ODMWS — WS 2022

Seminar

Maximilian Tschirschnitz

Lehrstuhl für Sicherheit in der Informatik / I20
Prof. Dr. Claudia Eckert
Technische Universität München

December 13, 2022

Review - Structure

1. Summary
2. Strength
3. Weakness
4. Comments

Review - Structure

1. Summary
2. Strength
3. Weakness
4. Comments

- $\sim$ ***About one page***

Summary

- Shortly summarize the contents
- What is the topic?
- What is new about it?
- Which new approaches were developped?
- $\sim$ *About one paragraph*

The paper introduces an alternative to the traditional Unix ptrace facility and an implementation of this alternative as a Linux loadable kernel module and a GDB server interfacing to the module. The alternative, called plutonium-dbg, is meant to address some shortcomings of ptrace, but ends up having its own different shortcomings, which the paper acknowledges and discusses.

Overall, this is a fine topic for research, a practically usable implementation, and a decent paper. There's no reason not to accept this paper.

Why Do I Have to Summerize the Paper?

- Helps you understand the paper
- Shows the author that you understood the paper
- **Or that you did not!**
- May show the reason for good or bad feedback
- Helps to detect what is unclear
- Shows the perspective of the reader

Strength

- Highlight the strength of the paper
- What did you like?
- Content related
- Focus on outstanding points

Strengths:
+ The paper is well written and easy to follow and the authors guide the reader through the different challenges
+ They address an important problem and they provide a solution. We as a community lack of specific tools for reversing on Linux compared to all the available programs for Windows
+ This solution is released on Github
+ They provide a really nice summary of all the known techniques to detect a debugger on Linux. This is the most completed list I know
+ The tool is compatible with the GDB protocol
+ Nice use of the kprobes and uprobes for addressing all the technical challenges

Weakness

- Highlight the Weakness of the paper
- What did you dislike?
- Content related
- Focus on issues that teach the author something
- Do **not** list spelling or grammar errors

Weaknesses:
- The tool supports only x86_64 binaries
- They insist plutonium-dbg is important for malware analysis but as Cozzi et al.[1] pointed out x86_64 is not the most common architecture for Linux malware.
- Maybe there are other instructions or events that as a side effect push on the stack RFLAGS so the solution proposed for the pushf is not solid
- The authors do not discuss at all how common are the mentioned evasive tricks. Table XIII of [1] gives an idea.
- The evaluation part does not mention which binaries were used for the experiments. Real malware samples? Quick pocs? Please specify.

Comments

- Text-related comments
- Target to improve the document
- Specific suggestions for improvement

Comments:
After headline of 1 and 3: A short introduction to the chapter would be good.
In 1.3: The groups should be mentioned. No detailed explanation needed, a short enumeration is enough.
In 2: A brief overview about the content of LO! and CONFUSE would help. It is unclear how exactly they are related to this paper and what their results are.
In 4.2: "Opaque Predicates are boolean functions that always return a fixed value regardless of their input." Example of missing citation mentioned above. Who introduces opaque predicates? Where is this definition taken from?
In 5.2: "This is a FunctionPass." It is unclear what is referred to by "this"
In 5.3: Typo at "after the second variable and insert the perdicate". "perdicate" should be changed to "predicate".

Your Task

- **Carefully** read through the two papers we provide you with
- Write reviews for both (about one page each)
- Hand in until **Dec, 20th** latest!
- Improve your draft based on reviews you receive afterwards

Don't be shy

- We do not tell you who is the author (you might know)
- We do not tell the author who has written the review

Writing an Abstract

Content

Your Abstract is a short version of your paper. It should reflect all core ideas and results.

- Motivation/Problem (2 - 3 sentences)
  How is your problem/solution relevant. What is the problem?
- Solution/Analysis Approach (second largest part)
  Which approach/technique was used to address this problem.
- Results
  Most important part.
- Conclusion
  What follows from your results/consequences?

Writing an Abstract

Style

- Easy to comprehend/follow for a broad audience.
- Closed in itself and self-explaining.
- Be honest and present limitations and problems truthfully.

Writing an Abstract

Format

- Between 150 and 200 words.
- No citations/references (usually).
- Concrete language (avoid words like ‘mostly’, ‘relatively’, etc.).
- Only content from your work.

Writing an Abstract

Not an Introduction

- Covers the whole paper.
- Introduction leads towards the remainder of the work.
- Introduction can be a bit more ‘scenic’.
- Abstract needs to get your point delivered in a snap (advertisement).

Further Reading:

- `https://users.ece.cmu.edu/~koopman/essays/abstract.html`
- `http://www.adelaide.edu.au/writingcentre/learning_guides/learningGuide_writingAnAbstract.pdf`
- `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3136027/`

Questions?