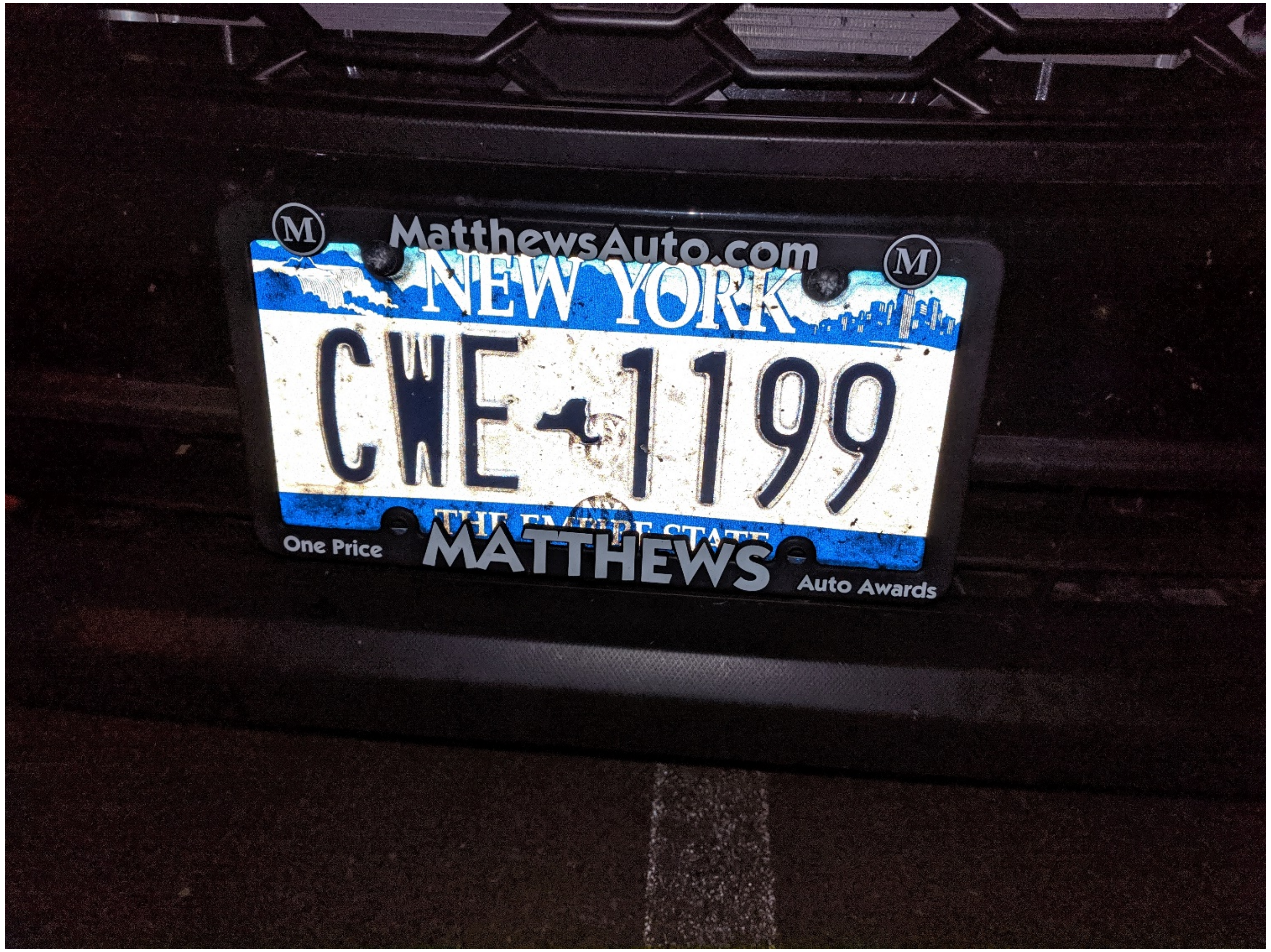
GrammaTech Experience with SATE 2019

Paul Anderson, VP of Engineering

paul@grammatech.com

A Funny Thing Happened on the Way to the Forum…



Overview

- CodeSonar Introduction
- Results on SQLite
- Results on Wireshark
- Conclusions

CodeSonar Introduction

- Deep Static Analysis of C and C++
  - Also binary – x86, x64, ARM, (PowerPC, MIPS in development)
  - Other languages by integration with other tools
- Whole Program, Path Sensitive, “Symbolic”, Taint-tracking
  - Also: precise parsing, incremental, concurrent
- Over 300 warning classes (including Misra)
  - Sweet spot is undefined behavior and API misuse
- Most customers are safety-critical embedded
  - Default configuration reflects this bias
  - *Recall* dominates *precision*

Results on SQLite

- Injected bugs were instances of 5 scions
- CodeSonar found all bugs of 4/5 scions
- Found none of the bugs of the other scion

- 26 of the 30 instances involved **sqlite3PendingByte**
  - A file static variable, hence global state

- Conclusions:
  - Injected bugs biased towards a small part of the state of the program
  - More variation within a scion would be helpful

Results on Wireshark

- CodeSonar found CVE-related defects, and missed some
- Many “new” bugs identified
  - Code not written to “embedded safety-critical” standards
  - Lots of assumptions that inputs are well formed
  - Lots of copy & paste
- Analysis would benefit from configuration tuning
  - E.g, turn on taint analysis
- Internal APIs that should be modeled
  - ep_alloc(), se_alloc()
    - Like malloc/free API
  - tvbuff.c
    - Buffered I/O library
    - E.g., tvb_get_ephemeral_string() would benefit from taint annotation

Conclusions

- Tuning of configuration is important to get good results
  - Factory settings are rarely optimal
- Modeling/hints for key libraries
  - Allocators
  - Taint sources & sinks
  - Some familiarity with the code base is best

Feedback for SATE

- Thanks for organizing this!
  - Your hard work is greatly appreciated

- Small examples of real programs are far better than micro tests
- Maybe vary the application domain?
  - It would be good to have a real embedded app
  - Although toolchain challenges may exist

Questions?

Thank you!

- My contact info:
  - Paul Anderson
  - paul@grammatech.com
  - http://www.grammatech.com