Comparing Malware Evasion Theory with Practice: Results from Interviews with Expert Analysts

Miuyin Yong Wong, Matthew Landen, Frank Li, Fabian Monrose, Mustaque Ahamad

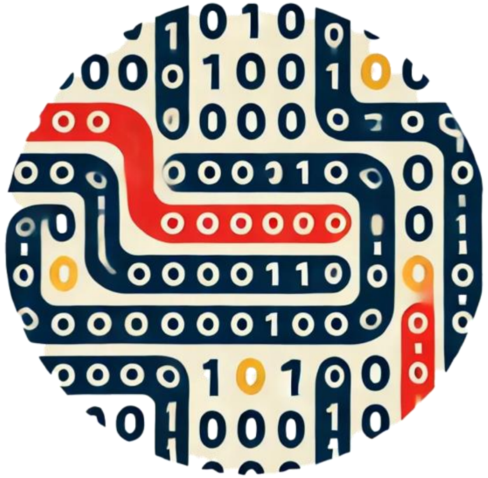
Georgia Tech

Malware Analysts

Malware Analysis is Not Straightforward

Malware Evasion Techniques

Obfuscation



Anti-Sandbox

Anti-disassembly

Anti-debugger

Little Research on the Challenges in Practice

Research Question

What evasive techniques are currently viewed as challenging by expert analysts and why?

User Study Methodology

1. **Recruitment**
   - Security organization mailing lists, group chats, personal contacts, social media, snowballing
2. **Pre-screening Survey**
   - Validate identity and experience
   - Invited 27 respondents
3. **Semi-structured interviews**
   - Challenging evasion techniques in practice
   - Workflows used to analyze evasive malware

Participants

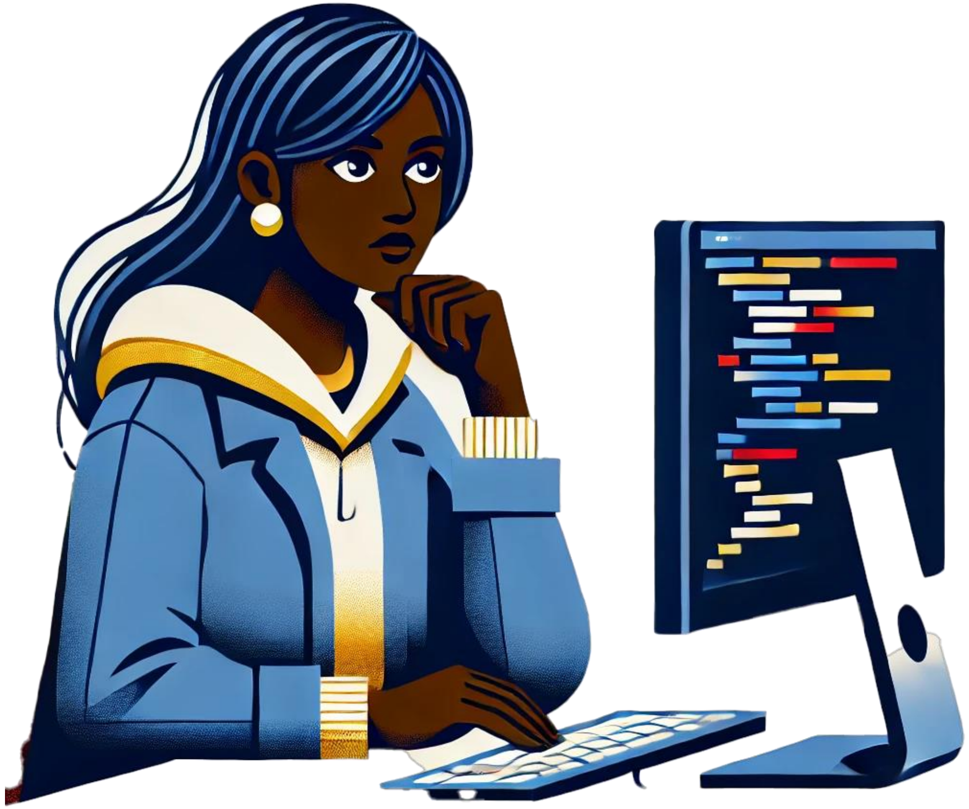
- 24 total participants
- Average of 10 years of experience
- Work in 15 established security groups of well-known companies (Google, Mandiant, FireEye, IBM, Proofpoint, SecureWorks, …)
- Variety of educational background

Most Challenging Evasion Techniques in Practice

Georgia Tech

#1 Obfuscation (9 participants)

“That obfuscation can show up in any kind of malware. JavaScript, PowerShell, Windows PEs, you name it. It’ll be everywhere.” – P1

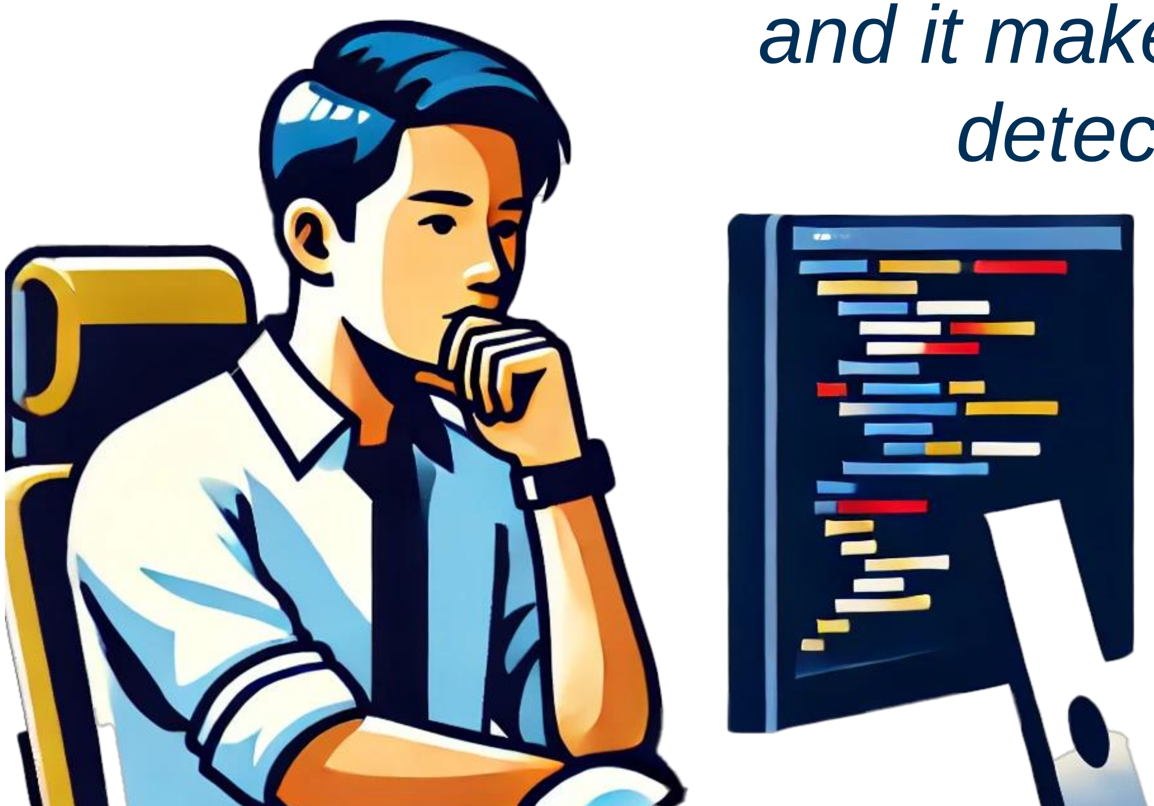


#2 Anti-disassembly (6 Participants)

“Alternative languages are becoming more problematic. Golang, Rust, and Delphi are three languages that when you write a program and compile it is a lot less straight forward than looking at compiled C” – P2

#3 Anti-debugging (3 participants)

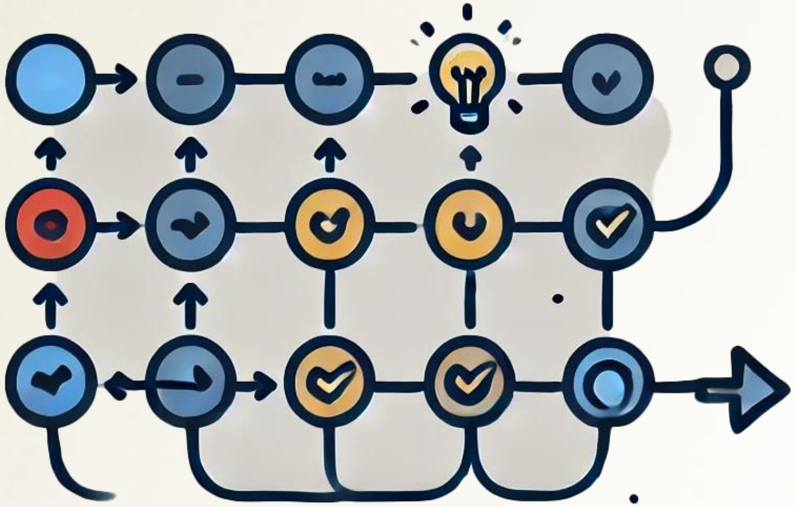
“If it’s designed in a way that I can’t even follow the code execution [...] that makes it really difficult to figure out which blocks I should narrow in on for static analysis, and it makes it really difficult to create detection signatures”– P20



#4 Anti-Sandbox (2 participants)

“The anti-sandbox stuff I could [...] just run out on a real system, and that real system is still instrumented with a lot of the same tools”– P20

What Workflows do Analysts Follow to Tackle these Challenges?



Georgia Tech

Workflows for Handling Evasive Techniques

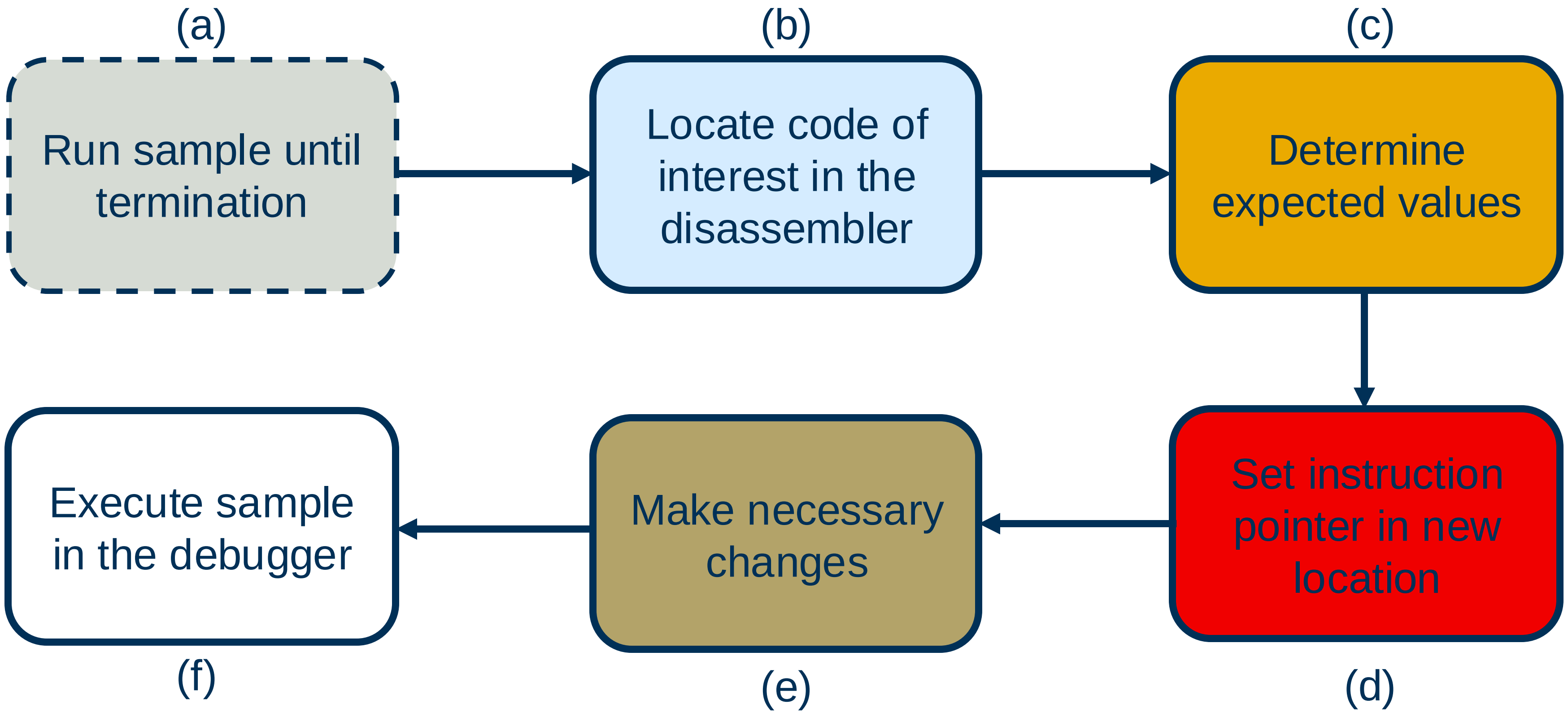
Forced Execution

Targeted Execution

Unpacking

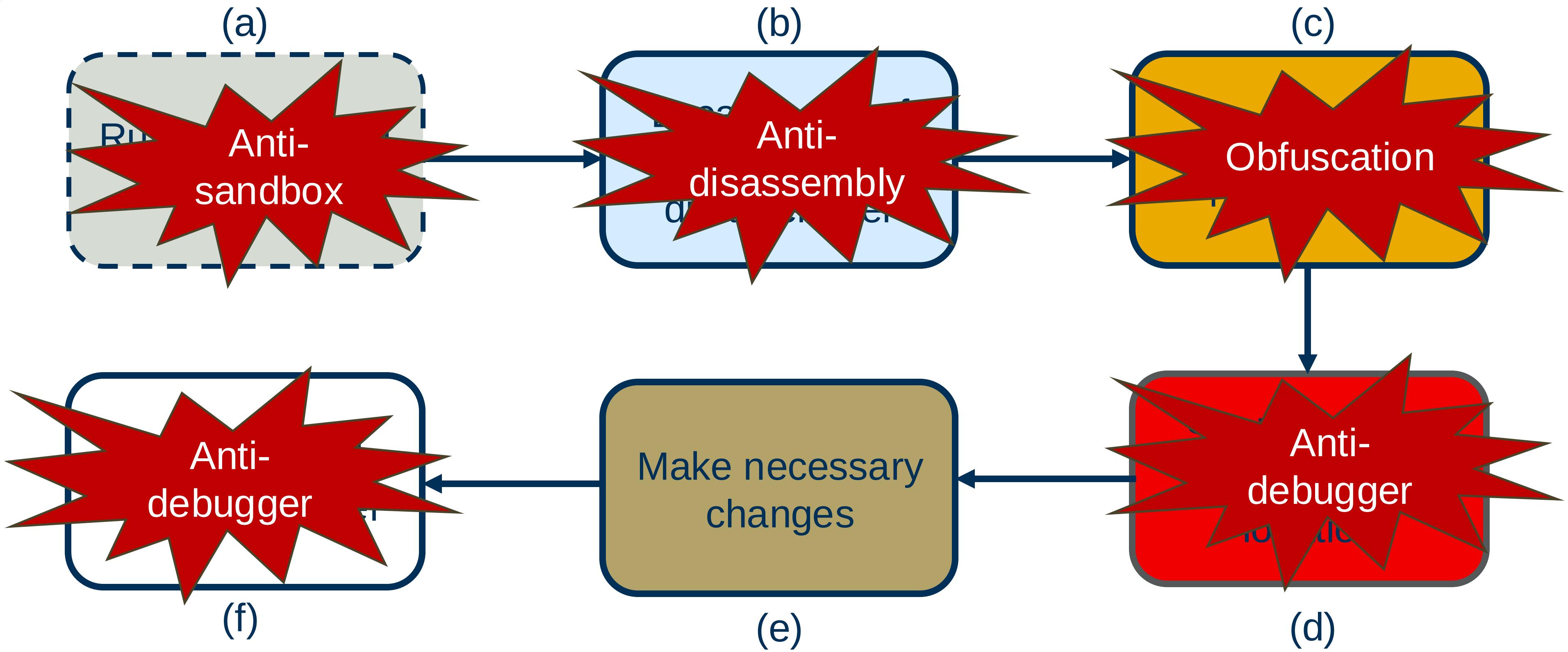
Targeted Execution Workflows

Ex: Decryption algorithm or suspicious behavior


(a)
Run sample until termination
(b)
Locate code of interest in the disassembler
(c)
Determine expected values
(d)
Set instruction pointer in new location
(e)
Make necessary changes
(f)
Execute sample in the debugger

Targeted Execution Workflows

Ex: Decryption algorithm or suspicious behavior



Existing Research Solutions

Has academic research tackled these challenges before?

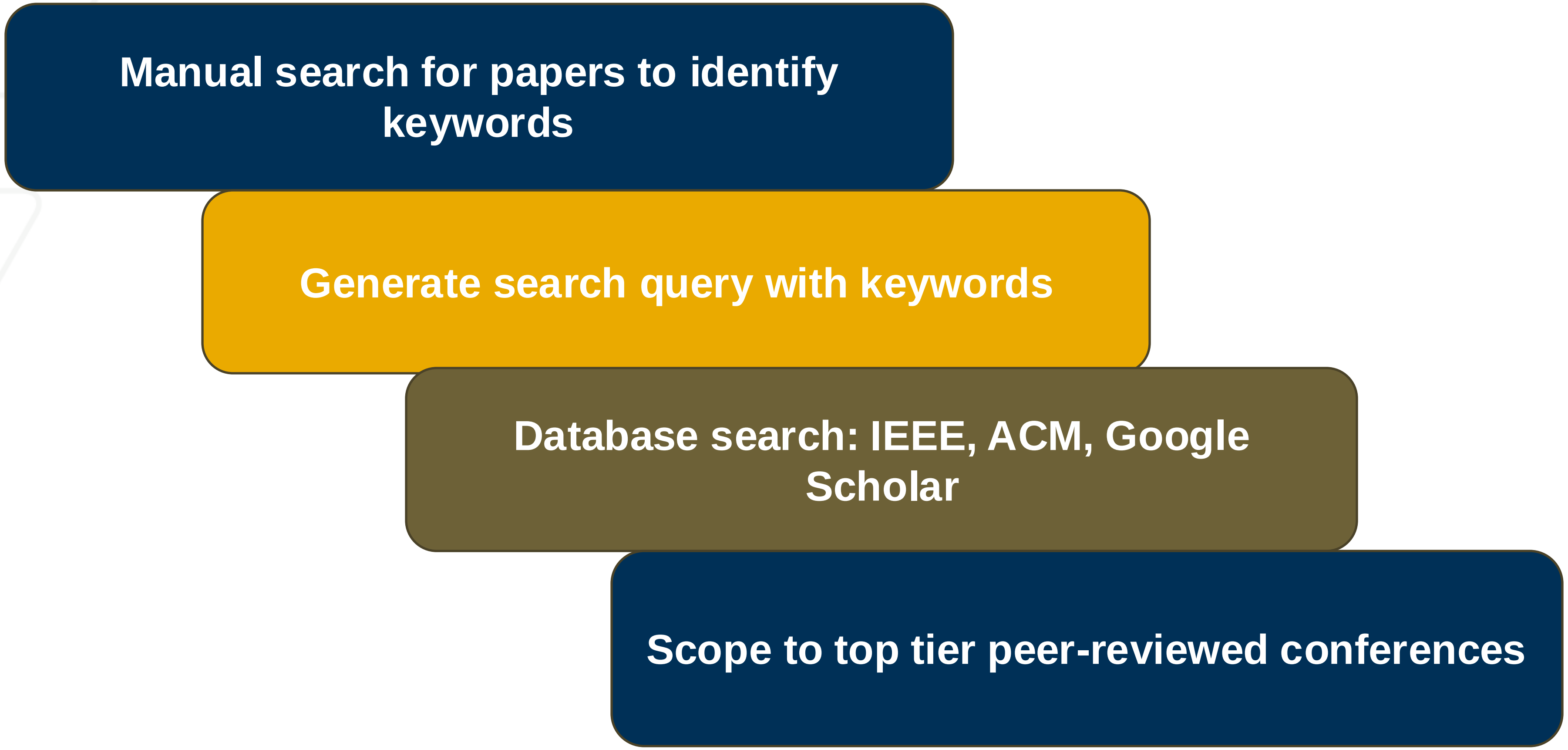
Yes!

Systematic Mapping

Which evasion techniques has the research community historically focused on?

Systematic Mapping Methodology

Followed recommendations from Dr. Peterson’s [1] and Dr. Kitchenham’s [2] on systematic mapping



Manual search for papers to identify keywords

Generate search query with keywords

Database search: IEEE, ACM, Google Scholar

Scope to top tier peer-reviewed conferences

[1] Petersen, Kai, Sairam Vakkalanka, and Ludwik Kuzniarz. "Guidelines for conducting systematic mapping studies in software engineering: An update." *Information and Software Technology* 64, 2015

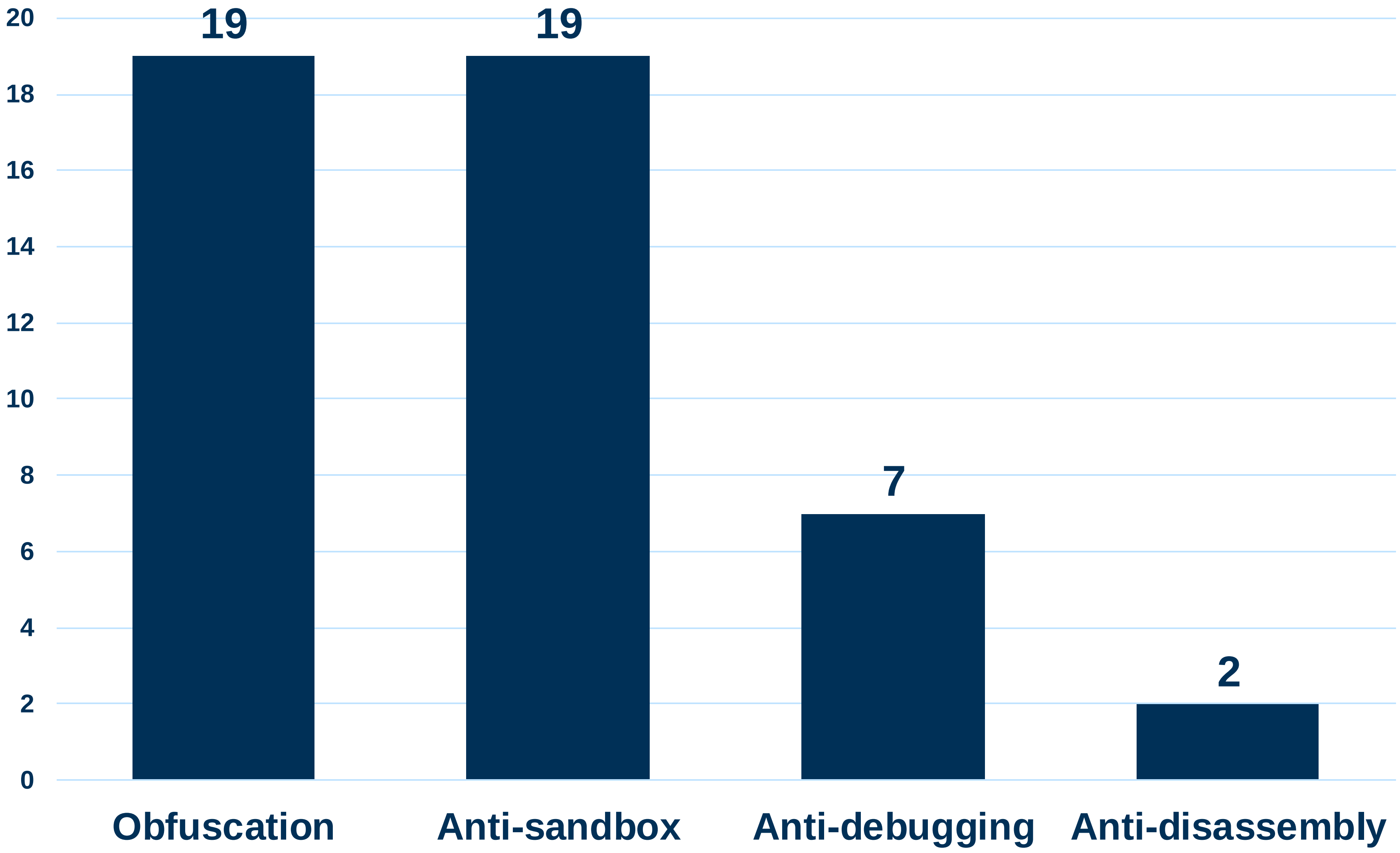
[2] Kitchenham, Barbara Ann, David Budgen, and Pearl Brereton. *Evidence-based software engineering and systematic reviews*. Vol. 4. CRC press, 2015.

Criteria for Papers

- **Inclusion Criteria (Title and Abstract)**
  - Must reference malware
  - Excludes mobile or IoT malware.
  - Not a survey or a measurement study.
  - References dynamic malware analysis, deobfuscation, unpacking, or disassembly
- **Exclusion Criteria (Full Text)**
  - Excludes research that does not directly help counter evasion techniques or provide alternative methods for analysis.
- **Search Evaluation**
  - Refine search query
  - 82.4% retrieval rate

45 Research Papers

Systematic Mapping Results

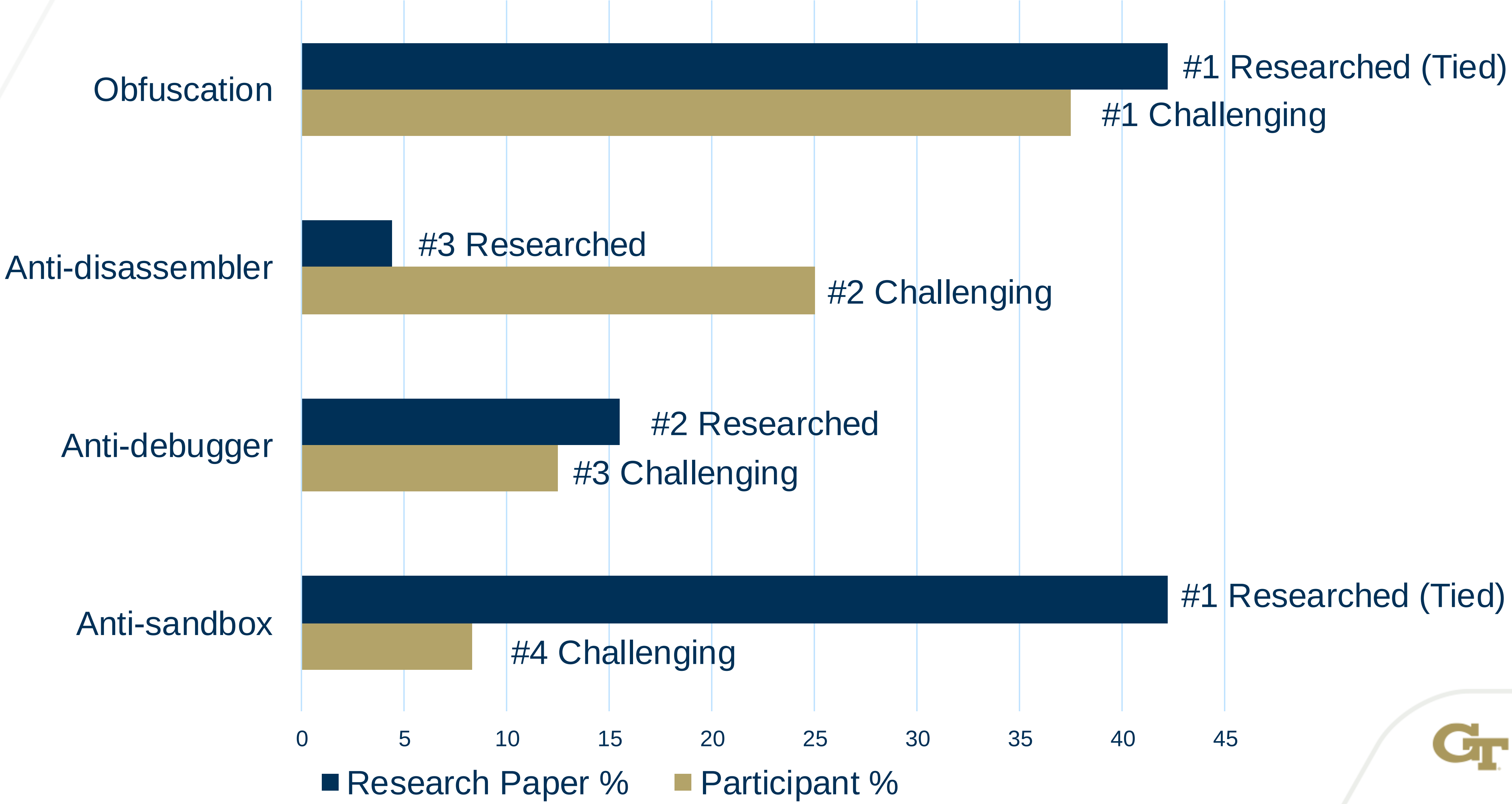


Comparative Analysis

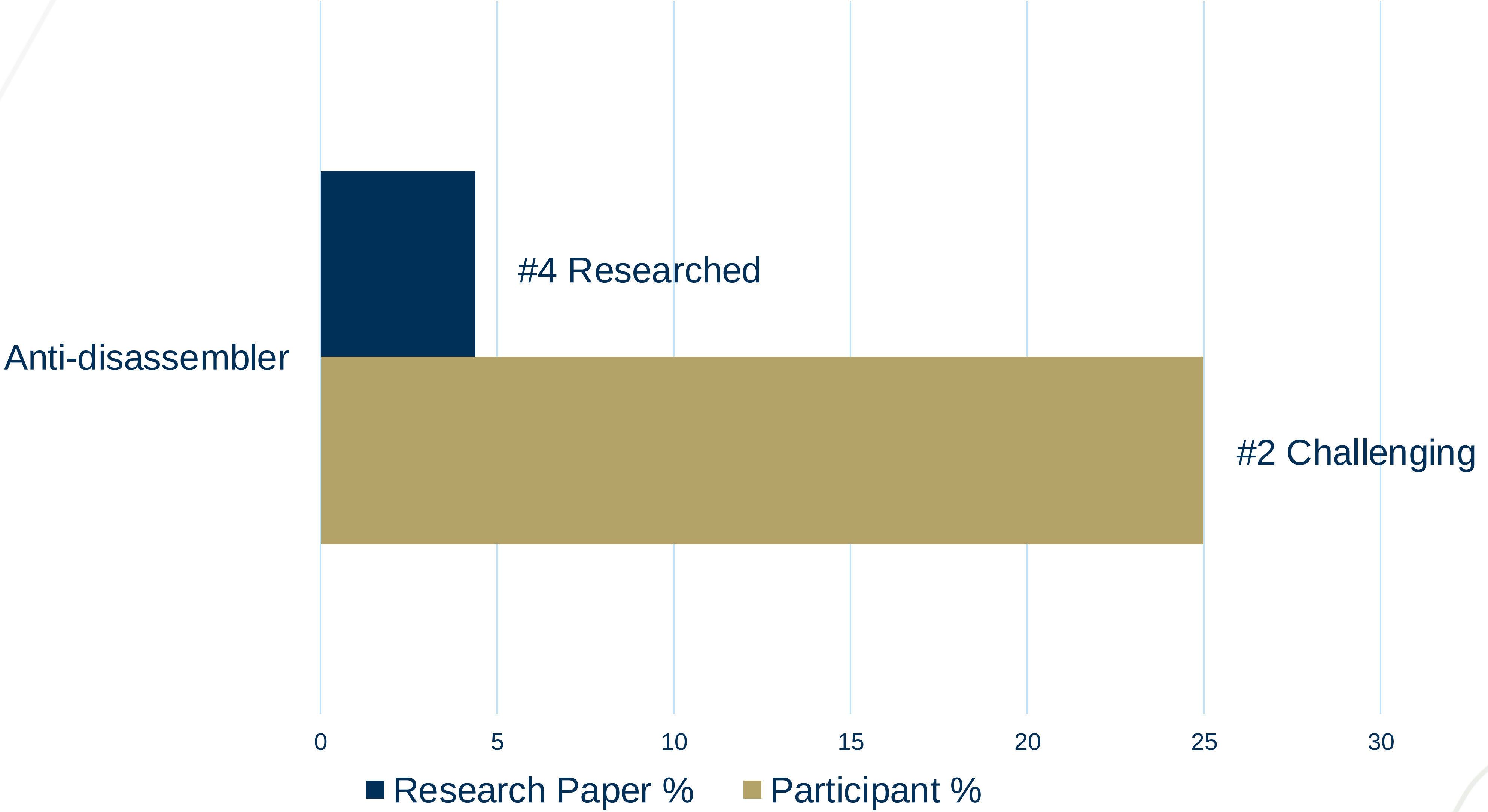
Historical Research Focus on Evasion Techniques

Current Evasion Challenges in Practice

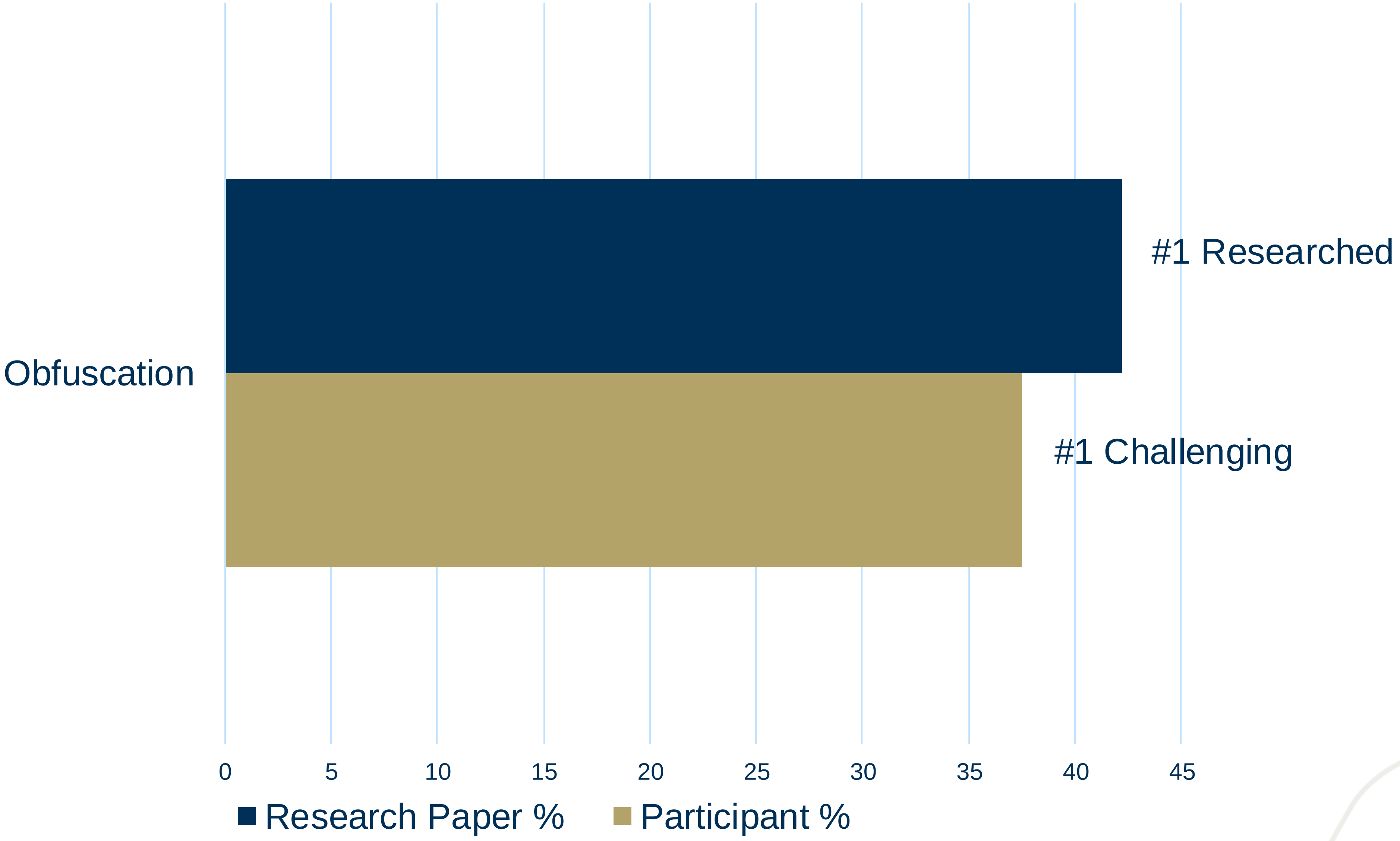
Comparing Malware Analysts Challenges with Research Contributions



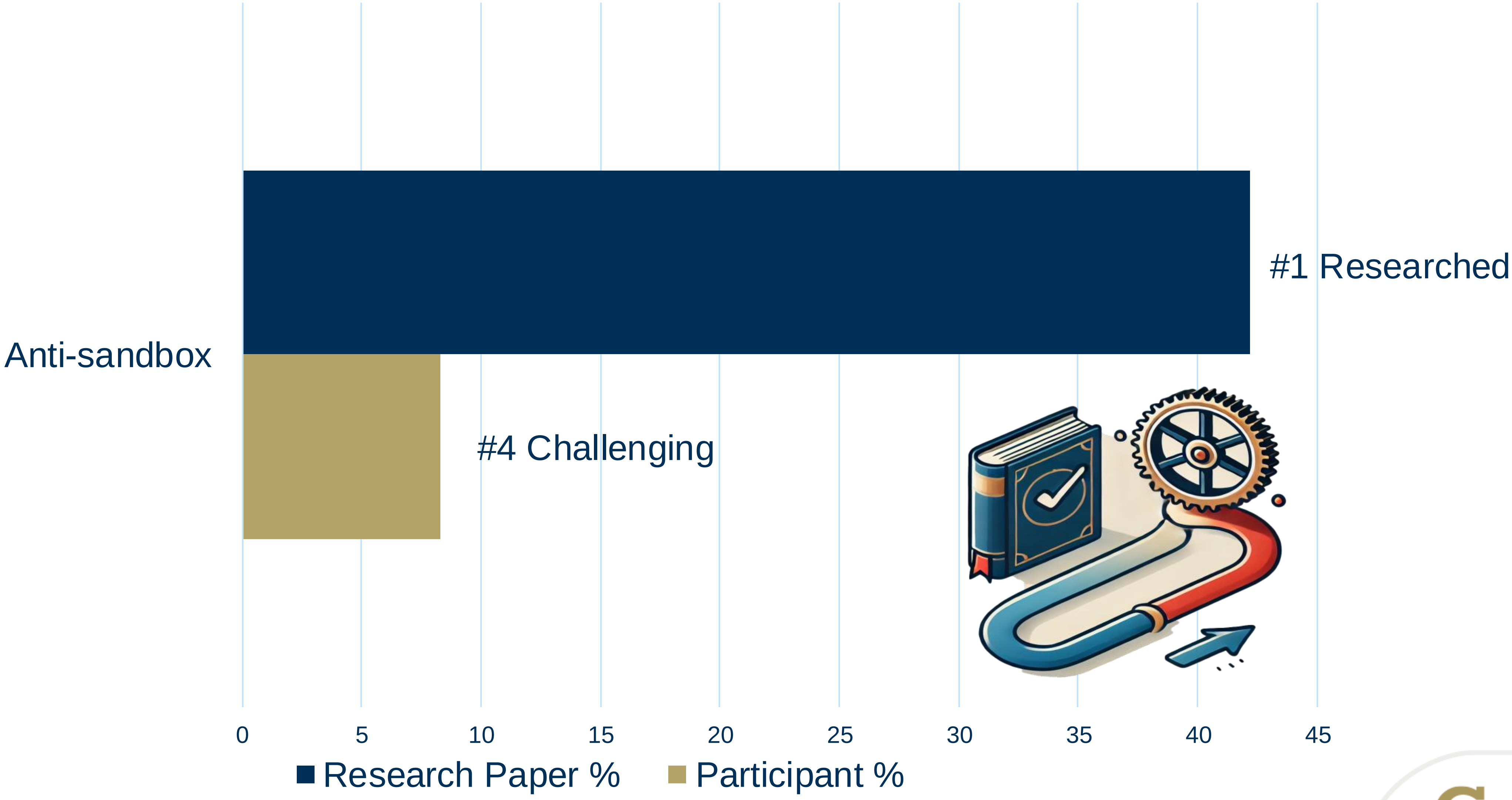
Comparing Malware Analysts Challenges with Research Contributions



Comparing Malware Analysts Challenges with Research contributions

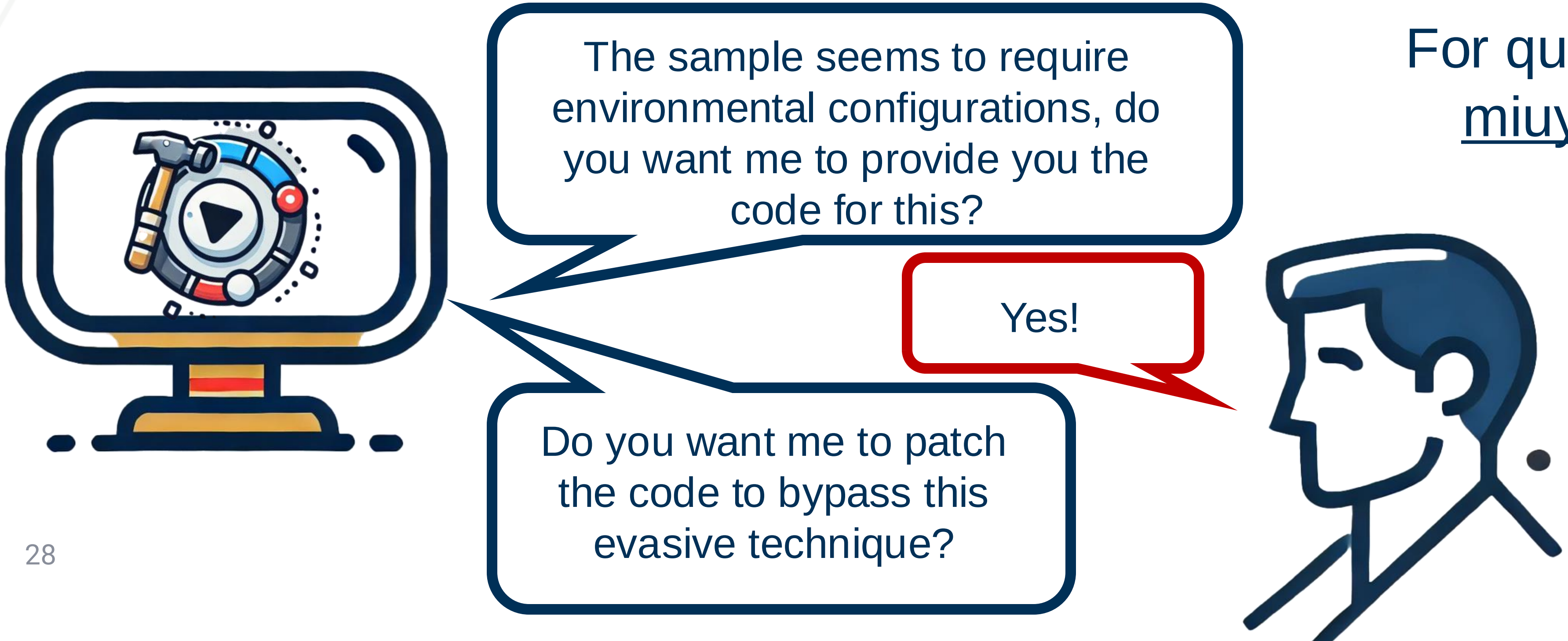


Comparing Malware Analysts Challenges with Research Contributions



Takeaways and Next Directions

- Focus on human analysts’ needs to identify critical and under-researched topics
  - Prioritize anti-disassembly research to address a major challenge for analysts
  - Overcome barriers to adoption solutions like obfuscation
- Designing tools with human analysts in mind



The sample seems to require environmental configurations, do you want me to provide you the code for this?

Yes!

Do you want me to patch the code to bypass this evasive technique?

Thank you!

For questions please contact:
miuyinyong@gatech.edu