Adversarial Software Analysis: Challenges and Research

Andrew Walenstein and Arun Lakhotia
Center for Advanced Computer Studies
University of Louisiana at Lafayette
www.cacs.louisiana.edu/labs/SRL
MALWARE DETECTION STORY
On learning the hard way

- Story of work done at the software research laboratory at UL Lafayette
  - ongoing malware research projects
  - Arun Lakhotia started a project in 2000
  - early work: virus detection using model checkers
Model checking for malice

- A PhD student of Arun's was working on detection
  - Key idea: Leverage years of model checking research
    - Define malicious patterns as predicates on suspicious flow
    - Let checker do the hard work of searching for matches
    - It would see through trickery
      - Junk code, constant hiding using arithmetic, etc.
  - Goal: Generic signature matches based on behavior
Architecture of prototype

VIRUS

disassemble → extract procedures → extract control & data flow → verify property

DATABASE

certify / reject
Architecture of prototype

IDA Pro → IDA Pro → textbook analysis → SPIN → PREDICATES

certify / reject
Mixed success with approach

• Research focus was on behaviour patterns
  – defined generic suspect behaviours

• Promise and disappointment
  – in lab conditions: promise
    • ran some examples, looked good
      [Singh&Lakhotia, 2003]
    • *in principle* the research seemed to be in right direction
  – on real-world viruses
    • silently failed on some of the first real viruses tried
    • *in practice* really very little was being solved
Examples of attacks

- disassemble
- extract procedures
- extract control & data flow
- verify property
- certify / reject
Pipeline: Disassembly

decode machine instructions (byte seq)

<table>
<thead>
<tr>
<th>ORIG BYTES</th>
<th>ASSEMBLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>401063: 5d</td>
<td>pop %ebp</td>
</tr>
<tr>
<td>401064: c3</td>
<td>ret %ebp</td>
</tr>
<tr>
<td>401065: 55</td>
<td>push %ebp</td>
</tr>
<tr>
<td>401066: 89</td>
<td>mov %ebp,ESP</td>
</tr>
<tr>
<td>401068: 83</td>
<td>sub $0x8,ESP</td>
</tr>
<tr>
<td>40106b: eb</td>
<td>jmp 0x401072</td>
</tr>
<tr>
<td>40106d: e8</td>
<td>movl 0x00000000,%esi</td>
</tr>
<tr>
<td>401072: e9</td>
<td>jmp 0x401072</td>
</tr>
<tr>
<td>401073: 45</td>
<td>movl 0x00000000,%esi</td>
</tr>
<tr>
<td>401074: 7d</td>
<td>clpl 0x3e7,0xfffffffffc(EBP)</td>
</tr>
</tbody>
</table>

bad disassembly (no jump target, no malicious call)

jump over junk
malicious func
Weak link even with strong chain

• Analysis thwarted by weak link
  – IDA Pro call identification assumptions attacked
    • lead to incorrect control flow
  – IDA Pro linear sweep algorithm attacked with junk
    • lead to incorrect disassembly

• Failure despite established state-of-art
  – “industrial strength disassembler”
  – tried and true flow analysis
  – sophisticated, popular model checker
Lessons learned

- Had us think about what we were doing
  - what theories, technology, science, engineering?
  - what we started thinking was: this is a pretty distinct area

- Issue to us:
  - how to respond to adversarial code
  - not just building a better disassembler
    - (although that’s a serious concern)
    - (and now have publications on disassembler research)
Focusing issue: adversarial relation

• Adversaries to analysis:
  – out there trying to foul up analysis on purpose
  – traditional RE/PA developed in “friendly” contexts

• Adversarial isn’t just “semi-friendly”
  – we know friendly code isn’t always very friendly
    • parsers: dialects, embedded languages, broken code
    • flow analysis: function pointers, callbacks, interrupts
  – RE/PA has its own difficulties
    • is it hard to argue malicious code is categorically any different than ordinary, supposedly-friendly code?
What’s in a name?

• Started using different names
  – settling eventually (?) on *adversarial software analysis* (ASA)

• Similar to “de-obfuscation” idea, which has merit
  – most generic form:
    • obfuscation = making it harder to understand or process
    • de-obfuscation = opposite (making it easier…?)
  – seems like de-obfuscation misses something to us
De-obfuscation?

- We wondered if de-obfuscation speaks to all the issues we wanted to consider
  - e.g. SPIN model checker
    - assume: adversary can make control flow inaccurate
    - ask: what can be done in model checking paradigm to deal with it?
      - working with uncertain flow information
      - providing likelihood (guess) that some malicious behavior pattern could be executed
    - calling it “de-obfuscation” seems like a mismatch to us
Moving forward

- Terminology is not our issue
  - hope we can focus attention on this area of RE/PA
    - lots of interesting, important challenges wide open

- Why CoBaSSA?
  - formative workshop setting
  - hope to:
    - share some thoughts, get some ideas back
    - build momentum in the area

- Remainder of talk:
  - overview issues to raise awareness
  - touch on possible future research directions
But before continuing

• One can tell much about an area by:
  – type of issues it cares deeply about
  – sort of knowledge it seeks to collect
    • theories, models, techniques
  – criteria in evaluations

• So to emphasize the connection to research identity, I’ll try to write statements of the form:
  “You might be an ASA researcher if…”
Remaining in the talk

Thrilling Malware Detection Story

Focus: adversarial attack and response failures, vulnerabilities, hardening

Future directions in ASA research
Adversarial attack and response

• Wouldn’t care about adversaries if they didn’t cause any grief
  – usually an assumption of imperfection on the response
    • e.g., can’t distinguish between code and data

• Imperfection assumption brings 3 issues to fore:
  1. failures
  2. vulnerabilities
  3. hardening analysis
    • removing vulnerabilities
    • mitigating failures, limit damages

• Theories/methods in ASA will relate to these
You might be an ASA researcher if...

- You might be an ASA researcher if
  ... you assume disassembly / cf / df is inaccurate.
  ... you want to find and measure your vulnerabilities.
  ... your statement of technical merit relates to hardening the analysis, not making a more capable one.
Remaining in the talk

Thrilling Malware Detection Story
Focus: adversarial attack and response
failures, vulnerabilities, hardening
Future directions in ASA research
Failures: mode & effect

• Lots of history of failure analysis
  – engineering, security

• Mode analysis:
  – hard failures: failing to produce output
  – soft failures: erroneous or inaccurate output
    • e.g., control flow inaccurate on obfuscated calls

• Effect analysis:
  – single chain is brittle: single point of failure
Failure analysis: mode, effect
Vulnerabilities

• Extraction / analysis contains vulnerabilities
  – a site to attack, even assuming correctly implemented

• Three main types we can think of:
  – Static analysis
    • “computationally difficult static analysis problems”
  – Assumption
    • assumed conventions, models
      – does your model assume only single entry procedures?
  – Infrastructure / environment
    • anti-debugging
Hardening and reliability concerns

• Design and evaluation goals in ASA typically focus on robustness or reliability
  – can sometimes contrast with RE/PA
    • e.g. compiler:
      – automated, conservative, efficiency required
    • e.g., forensic analysis environment
      – interactive, speculative, may be willing to run overnight
You might be an ASA researcher if...

• You might be an ASA researcher if...
  ... your introduction mentions failure modes & effects.
  ... you’re measuring reductions in vulnerabilities.
  ... graceful degradation is at least as important as accuracy.
Remaining in the talk

Thrilling Malware Detection Story
Focus: adversarial attack and response failures, vulnerabilities, hardening
Future directions in ASA research
Possibilities for the future

- We have a list of directions that seem promising
  - none that haven’t been discussed in RE before
  - but they look particularly useful in ASA

- Try to look into the future
  - view some directions as changes to the pipeline
Feeding back data & processing opportunistically

- breaking strict phases
- allows top-down knowledge to simplify lower-level
- allows handling of circular definitions
  - e.g., disassembler of [Kruegel et.al 2005]
    » code bytes are those could be executed
    » correct disassembly needs accurate control flow
    » control flow information needs disassembly
- “solution” is to revisit disassembly after control flow
Possibilities for the future

- Better use of history / knowledge
  - black/white lists are just one type of knowledge of the past
  - e.g., disassembler of [Kruegel et.al 2005]
    - used database of probabilities for bytes being code
- Better remembering history / knowledge
  - case based reasoning seem well-matched to the problem
Possibilities for the future

- Improving human-computer cooperation
  - forensic analysis will surely involve humans
  - expect advances in joint decision making
    - even for (especially for?) earlier phases
Possibilities for the future

• Not shown on chart (or discussed in paper)
  – removing assumptions in models and techniques
    • e.g.? models don’t assume procedures are single-entry
    • e.g.? slicing in presence of errors or uncertainty
  – imperfect representations and their processing
    • recording and analyzing guesswork or confidence
  – information fusion
    • multiple redundant independent systems
      – e.g., multiple disassemblers
    • aim to increase reliability of system
    • but need to fuse information; integrate knowledge
Conclusions

• Short points:
  – we find thinking in ASA terms to be helpful
    • want to pass it on, discuss it with people
  – ASA shifts focus on satsificing under uncertainty
    • biggest differences to us so far have been:
      – assumptions
        » that disassembler, etc. WILL fail
        » that we must reduce failure rate & the effect of failure
      – implications
        » that downstream components may affect upstream ones
Thank you!

Software Research Lab
Center for Advanced Computer Studies
University of Louisiana at Lafayette

Arun Lakhotia
Andrew Walenstein

M.Sc. Student
Rachit Mathur

Ph.D. Students
Michael Venable
Mohamed R. Chochane

CoBaSSA 2005