AN OBSERVATIONAL INVESTIGATION OF REVERSE ENGINEERS’ PROCESSES AND MENTAL MODELS

Daniel Votipka, Seth Rabin, Kristopher Micinski, Michelle L. Mazurek, and Jeffrey S. Foster
SOFTWARE REVERSE ENGINEERING

Looking at someone else's code to figure out how it works!
SOFTWARE REVERSE ENGINEERING

Looking at someone else’s code to figure out how it works!

• Vulnerability Discovery  

[Links: hackerone, bugcrowd]
SOFTWARE REVERSE ENGINEERING

Looking at someone else’s code to figure out how it works!

• Vulnerability Discovery
• Malware Analysis
SOFTWARE REVERSE ENGINEERING

Looking at someone else’s code to figure out how it works!

- Vulnerability Discovery
- Malware Analysis

30 students and RE professionals took ~39 minutes on average to reverse engineer decompiled code snippets with <150 lines.

Yakdan et al. 2016
CURRENT TOOLS
CURRENT TOOLS

Many sophisticated tools developed by academia, industry, and practitioners
CURRENT TOOLS

Many sophisticated tools developed by academia, industry, and practitioners
  • Ad-hoc, based on the developers’ intuition
CURRENT TOOLS

Many sophisticated tools developed by academia, industry, and practitioners
  • Ad-hoc, based on the developers’ intuition

Limited theoretical model of RE process
  • Actions, habits, and mental models
CURRENT TOOLS

Many sophisticated tools developed by academia, industry, and practitioners
  • Ad-hoc, based on the developers’ intuition

Limited theoretical model of RE process
  • Actions, habits, and mental models

Goal:
Develop a refined RE process model to guide future tool development.
PROGRAM COMPREHENSION

How developers process unfamiliar code during modification, maintenance, and debugging tasks

• Hypotheses/Questions - approach unfamiliar programs from a non-linear, fact-finding perspective
• Beacons - Patterns that allow the reader to quickly infer program behavior
• Simulation methods - Any process for parsing the program to determine its function
PROGRAM COMPREHENSION

How developers process unfamiliar code during modification, maintenance, and debugging tasks

- **Hypotheses/Questions** - approach unfamiliar programs from a non-linear, fact-finding perspective
- **Beacons** - Patterns that allow the reader to quickly infer program behavior
- **Simulation methods** - Any process for parsing the program to determine its function

Is reverse engineering different?
- No access to source code, developers, or documentation
- Adversarial environment
RESEARCH QUESTIONS

What high-level process do REs follow?
- Steps of the process
- Mental models

What technical approaches do REs use?

Are the RE and Program Comprehension processes different?
RESEARCH QUESTIONS

What high-level process do REs follow?
  • Steps of the process
  • Mental models

What technical approaches do REs use?

Are the RE and Program Comprehension processes different?
OBSERVATIONAL INTERVIEWS


- Participants demonstrated how they reverse engineered a recent program
- Noted and asked further questions regarding items of interest:
  - Beacons
  - Hypotheses/Questions
  - Simulation Methods
  - Decisions
  - Resources
## PARTICIPANTS

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Participants</td>
<td>16</td>
</tr>
<tr>
<td>Gender</td>
<td>14 Male, 1 Female</td>
</tr>
<tr>
<td>Age (Median)</td>
<td>18-29</td>
</tr>
<tr>
<td>Location</td>
<td>7 US states, 5 countries</td>
</tr>
<tr>
<td>Education (Median)</td>
<td>B.S.</td>
</tr>
<tr>
<td>Yrs. Experience</td>
<td>8.94 years</td>
</tr>
<tr>
<td>Skill</td>
<td>4.06 (Advanced)</td>
</tr>
<tr>
<td>Job</td>
<td>12* Vulnerability Discovery, 5* Malware Analysis,</td>
</tr>
</tbody>
</table>

*One participant performed both malware analysis and vulnerability discovery for employment*
## PARTICIPANTS

<table>
<thead>
<tr>
<th>Category</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participants</td>
<td>16</td>
</tr>
<tr>
<td>Gender</td>
<td>14 Male, 1 Female</td>
</tr>
<tr>
<td>Age (Median)</td>
<td>18-29</td>
</tr>
<tr>
<td>Location</td>
<td>7 US states, 5 countries</td>
</tr>
<tr>
<td>Education (Median)</td>
<td>B.S.</td>
</tr>
<tr>
<td>Yrs. Experience</td>
<td>8.94 years</td>
</tr>
<tr>
<td>Skill</td>
<td>4.06 (Advanced)</td>
</tr>
<tr>
<td>Job</td>
<td>12* Vulnerability Discovery, 5* Malware Analysis,</td>
</tr>
</tbody>
</table>

*One participant performed both malware analysis and vulnerability discovery for employment*
## PARTICIPANTS

<table>
<thead>
<tr>
<th>Participants</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>14 Male, 1 Female</td>
</tr>
<tr>
<td>Age (Median)</td>
<td>18-29</td>
</tr>
<tr>
<td>Location</td>
<td>7 US states, 5 countries</td>
</tr>
<tr>
<td>Education (Median)</td>
<td>B.S.</td>
</tr>
<tr>
<td>Yrs. Experience</td>
<td>8.94 years</td>
</tr>
<tr>
<td>Skill</td>
<td>4.06 (Advanced)</td>
</tr>
<tr>
<td>Job</td>
<td>12* Vulnerability Discovery, 5* Malware Analysis,</td>
</tr>
</tbody>
</table>

*One participant performed both malware analysis and vulnerability discovery for employment*
THREE PHASE RE MODEL

Overview

Sub-component Scanning

Focused Experimentation
OVERVIEW

Full program

Overview

- List strings and APIs
- Run the program
- Review metadata

Identify specific functions and code segments to focus on

Sub-component Scanning

Focused Experimentation
SUB-COMPONENT SCANNING

Overview

Program slices

Sub-component Scanning

• Scan beacons

Specific hypotheses/questions that require concrete information

Focused Experimentation
var zzo = function() {
  var ttw = [
    "http://www.microsoft.com/",
    "http://www.google.com",
    "http://www.bing.com"
  ];
  for (var i = 0, h, wep; i < ttw.length; i++){
    try {
      var h = new ActiveXObject("MSXML2.ServerXMLHTTP.6.0");
      h.open("GET", ttw[i]);
      h.setRequestHeader("User-Agent", _._.u);
      h.setRequestHeader("Cache-Control", "no-cache");
      h.setRequestHeader("Pragma", "no-cach");
      h.setRequestHeader("Connection", "close");
      h.send("");
      wep =
          new Date(
            h
            .getAllResponseHeaders()
            .split("Date: ")
            .pop()
            .split("\n")
            .shift()
            .getTime() / 1000;
      if (1388534400 < wep) {
        return wep;
      }
    } catch (e) {}}
it's just trying to make a connection to each of those [websites].
if it’s able to make a connection, it’s going to return a non-zero value.
“usually you see this activity if [malware] is trying to see if it has connectivity.”
SUB-COMPONENT SCANNING

Overview

Program slices

Sub-component Scanning

- Scan beacons
- Data flow/control flow paths

Specific hypotheses/questions that require concrete information

Focused Experimentation
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    z = 1
    val = a
    a += 1
}
if val == -1 {
    id_free(x)
}
else {
    safe()
}
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    z = 1
    val = a
    a += 1
}
if val == -1 {
    id_free(x)
}
else {
    safe()
}
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    z = 1
    val = a
    a += 1
}
if val == -1 {
    id_free(x)
}
else {
    safe()
}
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    z = 1
    val = a
    a += 1
}
if val == -1 {
    id_free(x)
}
else {    
safe()
}
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    z = 1
    val = a
    a += 1
}
if val == -1 {
    id_free(x)
}
else {
    safe()
}
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    z = 1
    val = a
    a += 1
}
if val == -1 {
    id_free(x)
}
else {
    safe()
}
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    ...
    val = a
    ...
}
if val == -1 {
    id_free(x)
}
else {
    safe()
}
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    ...
    val = a
    ...
}
if val == -1 {
    id_free(x)
}
else {
    safe()
}
val = 0
if y == 1 {
    x = y
    val = 1
}
else{
    a = -1
    ...
    val = a
    ...
}
if val == -1 {
    id_free(x)
}
else {
    safe()
}
SUB-COMPONENT SCANNING

Overview

Program slices

Sub-component Scanning

• Scan beacons
• Data flow/control flow paths

Specific hypotheses/questions that require concrete information

Focused Experimentation
FOCUSED EXPERIMENTATION

Overview

Sub-component Scanning
- Execution traces or a few lines of code
- Test hypotheses and produce concrete answers

Focused Experimentation
- Execute under inspection
- Compare to reference function
- Read line-by-line
FOCUSED EXPERIMENTATION

Overview

Sub-component Scanning
- Execution traces or a few lines of code
- Test hypotheses and produce concrete answers

Focused Experimentation
- Execute under inspection
- Compare to reference function
- Read line-by-line

No more than 50 lines reviewed by any participants
CROSS-PHASE TRENDS

Overview

Sub-component Scanning

Focused Experimentation
CROSS-PHASE TRENDS

Overview

Sub-component Scanning

Focused Experimentation

Methods

Static

Dynamic
CROSS-PHASE TRENDS

Overview

Sub-component Scanning

Focused Experimentation

Role of Experience
- Choose focus areas
- Recognize behaviors/vulnerabilities

Methods
- Static
- Dynamic

Choose method
CROSS-PHASE TRENDS

Overview

Sub-component Scanning

Focused Experimentation

Preferred tools improve readability
DISCUSSION

• Guidelines for usable tool design
DISCUSSION

- Guidelines for usable tool design
- Framework for tool evaluation
DISCUSSION

• Guidelines for usable tool design
• Framework for tool evaluation
• Insights for RE automation
SUMMARY

Three Phase Model:
• Overview
• Sub-component Scanning
• Focused Experimentation

Cross-phase trends:
• Begin with static methods and finish with dynamic
• Experience guides where to look

Takeaways:
• Guidelines for usable tool design
• Framework for tool evaluation
• Insights for RE automation

Questions:
dvotipka@cs.umd.edu
sec-professionals.cs.umd.edu