Using AI to Build More Secure Software

Dr. Mark Sherman

Technical Director, Cyber Security Foundations

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0415

The SEI is a DoD R&D Federally Funded Research and Development Center

Established in 1984 at Carnegie Mellon University

~650 employees (ft + pt), of whom about 70% are engaged in technical work

Initiated CERT cybersecurity program in 1988

Offices in Pittsburgh and DC, with several locations near customer facilities

~$140M in annual funding

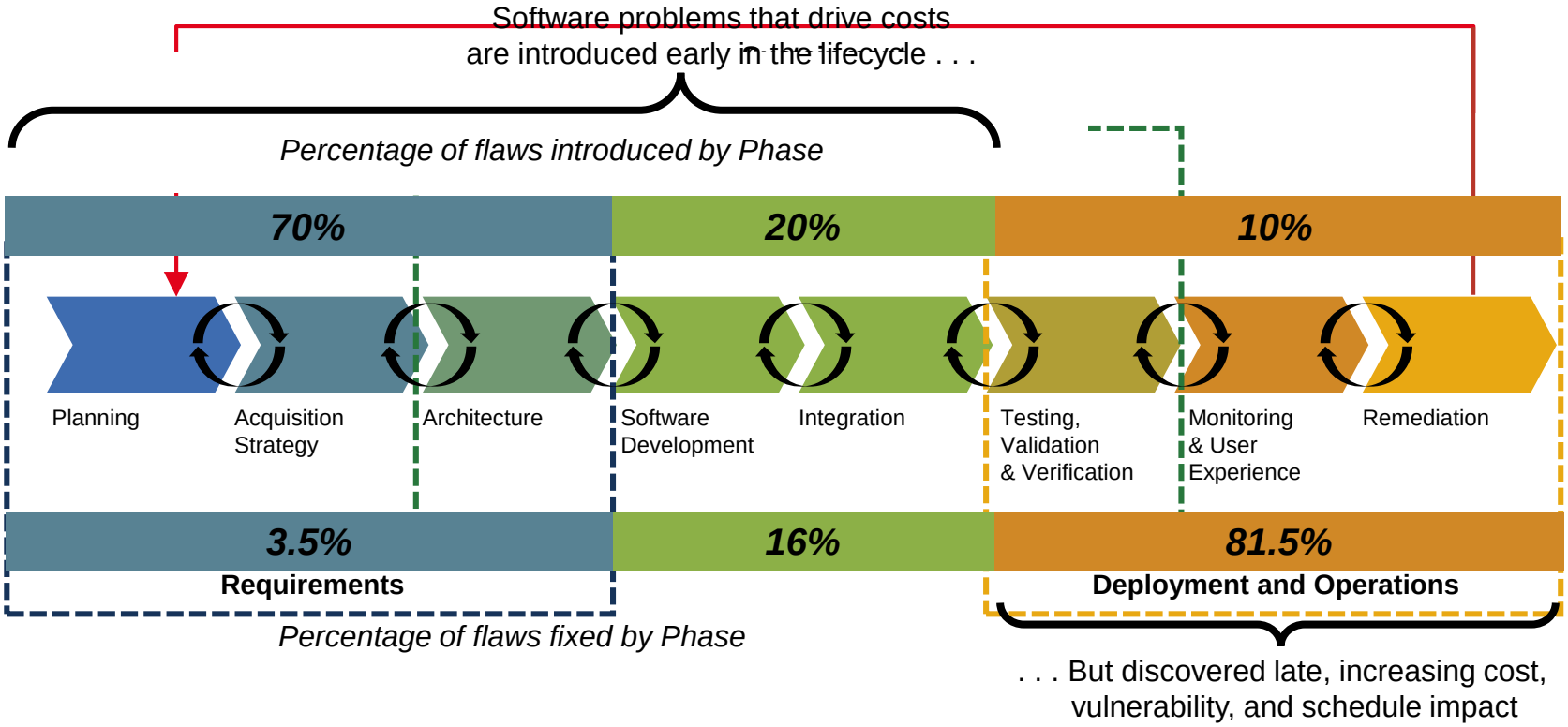
Software Cost and Vulnerability Threaten Military Capability

Finding and fixing software problems late in the acquisition lifecycle drives up cost and delays delivery

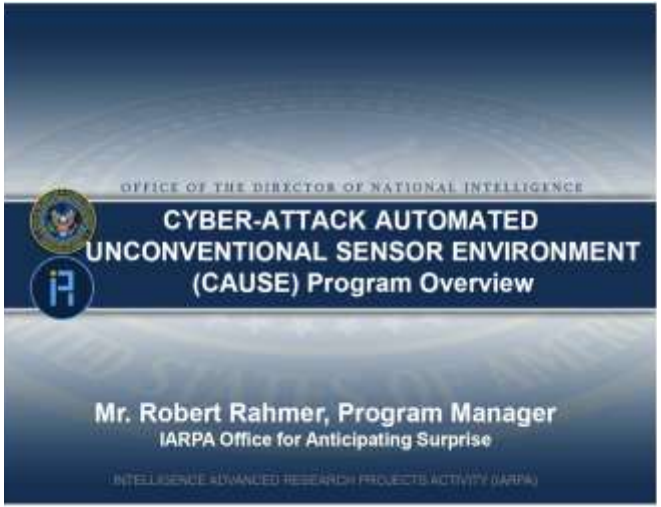
Latent cyber vulnerabilities and those exposed during operations or due to underlying dependencies put missions at risk

Statistically, a 10M LOC Weapons Platform written in C will be delivered with 280 – 1,400 exploitable vulnerabilities

Fixing Problems Late Drives Costs, Delays Deployment



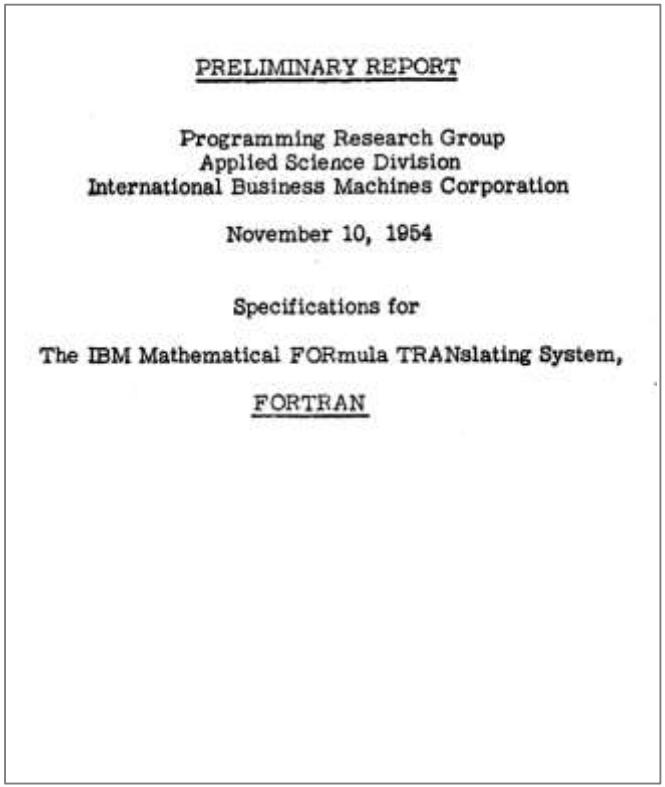
Predicting Threats – IARPA CAUSE



Sources: IARPA, Cyber-attack Automated Unconventional Sensor Environment (CAUSE), https://www.iarpa.gov/index.php/research-programs/cause https://www.iarpa.gov/images/files/programs/cause/CAUSE_Proposers_Day_Briefing.pdf

- Identify and evaluate unconventional and technical indicators in the earlier phases of cyber attacks that are leading indicators of later stages of the attack.
- Create highly efficient algorithms that will process massive data streams from diverse data sets to extract signals from noisy data.
- Create techniques to fuse traditional technical indicator sensor data and alternate unconventional indicator data sources to develop automated probabilistic warnings.
- Identify and evaluate techniques that enable sharing of disparate threat contextual information and indicators among multiple organizations and security professionals to forecast an attack.

AI in Automatic Programming – The Beginning



PRELIMINARY REPORT

Programming Research Group
Applied Science Division
International Business Machines Corporation

November 10, 1954

Specifications for

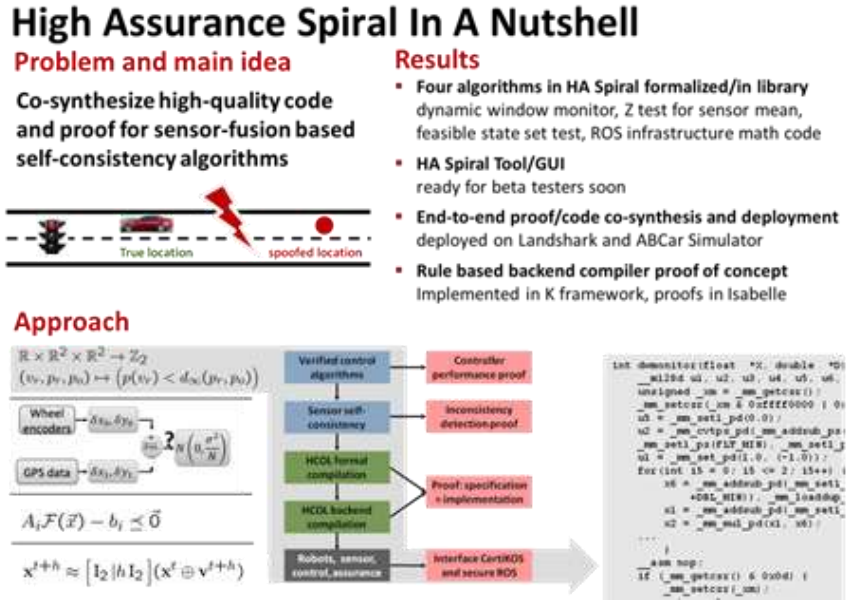
The IBM Mathematical FORmula TRANslating System,

FORTRAN

"The IBM Mathematical Formula Translating System or briefly, FORTRAN, will comprise a large set of programs to enable the IBM 704 to accept a concise formulation of a problem in terms of a mathematical notation and to produce automatically a high speed 704 program for the solution of the problem."

Source: J.W. Backus, H. Herrick and I. Ziller,
https://archive.computerhistory.org/resources/text/Fortran/102679231.05.01.acc.pdf

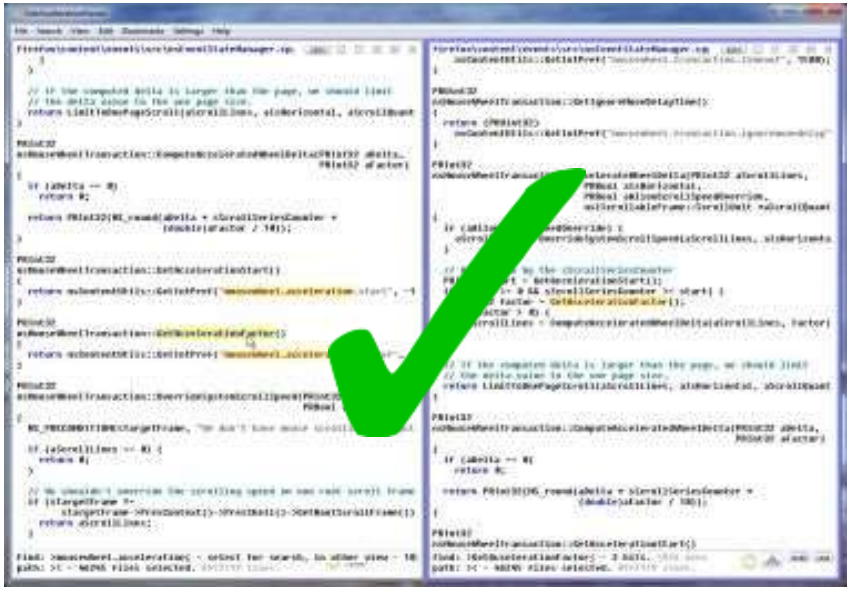
Generating Coded thru Search – High Assurance SPIRAL



"*High Assurance SPIRAL* aims to solve the last mile problem for the synthesis of high assurance implementations of controllers for vehicular systems that are executed in todays and future embedded and high performance embedded system processors."

Sources: Franz Franchetti, José M. F. Moura, Manuela Veloso, Andre Platzer, Soummya Kar, David Padua, Jeremy Johnson, Mike Franusich, High Assurance Spiral: Scalable and Performance Portable Domain-Specific Control System Synthesis, https://users.ece.cmu.edu/~franzf/hacms.htm; http://www.spiral.net/

Finding Programming Vulnerabilities – Source Code as Natural Language



Analyze Source Code for Insecure Coding

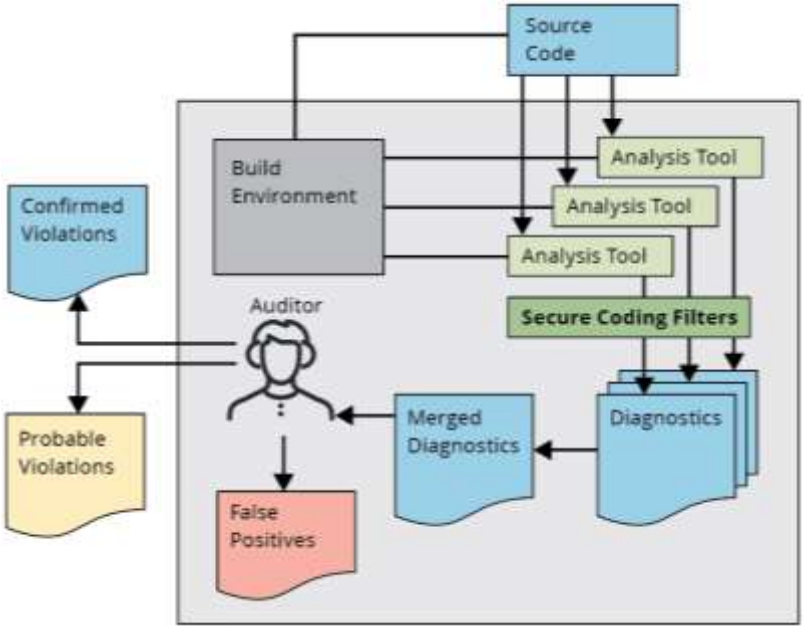
- Supplements Compiler-style Checking
- Treats Programs Like Natural Language

Sources: Carson D. Sestili, William S. Snavely, Nathan M. VanHoudnos, Towards security defect prediction with AI, Sep 12, 2018, https://arxiv.org/abs/1808.09897

Song Wang, Taiyue Liu, and Lin Tan. 2016. Automatically learning semantic features for defect prediction. In *Proceedings of the 38th International Conference on Software Engineering* (ICSE '16). ACM, New York, NY, USA, 297-308. DOI: https://doi.org/10.1145/2884781.2884804

Uri Alon, Meital Zilberstein, Omer Levy, and Eran Yahav. 2019. code2vec: learning distributed representations of code. Proc. ACM Program. Lang. 3, POPL, Article 40 (January 2019), 29 pages. DOI: https://doi.org/10.1145/3290353

Combining Multiple Tools With AI To Find Source Code Flaws – SCALe



Using AI to Combine Tool and Environmental Data

- Multiple static code analyzers
- Multiple environmental features
- Multiple classification techniques

Source: Lori Flynn, SCALe: A Tool for Managing Output from Static Analysis Tools, Sept 24, 2018, https://insights.sei.cmu.edu/sei_blog/2018/09/scale-a-tool-for-managing-output-from-static-code-analyzers.html ; Lori Flynn, Automating Static Analysis Alert Handling with Machine Learning, MIT Lincoln Labs Cyber Security, Exploitation and Operations Workshop, June 19, 2018

Theorem Proving for Secure Software – DARPA High-Assurance Cyber Military Systems (HACMS)

Key HACMS technologies include interactive software synthesis systems, verification tools such as theorem provers and model checkers, and specification languages. Recent fundamental advances in the formal methods community, including advances in satisfiability (SAT) and satisfiability modulo theories (SMT) solvers, separation logic, theorem provers, model checkers, domain-specific languages and code synthesis engines suggest that this approach is feasible.

Source: Raymond Richards, DARPA, High-Assurance Cyber Military Systems (HACMS), https://www.darpa.mil/program/high-assurance-cyber-military-systems

Using AI to Drive Test Inputs – Fuzzing

"Fuzzing:" Generating and Testing Random Inputs

Original: Random or Deterministic

Now: Use AI to Guide Generation of Sample Inputs

Sources: Allen Householder, Announcing CERT Basic Fuzzing Framework Version 2.8, Oct. 5, 2016, https://insights.sei.cmu.edu/cert/2016/10/announcing-cert-basic-fuzzing-framework-bff-28.html

Guanhua Yan ; Junchen Lu ; Zhan Shu ; Yunus Kucuk, ExploitMeter: Combining Fuzzing with Machine Learning for Automated Evaluation of Software Exploitability, 2017 IEEE Symposium on Privacy-Aware Computing (PAC), 1-4 Aug. 2017, https://doi.org/10.1109/PAC.2017.10

Dongdong She, Kexin Pei, Dave Epstein, Junfeng Yang, Baishakhi Ray, Suman Jana, NEUZZ: Efficient Fuzzing with Neural Program Smoothing, To appear in the 40th IEEE Symposium on Security and Privacy, May 20--22, 2019, San Francisco, CA, USA

Automated Program Repair – DARPA Cyber Grand Challenge

"Mayhem" demonstrated automated cyber defense

- Detect attack on program
- Analyze changes to program
- Deploy updated software

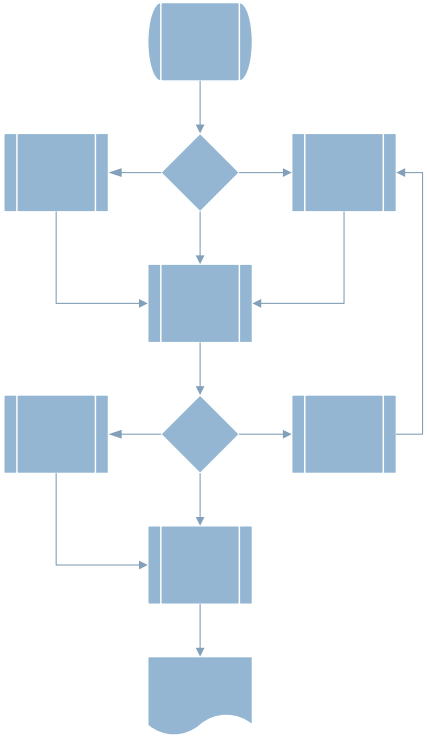
Source: DARPA, "Mayhem" Declared Preliminary Winner of Historic Cyber Grand Challenge, Aug 4, 2016, https://www.darpa.mil/news-events/2016-08-04

AI Supporting Judgement – IBM Watson to Improve Assurance

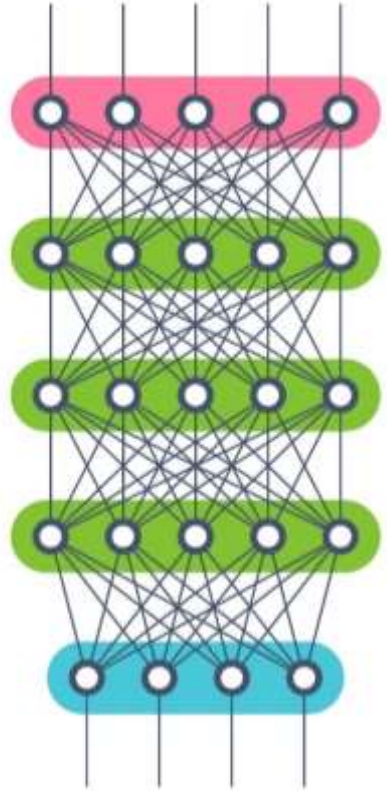
- Acquisition programs generate voluminous documentation
- Assurance is based on assembling and reviewing relevant evidence from documents
- Finding appropriate evidence or explanations can be challenging
- SEI Proof of Concept

Source: Mark, Sherman, Verifying Software Assurance with IBM's Watson, https://www.youtube.com/watch?v=aW3497xhypY, Sep 11, 2017

Machine Learning is a Different Style of Programming



VS



AI Attacks Are Different

Pixel Manipulation

Feature Differentiation

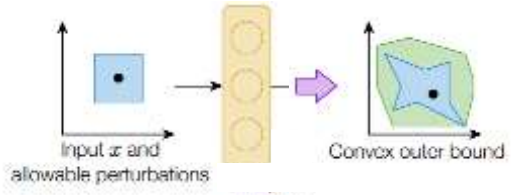
Source: Athalye, A., Engstrom, L., Ilyas, A., & Kwok, K. (2017, July 24). *Synthesizing Robust Adversarial Examples*. *arXiv [cs.CV]*. Retrieved from http://arxiv.org/abs/1707.07397

Source: Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (CCS '16). ACM, New York, NY, USA, 1528-1540. DOI: https://doi.org/10.1145/2976749.2978392

Some Technical Approaches for Defending AI Systems

Training Defenses

Wong & Kolter (2017)
output bound



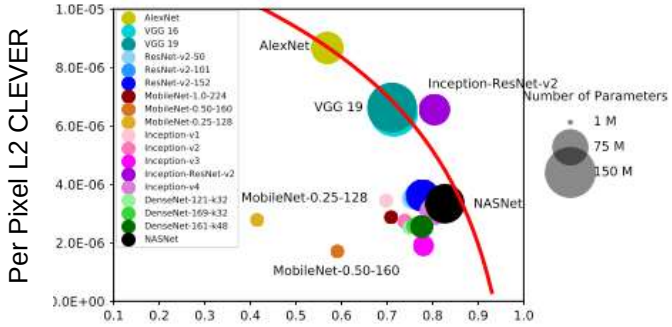
Causal Defenses

Tsipras et al. (2018)
adversarial data augmentation

Turtle → Bird

Engineering Defenses

Su et al. (2018) empirically demonstrates robustness/accuracy trade off in ImageNet models



Source: Wong, E., & Kolter, J. Z. (2017). Provable defenses against adversarial examples via the convex outer adversarial polytope. ArXiv:1711.00851 [Cs, Math]. Retrieved from http://arxiv.org/abs/1711.00851; Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2018). Robustness May Be at Odds with Accuracy. ArXiv:1805.12152 [Cs, Stat]. Retrieved from http://arxiv.org/abs/1805.12152; Su, D., Zhang, H., Chen, H., Yi, J., Chen, P.-Y., & Gao, Y. (2018). Is Robustness the Cost of Accuracy? – A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. ArXiv:1808.01688 [Cs]. Retrieved from http://arxiv.org/abs/1808.01688; Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. 2011. Adversarial machine learning. In Proceedings of the 4th ACM workshop on Security and artificial intelligence (AISec '11). ACM, New York, NY, USA, 43-58. DOI=http://dx.doi.org/10.1145/2046684.2046692

Summary: Using AI to Build More Secure Software

Problem: The Need to Build Secure Software

Threat Analysis: What To Protect Against

Code Development: Assisting Programmers to Build More Secure Software

Building AI Systems Securely: Next Generation of Software Face New Attacks

Contact Us

Carnegie Mellon University

Software Engineering Institute

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

412-268-5800

888-201-4479

info@sei.cmu.edu

www.sei.cmu.edu