unstrip: Restoring Function Information to Stripped Binaries Using Dyninst

Emily Jacobson and Nathan Rosenblum
Paradyn Project

Paradyn / Dyninst Week
Madison, Wisconsin
May 2-4, 2011
Binary Tools Need Symbol Tables

- **Debugging Tools**
  - GDB, IDA Pro,…

- **Instrumentation Tools**
  - PIN, Dyninst,…

- **Static Analysis Tools**
  - CodeSurfer/x86,…

- **Security Analysis Tools**
  - IDA Pro,…
unstrip = stripped parsing + binary rewriting

unstrip:

```assembly
push %ebp
mov %esp,%ebp
sub %0x8,%esp
mov 0x8(%ebp),%eax
add $0xffffffff8,%esp
push %eax
call 80c3bd0
push %eax
call 8057220
mov %ebp,%esp
pop %ebp
```

```assembly
<target8056f50>:
push %ebp
mov %esp,%ebp
sub %0x8,%esp
mov 0x8(%ebp),%eax
add $0xffffffff8,%esp
push %eax
call <target80c3bd0>
push %eax
call <target8057220>
mov %ebp,%esp
pop %ebp
```
New Semantic Information

- Important semantic information: program’s interaction with the operating system (system calls)
- These calls are encapsulated in wrapper functions

Library fingerprinting: identify functions based on patterns learned from exemplar libraries
unstrip = stripped parsing
+ library fingerprinting
+ binary rewriting
<accept>:
  mov  %ebx,  %edx
  mov  %0x66,%eax
  mov  $0x5,%ebx
  lea  0x4(%esp),%ecx
  int  $0x80
  mov  %edx,  %ebx
  cmp  %0xffffffff, %eax
  jae  8048300
  ret
  mov  %esi, %esi

Set up system call arguments

Error check and return

Invoke a system call
The same function can be realized in a variety of ways in the binary.
Semantic Descriptors

- Instead, we’ll take a semantic approach
- Record information that is likely to be invariant across multiple versions of the function

```assembly
<accept>:
    mov %ebx, %edx
    mov %0x66,%eax
    mov $0x5,%ebx
    lea 0x4(%esp),%ecx
    int $0x80
    mov %edx, %ebx
    cmp %0xffffffff83,%eax
    jae 8048300
    ret
    mov %esi,%esi
```
We parse an input binary, locate system calls and wrapper function calls, and employ dataflow analysis.
Building a Descriptor Database

Locate wrapper functions

glibc reference library

Build semantic descriptors

Unstrip:

```
<accept>:
    mov %ebx, %edx
    mov %0x66,%eax
    mov $0x5,%ebx
    lea 0x4(%esp),%ecx
    int $0x80
...
```

{<socketcall, 5>}: accept

{<socketcall, 4>}: listen

{<getpid>}: getpid

...
Building a Descriptor Database

Locate wrapper functions

- glibc reference library

Build semantic descriptors

descriptor database

unstrip

Descriptor Database

unstrip: Restoring Function Information to Stripped Binaries

- `<accept>`:
  - mov %ebx, %edx
  - mov %0x66, %eax
  - mov $0x5, %ebx
  - lea 0x4(esp), %ecx
  - int $0x80
  - ...

- `<socketcall, 5>`: accept
- `<socketcall, 4>`: listen
- `<getpid>`: getpid
- ...

Para dyn

Dyn inst
For each wrapper function
{
    1. Build the semantic descriptor.
    2. Search the database for a match (two stages).
    3. Add label to symbol table.
}
Evaluation

- To evaluate across three dimensions of variation, we constructed three data sets:
  - compiler version
  - library version
  - distribution vendor

- In each set, compile statically-linked binaries, build a DBB, compare unstrip to IDA Pro’s FLIRT

- Evaluation measure is accuracy
Evaluation Results: Compiler Version Study

GCC 3.4.4 Patterns Predicting Each Library

- **unstrip**: Restoring Function Information to Stripped Binaries
Evaluation Results: Library Version Study

Number of glibc 2.2.4 Patterns Predicting Each Library

- 2.2.4
- 2.3.2
- 2.3.4
- 2.5
- 2.11.1

unstrip: Restoring Function Information to Stripped Binaries
Evaluation Results: Distribution Study

Fedora Patterns Predicting Each Library

- **Fedora**
- **Mandrivia**
- **OpenSuse**
- **Ubuntu**

**unstrip**: Restoring Function Information to Stripped Binaries

Legend:
- Blue: unstrip
- Red: IDA Pro
For full details, tech report available online at:


unstrip is available at:


Come see the unstrip demo today at
2:00 or 2:30 (in 1260 WID/MIR)
Extra Slides

- Some additional results
Evaluation Results: Distribution Study

Mandrivia Patterns Predicting Each Library:

- Fedora
- Mandrivia
- OpenSuse
- Ubuntu

accuracy

unstrip and IDA Pro: Restoring Function Information to Stripped Binaries
Evaluation Results: Toolchain Study (one predicts the rest)

Accuracy

<table>
<thead>
<tr>
<th>Version</th>
<th>3.4.4</th>
<th>4.0.2</th>
<th>4.1.2</th>
<th>4.2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>unstrip</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>IDA Pro</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

unstrip: Restoring Function Information to Stripped Binaries
Evaluation Results: Library Version Study
(one predicts the rest)

Accuracy

<table>
<thead>
<tr>
<th>glibc version</th>
<th>unstrip</th>
<th>IDA Pro</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.4</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>2.3.2</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>2.3.4</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>2.5</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>2.11.1</td>
<td>1</td>
<td>0.25</td>
</tr>
</tbody>
</table>

unstrip: Restoring Function Information to Stripped Binaries
Evaluation Results: Distribution Study
(one predicts the rest)

Accuracy

Distribution Vendor

Fedora
Mandrivia
OpenSuse
Ubuntu

unstrip
IDA Pro

unstrip: Restoring Function Information to Stripped Binaries