Authorship Data Sets

**code jam**

- C and C++ programs
- 8-16 programs per contestant
- 191 contestants

**WISCONSIN UNIVERSITY OF WISCONSIN–MADISON**

**COMPUTER SCIENCES (CS 537)**

- C programs
- 4-7 programs per student
- 32 students
Authorship Data Sets

C and C++ programs
8-16 programs per contestant
191 contestants
4-7 programs per student
32 students

537 set is **harder**
- fewer data points
- template code / instructor guidance

- [Image of a bar chart with two bars, one labeled 4-16 and the other labeled 4-16]
<table>
<thead>
<tr>
<th>Term</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-grams</td>
<td>391,056</td>
</tr>
<tr>
<td>Idioms</td>
<td>54,705</td>
</tr>
<tr>
<td>Library calls</td>
<td>152</td>
</tr>
<tr>
<td>Graphlets</td>
<td>37,358</td>
</tr>
<tr>
<td>Supergraphlets</td>
<td>117,997</td>
</tr>
<tr>
<td>Call graphlets</td>
<td>8,062</td>
</tr>
<tr>
<td>Features</td>
<td>1,900</td>
</tr>
<tr>
<td>Count</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>391,056</td>
<td>N-grams</td>
</tr>
<tr>
<td>54,705</td>
<td>Idioms</td>
</tr>
<tr>
<td>152</td>
<td>Library calls</td>
</tr>
<tr>
<td>37,358</td>
<td>Graphlets</td>
</tr>
<tr>
<td>117,997</td>
<td>Supergraphlets</td>
</tr>
<tr>
<td>8,062</td>
<td>Call graphlets</td>
</tr>
<tr>
<td>1,900</td>
<td>Features</td>
</tr>
</tbody>
</table>

20 random programmers
391,056 N-grams  37,358 graphlets
54,705 idioms    117,997 supergraphlets
152 library calls 8,062 call graphlets

1,900 features

CJ 2009  CJ 2010  CS 537
Exact  77.8%  76.8%  38.4%

20 random programmers x 20 experiments
391,056 N-grams  37,358 graphlets
54,705 idioms      117,997 supergraphlets
152 library calls  8,062 call graphlets

1,900 features

CJ 2009       CJ 2010       CS 537
Exact    77.8%    76.8%    38.4%
Top-5     94.7%    93.7%    84.3%

20 random programmers x 20 experiments
391,056 N-grams 37,358 graphlets
54,705 idioms 117,997 supergraphlets
152 library calls 8,062 call graphlets

1,900 features

CJ 2009  CJ 2010  CS 537
Exact    77.8%   76.8%   38.4%
Top-5    94.7%   93.7%   84.3%

still much better than chance
Sensitivity to Problem Size

![Graph](image)

- **Exact**
- **Correct in top 5**

Axes:
- **Y-axis**: Accuracy
- **X-axis**: Authors
Stylistic Clustering
Programs, no training data
Programs, no training data

Conclude:
Programs, no training data

Conclude:
Clustering & Distance Metrics

Euclidean distance

\[ D(x_a, x_b) = \sqrt{(x_a - x_b)^T(x_a - x_b)} \]
Clustering & Distance Metrics

Euclidean distance

\[ D(x_a, x_b) = \sqrt{(x_a - x_b)^T(x_a - x_b)} \]
Clustering & Distance Metrics

Euclidean distance

\[ D(x_a, x_b) = \sqrt{(x_a - x_b)^T(x_a - x_b)} \]
Clustering & Distance Metrics

Euclidean distance

\[ D(x_a, x_b) = \sqrt{(x_a - x_b)^T(x_a - x_b)} \]

Mahalanobis distance

\[ D_A(x_a, x_b) = \sqrt{(x_a - x_b)^T A (x_a - x_b)} \]
Clustering & Distance Metrics

Euclidean distance

\[ D(x_a, x_b) = \sqrt{(x_a - x_b)^T (x_a - x_b)} \]

Mahalanobis distance

\[ D_A(x_a, x_b) = \sqrt{(x_a - x_b)^T A (x_a - x_b)} \]

\[ A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \]
Transfer Learning
Transfer Learning
Transfer Learning
Transfer Learning

Large-margin Nearest Neighbors (LMNN)
Weinberger, Saul 2009

semi-definite program 😞
one-time cost 😊
Stylistic Transfer
Stylistic Transfer
Stylistic Transfer
Cluster Evaluation

![Graph showing accuracy of test set authors over different cluster numbers.]
Cluster Evaluation

supervised classifier accuracy

cluster accuracy (Mahalanobis)

cluster accuracy (Euclidean)
Conclusions
Conclusions

programmer style leaks through compilation
Conclusions

programmer style leaks through compilation

can identify styles without prior examples
Conclusions

- programmer style leaks through compilation
- can identify styles without prior examples
- many open issues
thank you
backup slides
Authorship Open Issues
Future Work: Social Code Networks