Using AI to Build More Secure Software

Dr. Mark Sherman
Technical Director
Cyber Security Foundations

Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213
Jan 21, 2020

Global Big Data Conference

www.globalbigdataconference.com

Twitter : @bigdataconf

#GlobalAI

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.

DM20-0030

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

Established in 1984 at Carnegie Mellon University

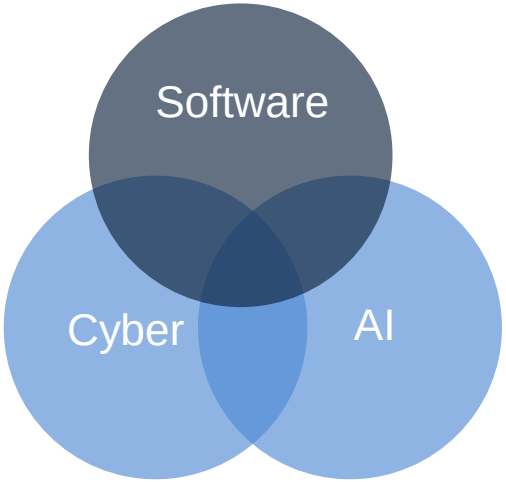
~650 employees (ft + pt), of whom about 70% are engaged in technical work

Initiated CERT cybersecurity program in 1988: the birthplace of cybersecurity

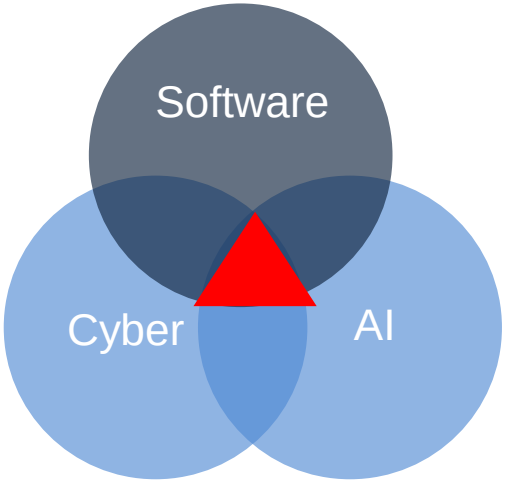
Offices in Pittsburgh and DC, with several locations near customer facilities

~$140M in annual funding

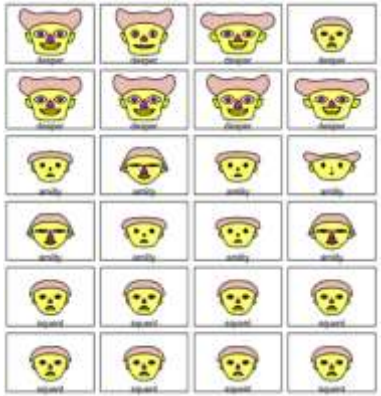
SEI Strategic Framework



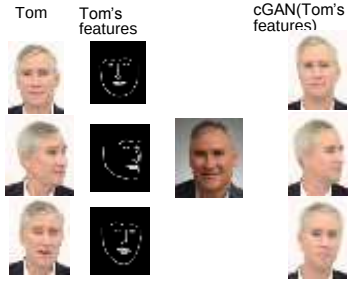
SEI Strategic Framework – Today's Focus



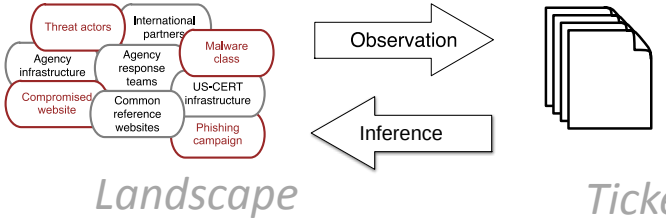
AI Is Playing an Increasing Role in Cybersecurity



Classifying Malware



Spotting Deep Fakes



Detecting Campaigns

- Detecting misinformation
- Spotting command-and-control paths
- Cyber training

- Technical debt detection
- Satellite image recognition
- Insider threat detection

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

F-15 Fighter Jet Hacked

Image: A U.S. Air Force F-15E Strike Eagle. (U.S. Air Force photo by Senior Airman Erin Trowe)

THE CYBERSECURITY 202: HACKERS JUST FOUND SERIOUS VULNERABILITIES IN A U.S. MILITARY FIGHTER JET

LAS VEGAS — In a Cosmopolitan hotel suite 16 stories above the Def Con cybersecurity conference this weekend, a team of highly vetted hackers tried to sabotage a vital flight system for a U.S. military fighter jet. And they succeeded.

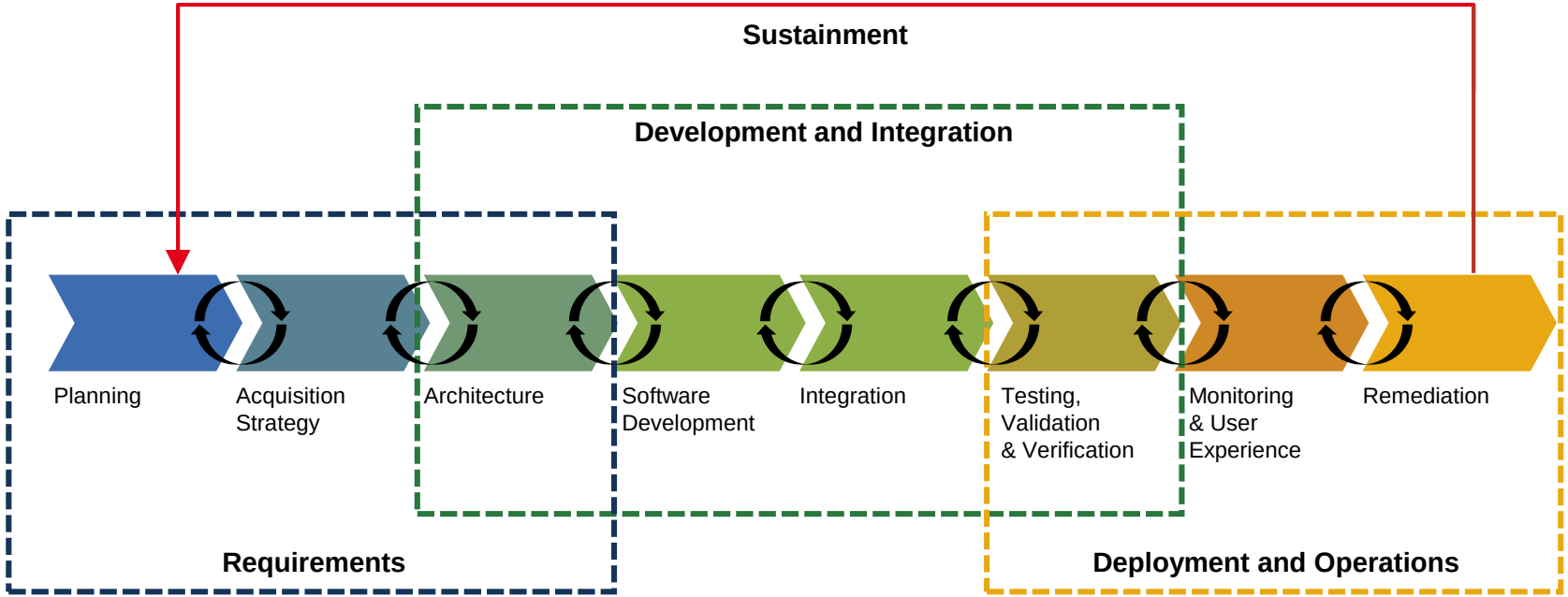
It was the first time outside researchers were allowed physical access to the critical F-15 system to search for weaknesses. And after two long days, the seven hackers found a mother lode of vulnerabilities that — if exploited in real life — could have completely shut down the Trusted Aircraft Information Download Station, which collects reams of data from video cameras and sensors while the jet is in flight.

Will Roper, a top U.S. Air Force acquisitions executive, told the Washington Post: "there are millions of lines of code that are in all of our aircraft and if there's one of them that's flawed, then a country that can't build a fighter to shoot down that aircraft might take it out with just a few keystrokes."

Joseph Marks, Aug 14, 2019

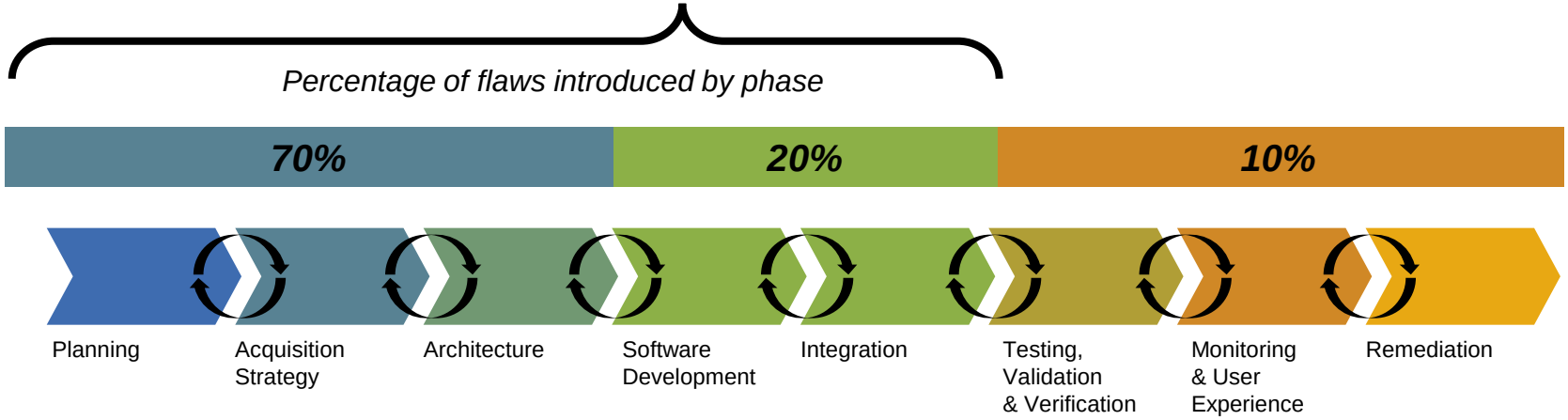
Source: https://https://www.washingtonpost.com/news/powerpost/paloma/the-cybersecurity-202/2019/08/14/the-cybersecurity-202-hackers-just-found-serious-vulnerabilities-in-a-u-s-military-fighter-jet/5d53111988e0fa79e5481f68/

Fixing Problems Late Drives Costs, Delays Deployment



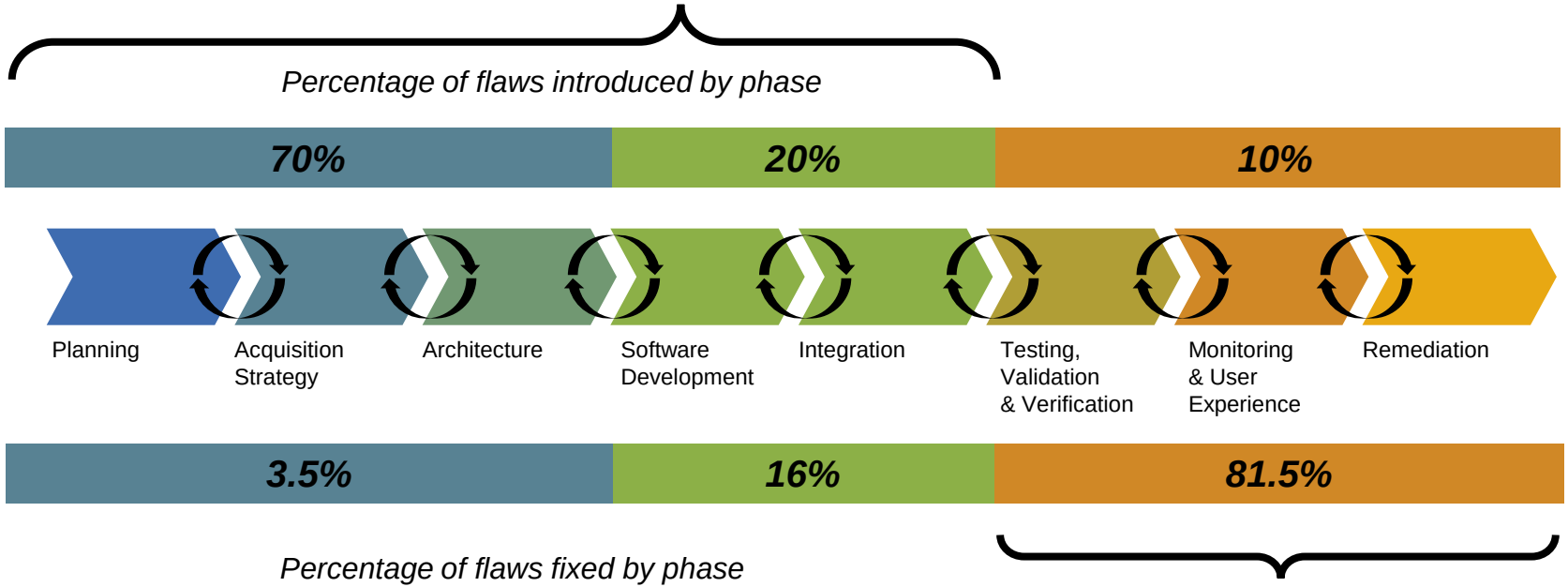
Fixing Problems Late Drives Costs, Delays Deployment

Software problems that drive costs
are introduced early in the lifecycle . . .



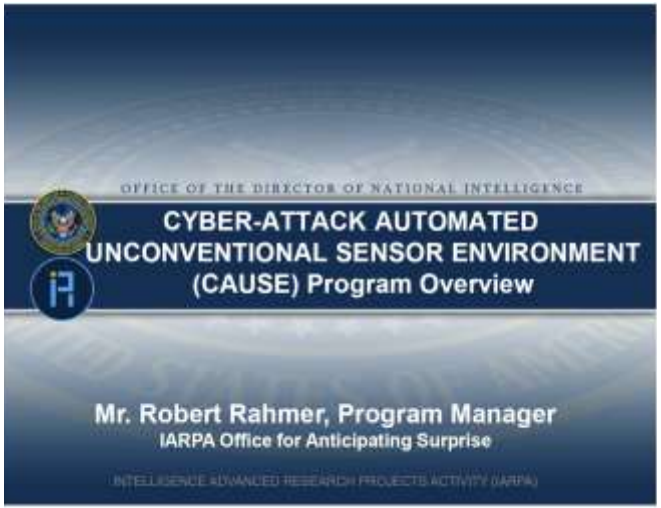
Fixing Problems Late Drives Costs, Delays Deployment

Software problems that drive costs
are introduced early in the lifecycle . . .



. . . But discovered late, increasing cost,
vulnerability, and schedule impact

Predicting Threats – IARPA CAUSE



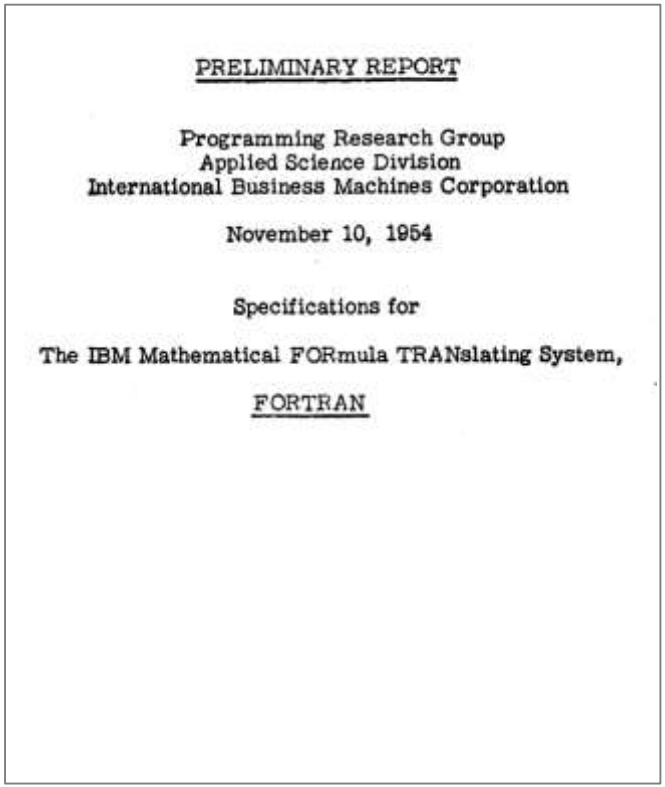
Sources: IARPA, Cyber-attack Automated Unconventional Sensor Environment (CAUSE), https://www.iarpa.gov/index.php/research-programs/cause https://www.iarpa.gov/images/files/programs/cause/CAUSE_Proposers_Day_Briefing.pdf

- 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

"The System will accept a concise formulation of a problem in terms of a mathematical notation and produce automatically a high speed program for the solution of the problem."

AI in Automatic Programming – The Beginning



PRELIMINARY REPORT

Programming Research Group
Applied Science Division
International Business Machines Corporation

November 10, 1954

Specifications for

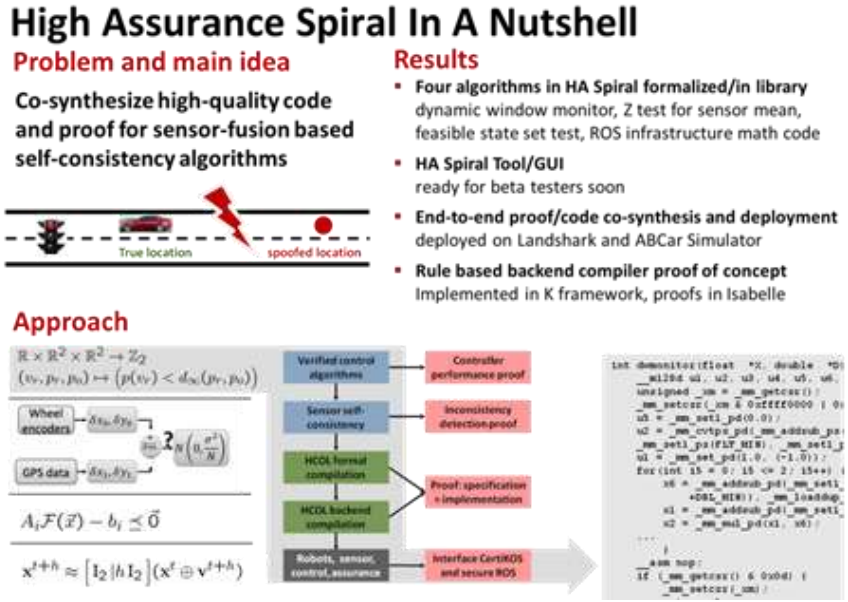
The IBM Mathematical FORmula TRANslating System,

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

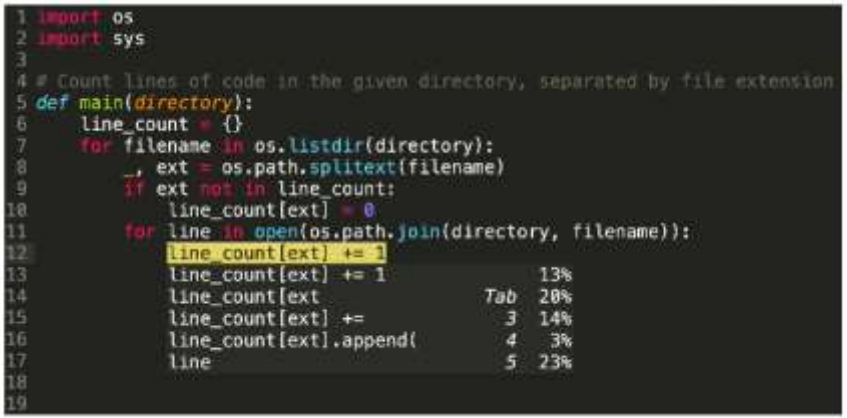
AI in Automatic Programming: Generating Coded Through Search – High Assurance SPIRAL



"*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/

Using AI for Autocompletion



Safe, correct code could be written incrementally

- Using n-grams
- Using deep learning (Generative Pretrained Transformer 2)

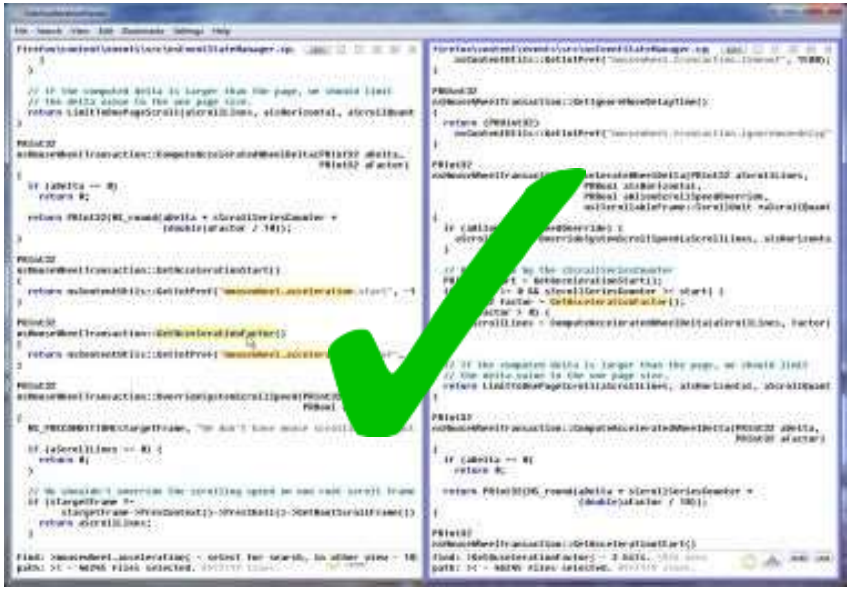
Sources:

E. Schutte, Autocomplete from StackOverflow, 2016, https://emilschutte.com/stackoverflow-autocomplete/

(Jacob Jackson) TabNine, “Autocompletion with deep learning,” July 18, 2019, https://tabnine.com/blog/deep

L. Tung, “New tool promises to turbo-charge coding in major programming languages,” July 25, 2019, https://www.zdnet.com/article/new-tool-promises-to-turbo-charge-coding-in-major-programming-languages/

Finding Programming Vulnerabilities – Source Code as Natural Language



Analyzing source code for insecure coding

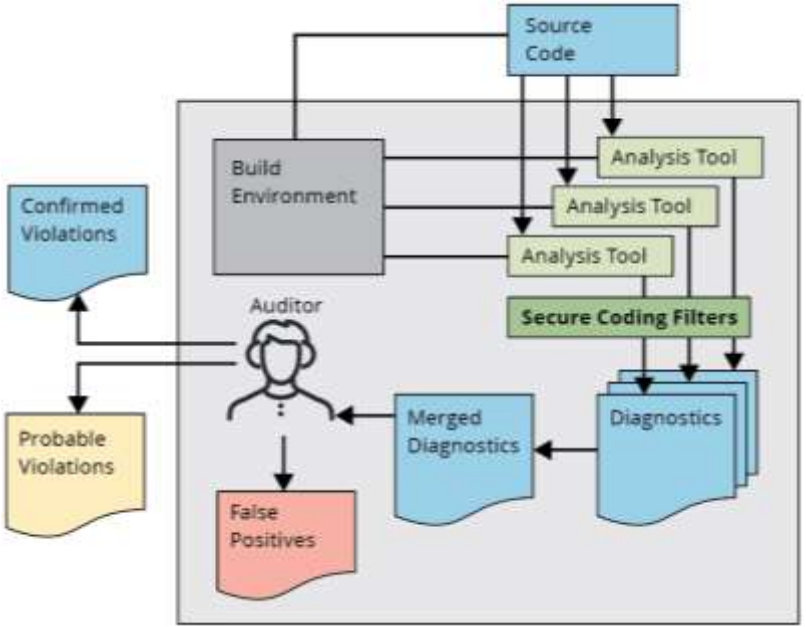
- Supplements compiler-style checking
- Treats programs like natural language

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 and machine learning 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

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: A. 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

G. Yan, J. Lu, Z. Shu ; Y. 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

D. She, K. Pei, D. Epstein, J. Yang, B. Ray, S. Jana, "NEUZZ: Efficient Fuzzing with Neural Program Smoothing," 40th IEEE Symposium on Security and Privacy, May 20--22, 2019, San Francisco, CA, USA, https://arxiv.org/pdf/1807.05620.pdf

Using AI to Improve Penetration Testing

Variety and combination of manual techniques can be executed by an AI system

- AI planning using an attack graph against attack surfaces
- Markov Decision Process (or Partially Observable Markov Decision Process) over application state
- Reinforcement learning

Sources:

K. Durkota and V. Lisy, "Computing Optimal Policies for Attack Graphs with Action Failures and Costs," Conference: Proceedings of the 7th Starting AI Researchers' Symposium (STAIRS), December 2013, https://www.researchgate.net/profile/Karel_Durkota/publication/273640839_Computing_Optimal_Policies_for_Attack_Graphs_with_Action_Failures_and_Costs

C. Sarraute, O. Buffet, and J. Hoffmann, "POMDPs Make Better Hackers: Accounting for Uncertainty in Penetration Testing," AAAI, 2012, https://arxiv.org/pdf/1307.8182

J. Schwartz, "Autonomous Penetration testing using Reinforcement Learning, Nov 16, 2018, https://arxiv.org/ftp/arxiv/papers/1905/1905.05965.pdf

Using AI for Automated Cyber Red Teaming

Red teaming simulates an attack across a system to evaluate the cybersecurity of a mission

Typically depends on having

- Cyber ontology of mission and system elements supporting mission
- Attack trees
- Mission priorities
- Planners: state-space planners, planning graph planners, and hierarchical task network-based planners

Sources:

S. Randhawa, et al, “Mission-Centric Automated Cyber Red Teaming,” ARES 2018, Proceedings of the 13th International Conference on Availability, Reliability and Security, August 27–30, 2018, Hamburg, Germany, https://www.researchgate.net/publication/327005899_Mission-Centric_Automated_Cyber_Red_Teaming

S. Upton, et al, "Breaking blue: Automated red teaming using evolvable simulations," Proceedings of Genetic and Evolutionary Computation Conference 2004, 2004, http://gpbib.cs.ucl.ac.uk/gecco2004/WMSA015.pdf

J. Yuen, "Automated Cyber Red Teaming," DTIC Document2015, https://apps.dtic.mil/docs/citations/ADA618584

Automated Program Repair – DARPA Cyber Grand Challenge

"Mayhem" demonstrated automated cyber defense

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

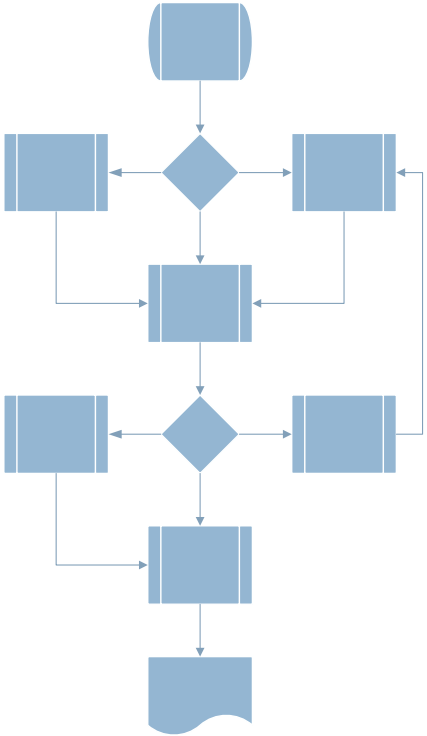
Source: DARPA, "Mayhem" Declared Preliminary Winner of Historic Cyber Grand Challenge, Aug 4, 2016, https://www.darpa.mil/news-events/2016-08-04

AI Supporting Judgement – IBM Watson to Improve Assurance

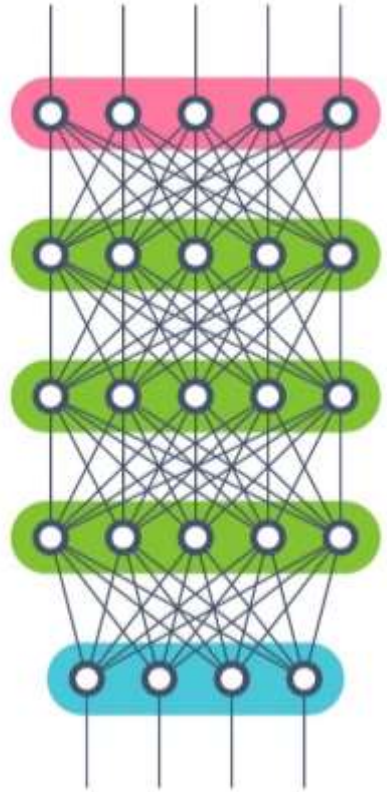
- Acquisition programs generate voluminous documentation
- Assurance is based on assembling and reviewing relevant evidence from documents
- Finding appropriate evidence or explanations can be challenging
- SEI proof of concept

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



Attack Surface Expanded

Training data

Testing data

Classifier

Model decomposition (shadow model)

AI Attacks Are Different

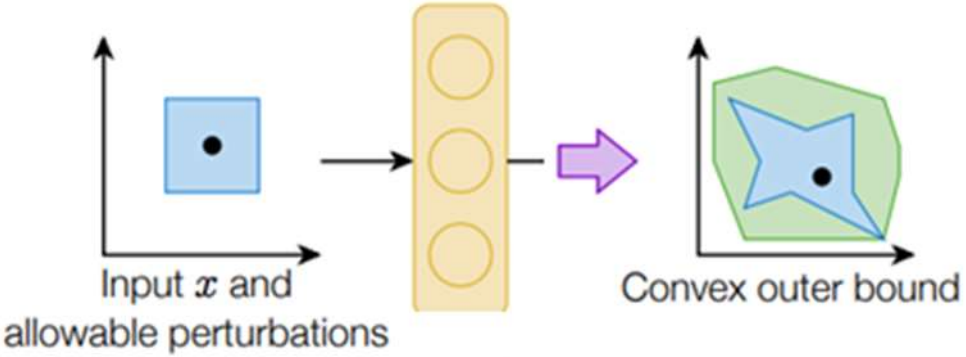
Pixel Manipulation

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

Technical Approaches for Defending AI Systems



Smooth out classification regions

Inherent tradeoff between precision and recall

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

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 faces 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