RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **DSO-T12**

Which Developers and Teams Are More Likely to Write Vulnerable Software?

Anita D'Amico, Ph.D.

CEO
Code Dx, Inc.
@anitadamico

Chris Horn

Senior Researcher
Secure Decisions
@chornsec

Lightning round of my talk

- New research links human factors to software quality and security
- Certain characteristics of software developers and work environments correlate with quality and security issues in code
- You can use knowledge of how human factors affect performance in medicine and transportation to structure work environments that yield more secure software
- You can use information about human factors to:
  - More efficiently hunt for vulnerabilities in code
  - Structure your software development team to write better and more secure code

Outline of today's talk

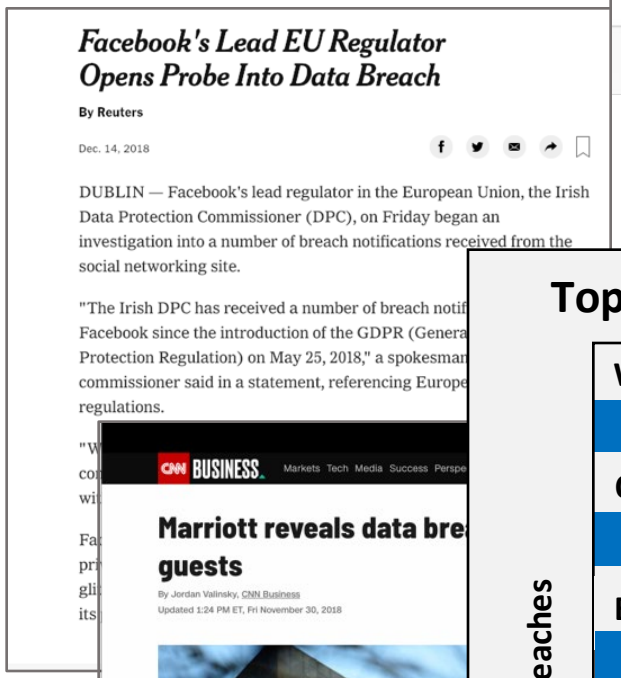
- **Why** investigate human factors that affect code quality and security?
- **How** do we conduct research to discover these human factors?
- **What** has been discovered thus far?
  - Work environment
  - Team characteristics
  - Developer behaviors & characteristics
- **Where** can we draw lessons learned from non-software domains?
  - Factors that affect human performance in transportation, medicine & healthcare, occupational safety

RSAConference2020

Why?

Why should we investigate human factors that affect code quality and security?

Software vulnerabilities are a major gateway to breaches



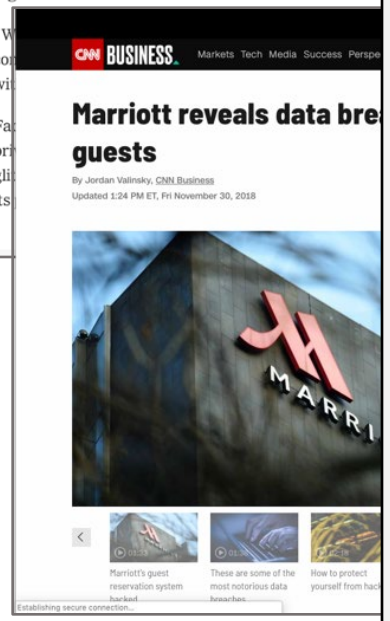
Facebook's Lead EU Regulator Opens Probe Into Data Breach

By Reuters

Dec. 14, 2018

DUBLIN — Facebook's lead regulator in the European Union, the Irish Data Protection Commissioner (DPC), on Friday began an investigation into a number of breach notifications received from the social networking site.

"The Irish DPC has received a number of breach notif
Facebook since the introduction of the GDPR (Genera
Protection Regulation) on May 25, 2018," a spokesman
commissioner said in a statement, referencing Europe
regulations.

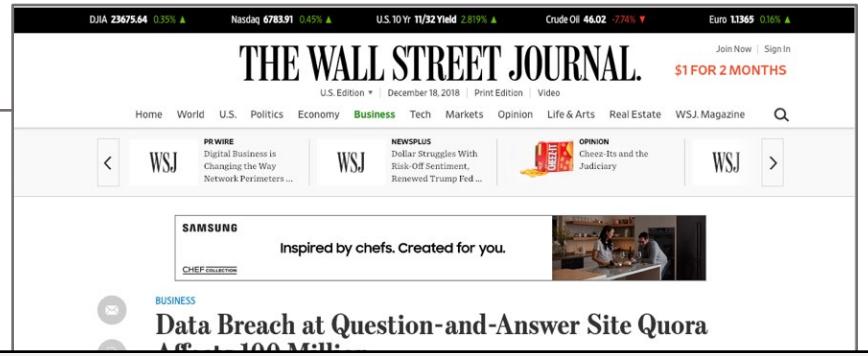


CNN BUSINESS Markets Tech Media Success Perspe

**Marriott reveals data bre
guests**

By Jordan Valinsky, CNN Business

Updated 1:24 PM ET, Fri November 30, 2018



DJIA 23675.64 0.35% Nasdaq 6783.91 0.45% U.S. 10 Yr 11/32 Yield 2.819% Crude Oil 46.02 -7.74% Euro 1.1365 0.16%

THE WALL STREET JOURNAL. Join Now | Sign In $1 FOR 2 MONTHS

U.S. Edition | December 18, 2018 | Print Edition | Video

Home World U.S. Politics Economy Business Tech Markets Opinion Life & Arts Real Estate WSJ. Magazine

PRWIRE Digital Business is Changing the Way Network Perimeters ...

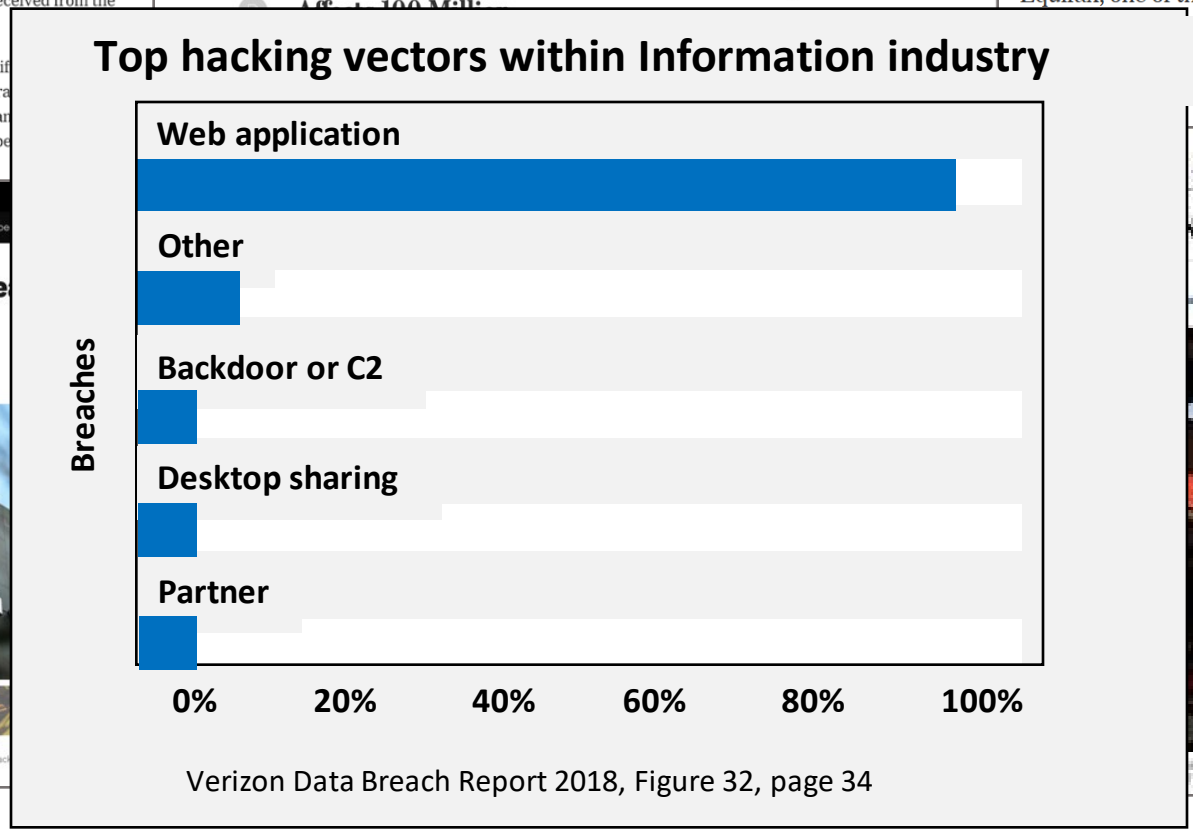
NEWSPLUS Dollar Struggles With Risk-Off Sentiment, Renewed Trump Fed ...

OPINION Cheez-Its and the Judiciary

SAMSUNG Inspired by chefs. Created for you. CHEF COLLECTION

BUSINESS

Data Breach at Question-and-Answer Site Quora



Verizon Data Breach Report 2018, Figure 32, page 34

Equifax Says Cyberattack May Have Affected 143 Million in the U.S.

By Tara Siegel Bernard, Tiffany Hsu, Nicole Perlroth and Ron Lieber

Sept. 7, 2017

1030

Equifax, one of the three major consumer credit reporting
Thursday that hackers had gained access to
t potentially compromised sensitive information
erican consumers, including Social Security

nt my data back, data back, data
k: Chili's hit by data breach

Software vulnerabilities remain undiscovered for years

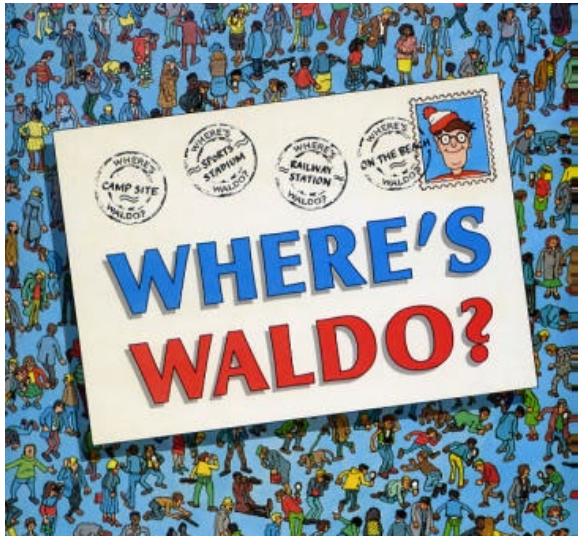
- Heartbleed took 2 years to discover
  - 500,000 secure web servers were vulnerable to theft of private keys and passwords[1]
- Apache Struts vulnerability (in Equifax breach) took 4 years to discover
  - Vulnerability exposed personal financial information of 143 million Americans[2]
- On average, vulnerabilities in open source projects remain undiscovered for two years[3]

Static Application Security Testing (SAST) and manual code reviews don't find all software vulnerabilities

One static analysis tool, on average, will only detect

14%

of all security weaknesses[1]



Manual code reviewers have difficulty finding vulnerabilities in code

Where would you hunt for vulnerabilities in code?

*Could you search for security issues based on **human factors**?*

Developer characteristics

Team characteristics

When code was written

Where code was written

Human factors are properties of people and their environment that affect human performance

Psychological

Individual
- Learning
- Ability to focus attention
- Short/long term memory
- Decision making

Group
- Collaboration & conflict
- Communication
- Cultural norms …

Physiological

- Hearing sensitivity
- Fatigue
- Circadian rhythm
- Endurance
- Health …

Environmental

- Distractions
- Temperature
- Lighting …

Human factors are considered in safety-critical systems.
Why not software engineering?

How?

How do we conduct research to discover human factors that affect code quality and security?

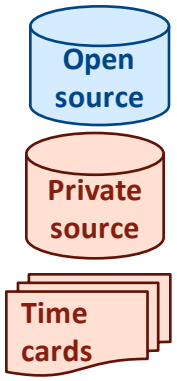
Prior human factors research in academia & industry

- Mine existing source code repositories and other systems for indirect measures of human factors
  - Analyze relationships to known vulnerabilities and failures
  - Medium & large open source software projects
    - Linux kernel, Chromium browser, PostgreSQL, etc.
  - No direct measurement of human factors
- Limited studies of proprietary development
  - Mostly large organizations, e.g. Microsoft, AT&T
- We are performing research under a DARPA R&D contract
  - Expanding research to proprietary development
  - Studying human factors directly

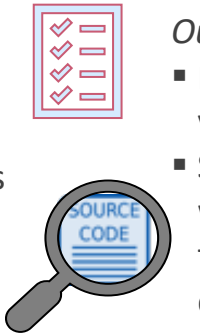
Technical approach to DARPA-funded study

Goal: Identify human factors that indicate where vulnerabilities may occur in open source and proprietary code

Retrospective analysis of software repositories



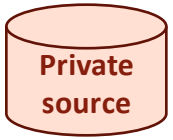
Predictors
Human behaviors and characteristics mined from development data sources

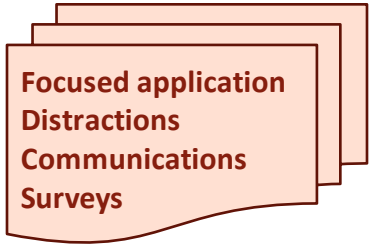


Outcomes
- Public vulnerabilities
- Security weaknesses from source code scans

Concurrent analysis of software being developed

More predictors & outcomes
Collected from instrumented development environment





Vulnerability history analysis

Discover predictors
Curate & explore history of open source vulnerabilities (intro, persistence, discovery, etc.)

What's different?

Proprietary environment

Static application security test (SAST) findings as security outcome measures

Studying developers *as they code*

Analysis of *vulnerability histories* to identify contributors to *introduction, failure to discover,* and eventual *discovery of* vulnerabilities

Research aims to answer: Can human factors predict code quality & security?

Predictors = Human Factors

- Developer's focused attention
- Context switching
- Team size
- Team collaboration
- Co-location of developers
- File editing behaviors
- Team communication
- Number of hours worked
- Time of day
- Fatigue

Outcomes = Code Security & Quality

- Publicly disclosed vulnerabilities
- Number and type of security weaknesses found by SAST
- Bug frequency
- Failure rates

RSAConference2020

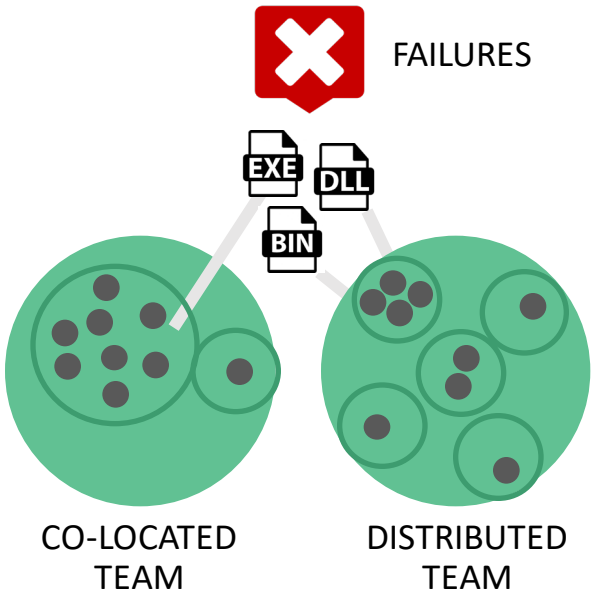
What?

What has been discovered thus far?

Does team co-location influence code quality?

- Microsoft studied post-release failures in Windows Vista and Office 2010 binaries
  - Compared binaries authored by co-located and distributed teams

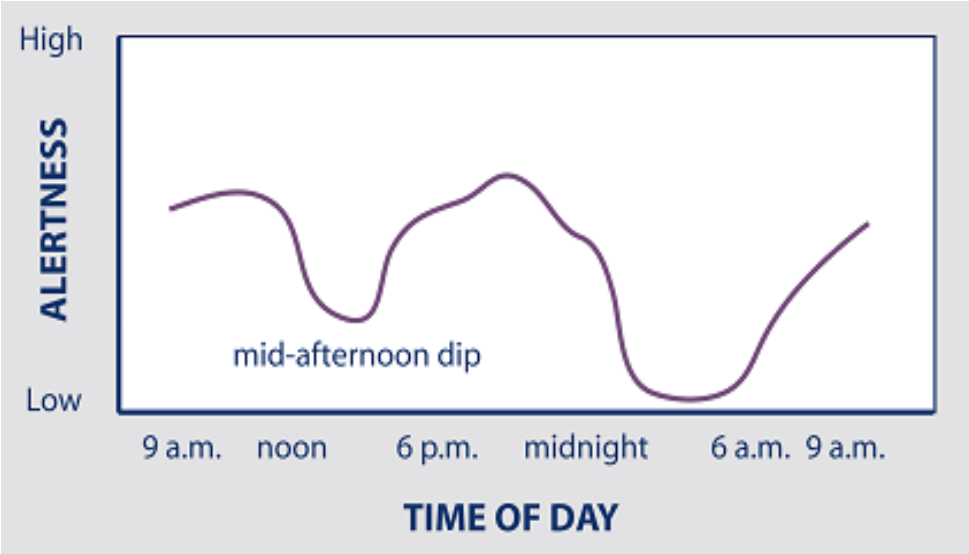
NO No difference in failure rate btw. teams[1,2]
  - No difference between teams in same building, cafeteria, campus, locality, or continent[1]
  - Binaries authored by distributed teams had ≤6% higher rate of failure[1]



Does time of day of code commits influence code quality?

YES

- Late night commits have more bugs than morning commits
  - Percent of buggy commits between midnight and 4 AM is higher than commits between 7 AM and noon[1]



Notional chart of typical circadian rhythm showing alertness throughout day

Does focus of developer's attention influence code security?

- Unfocused contribution is an indicator of how much attention developers focus on specific files
  - A file has high unfocused contribution when:
    - Developers of a file are also busy modifying *other* files, or
    - When the number of unique contributors to a file increases

YES More unfocused contribution ↔ more insecure code

  - Chromium and Apache Web Server files with 1+ vulnerability had higher unfocused contribution
  - Four repos we studied (2 open source, 2 proprietary): as unfocused contribution increases so do the number of static analysis findings

Does number of developers who contribute to a file influence code quality and security?

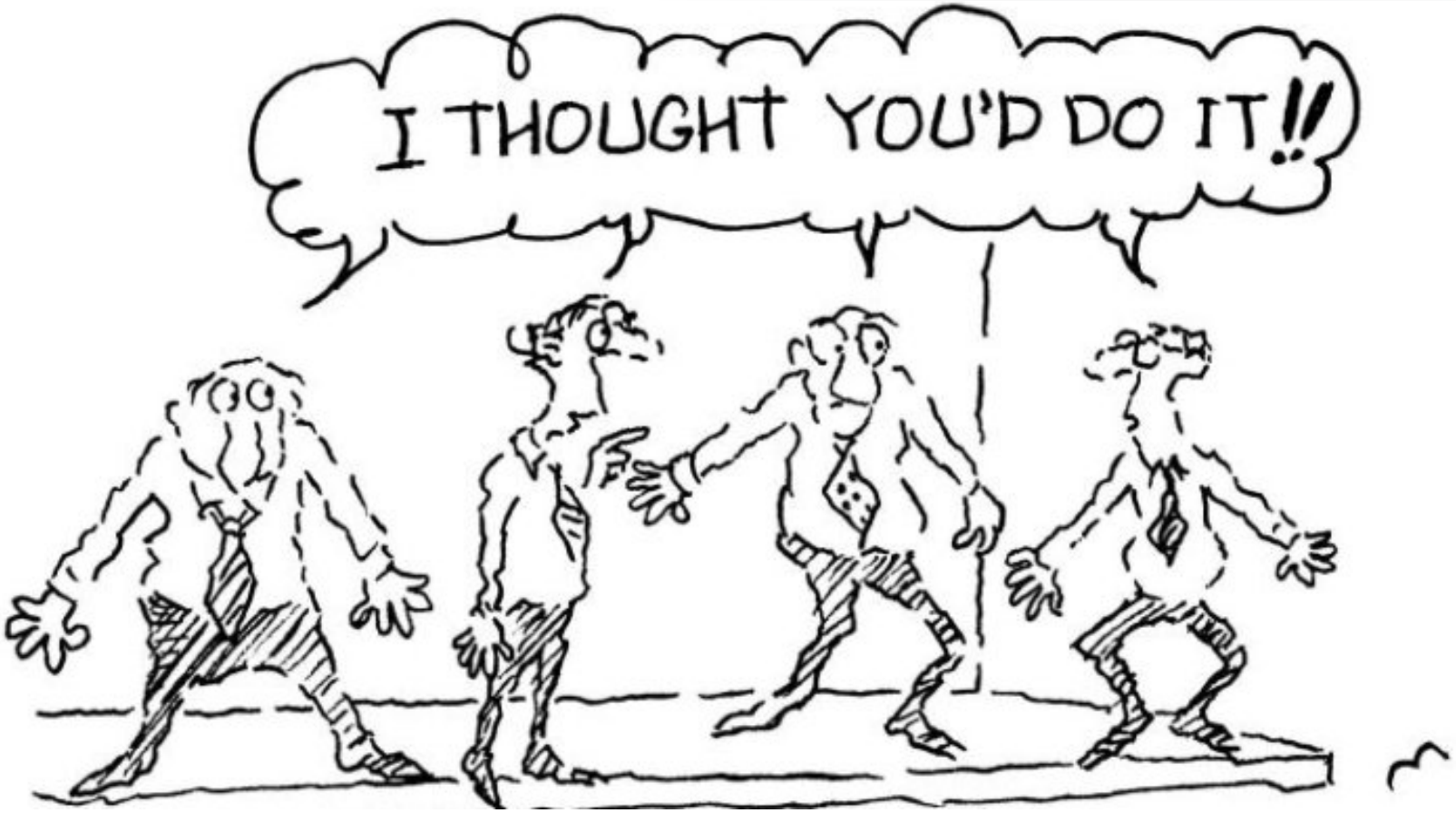
YES More developers ↔ more quality issues

- Microsoft: more developers ↔ more pre- and post-release failures[2]

YES More developers ↔ more security issues

- **Linux** kernel, source code files with **9+ developers** were **16 times more likely to have a vulnerability**[1] than files with fewer developers
- **Chromium** files with **9+ devs were 68 times more likely** to have a vulnerability
- **Apache** web server files with **9+ developers were 117 times more likely**
- Four projects we studied (2 open, 2 proprietary), source code files with **more developers contained more static analysis findings**

Why does number of developers affect code quality or security?



Perhaps a *bystander effect*?

Do large numbers of developers always have a bad effect?

NO More developers  more quality issues

- In four open source projects, lines of code modified to fix a defect had fewer contributors than other source lines[1]
- Study of telephone switching software modules found no correlation between numbers of developers and bug fixes[2]
- AT&T found number of developers contributing to a file had a negligible effect on performance of a fault prediction model[3]

Quality & security issues are similar, but not the same[4,5]

Does developer experience influence code quality?

- Developer experience defined as:
  - Number of commits to the repository, component, file, etc.[1]
  - Time since first commit to the repository[2]

YES

More developer experience ↔ higher quality software

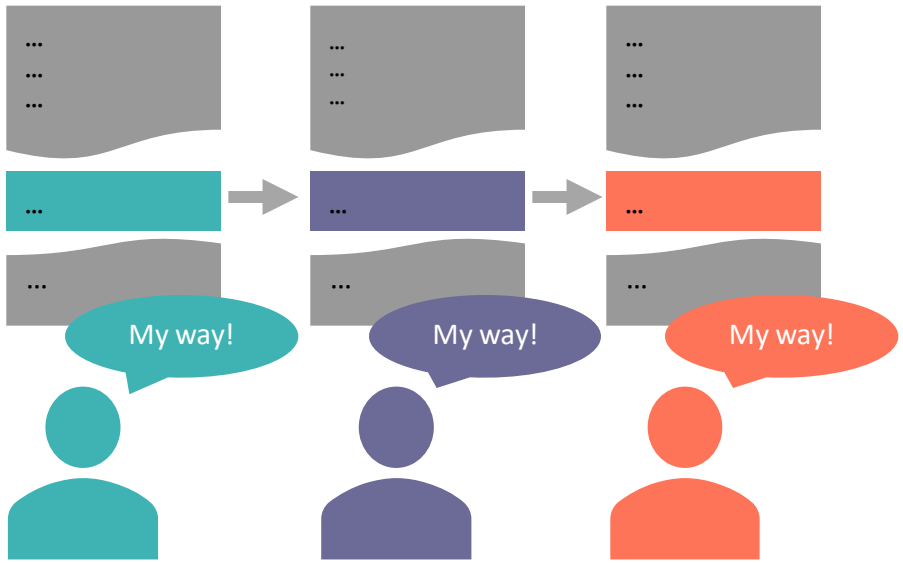
  - Microsoft found components with more minor contributors have more pre- & post-release failures[1]
    - Minor contributor is a developer whose made a small number of commits (≤ 5%) relative to total for component
  - In Linux Kernel and PostgreSQL[2], experienced devs had fewer buggy commits
    - Developers who contribute daily write fewer buggy commits
    - Day-job developers are more likely to produce bugs

Does how developers interact with each other's code influence code quality or security?

- "Interactive churn" measures percent of commit's line changes that modify code last touched by another developer

YES Editing others' code ↔ more vulnerabilities

  - In Chromium and Apache Web Server, files with at least one known vulnerability had commits with more interactive churn than files without a known vulnerability



RSAConference2020

Where?

Where can we draw lessons learned from non-software domains?

Human factors are widely known to affect performance & safety

- Transportation
  - US Federal Aviation Administration (FAA) publishes “Dirty Dozen” list of 12 human factors that lead to accidents
  - US National Transportation Safety Board (NTSB) performs root cause analyses
- Medicine & healthcare
  - World Health Organization (WHO), National Institutes of Health (NIH) training materials
- US Occupational Safety and Health Organization (OSHA)

Fatigue & vigilance

- Fatigue well-known to degrade human performance
  - After 17–19 hours without sleep, performance can be equal or worse than with a blood alcohol content (BAC) of 0.05%[1,2]
- Medicine - Fatigue-related rules[3] for medical student residents
  - 80 hours per week limit, max shift duration of 24 hours, 1 day off per week, "on-call" no more than once ever 3 nights
- U.S. Department of Transportation: max commercial drive time of 11 hours + mandatory breaks[4]
- U.S. Navy sleep regulations are based on accident investigations[5]
  - More predictable sleep schedules, oriented to circadian rhythm

Culture

- Culture
  - Complex mixture of beliefs, values, attitudes, behaviors, and goals[1,2,3]
  - "Set of shared, taken-for-granted implicit assumptions that a group holds and that determines how it perceives, thinks about and reacts to its various environments" [5]
- Health care: Culture directly linked to medical outcomes
  - Safety culture in ICUs is positively correlated with patient outcomes[4]
  - 2-year culture change intervention in 10 U.S. hospitals yielded improved patient outcomes[5]
- Security culture can affect level of attention developers give to designing security into software, and response to security issues

RSAConference2020

How to apply these findings in your workplace

How can you apply these human factors findings?

Use human factors to point you to code that might have vulnerabilities. Look at:

- Code committed after midnight
- Files where 9 or more developers contributed to a file
- Files where developers are often editing each other's code

Manage the development environment to produce more secure code

- Keep developers focused on just a few files; don't spread them across many different ones
- Limit the number of developers contributing to files
- More closely review code committed by developers who have little experience with the code base
- Introduce security culture

To participate in research or learn more, contact

Dr. Anita D'Amico
CEO,
Code Dx, Inc.
anita.damico@codedx.com
@anitadamico

Chris Horn
Senior Researcher
Secure Decisions
chris.horn@securedecisions.com
@chornsec

RSAConference2020

References

Citations of scientific findings referenced in this presentation

References

Citations are by slide and footnote number: [slide# – citation#]

[6 – 1] Based on Synopsys estimate using 2014 Netcraft Web Server Survey data, http://heartbleed.com/

[6– 2] B. Fung, "Equifax's massive 2017 data breach keeps getting worse," Washington Post. [Online].

[6 – 3] https://www.se.rit.edu/~swen-331/projects/history/

[7– 1] Kris Britton and Chuck Willis, "Sticking to the Facts: Scientific Study of Static Analysis Tools", Sept 2011: http://vimeo.com/32421617

[15 – 1] C. Bird, N. Nagappan, P. Devanbu, H. Gall, and B. Murphy, "Does distributed development affect software quality? An empirical case study of Windows Vista," 2009, pp. 518–528.

[15 – 2] Kocaguneli, Ekrem, Thomas Zimmermann, Christian Bird, Nachiappan Nagappan, and Tim Menzies. 2013. "Distributed Development Considered Harmful?" In *Proceedings of the 2013 International Conference on Software Engineering*, 882–890. ICSE '13. Piscataway, NJ, USA: IEEE Press.

[16 – 1] J. Eyolfson, L. Tan, and P. Lam, "Do Time of Day and Developer Experience Affect Commit Bugginess," in Proc. Eighth Working Conf. Mining Software Repositories, 2011, pp. 153–162.

[18 – 1] A. Meneely and L. Williams, "Secure Open Source Collaboration: An Empirical Study of Linus' Law," in Proceedings of the 16th ACM Conference on Computer and Communications Security, New York, NY, USA, 2009, pp. 453–462

[18 – 2] C. Bird, N. Nagappan, B. Murphy, H. Gall, and P. Devanbu, "Don't Touch My Code!: Examining the Effects of Ownership on Software Quality," in Proceedings of the 19th ACM SIGSOFT Symposium and the 13th Euro Conf on Found of Soft Eng, New York, NY, USA, 2011, pp. 4–14.

[20 – 1] F. Rahman and P. Devanbu, "Ownership, Experience and Defects: A Fine-grained Study of Authorship," in Proc of the 33rd Intl Conf on Software Engineering, New York, NY, USA, 2011, pp. 491–500

[20 – 2] T. L. Graves, A. F. Karr, J. S. Marron, and H. Siy, "Predicting Fault Incidence Using Software Change History," IEEE Transactions on Software Engineering, vol. 26, no. 7, pp. 653–661, Jul. 2000.

[20 – 3] E. J. Weyuker, T. J. Ostrand, and R. M. Bell, "Do too many cooks spoil the broth? Using the number of developers to enhance defect prediction models," Empirical Software Engineering, vol. 13, no. 5, pp. 539–559, Oct. 2008

[20 – 4] Munaiah, Nuthan, Felivel Camilo, Wesley Wigham, Andrew Meneely, and Meiyappan Nagappan. "Do Bugs Foreshadow Vulnerabilities? An in-Depth Study of the Chromium Project." Empirical Software Engineering 22, no. 3 (June 2017): 1305–1347. https://doi.org/10.1007/s10664-016-9447-3.

[20 – 5] Gegick, Michael, Pete Rotella, and Laurie Williams. "Toward Non-Security Failures as a Predictor of Security Faults and Failures." In Engineering Secure Software and Systems, edited by Fabio Massacci, Samuel T. Redwine, and Nicola Zannone, 5429:135–49. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. https://doi.org/10.1007/978-3-642-00199-4_12.

Approved for Public Release, Distribution Unlimited, 23 September 2019.

References

Citations are by slide and footnote number: [slide# – citation#]

[21 – 1] J. Eyolfson, L. Tan, and P. Lam, "Do Time of Day and Developer Experience Affect Commit Bugginess," in Proc. Eighth Working Conf. Mining Software Repositories, 2011, pp. 153–162.

[21 – 2] C. Bird, N. Nagappan, B. Murphy, H. Gall, and P. Devanbu, "Don't Touch My Code!: Examining the Effects of Ownership on Software Quality," in Proceedings of the 19th ACM SIGSOFT Symposium and the 13th Euro Conf on Found of Soft Eng, New York, NY, USA, 2011, pp. 4–14.

[24 – 1] S. Wang and N. Nagappan, "Characterizing and Understanding Software Developer Networks in Security Development," CoRR, vol. abs/1907.12141, 2019.

[25 – 1] Williamson, A, and A. Feyer. "Moderate Sleep Deprivation Produces Impairments in Cognitive and Motor Performance Equivalent to Legally Prescribed Levels of Alcohol Intoxication." Occupational and Environmental Medicine 57, no. 10 (October 2000): 649–55. https://doi.org/10.1136/oem.57.10.649.

[25 – 2] Dawson, Drew, and Kathryn Reid. "Fatigue, Alcohol and Performance Impairment." Nature 388, no. 6639 (July 1997): 235–235. https://doi.org/10.1038/40775.

[25 – 3] "Duty Hours and Patient Safety." AHRQ Patient Safety Network, January 2019. https://psnet.ahrq.gov/primers/primer/19/duty-hours-and-patient-safety.

[25 – 4] "Summary of Hours of Service Regulations." Text. Federal Motor Carrier Safety Administration, December 30, 2013. https://www.fmcsa.dot.gov/regulations/hours-service/summary-hours-service-regulations.

[25 – 5] Ziezulewicz, Geoff. "Navy Issues New Sleep and Watch Schedule Rules for the Surface Fleet." Navy Times, September 20, 2017. https://www.navytimes.com/news/your-navy/2017/09/20/navy-issues-new-sleep-and-watch-schedule-rules-for-the-surface-fleet/.

[27 – 1] "What Is a PE?" Accessed September 16, 2019. https://www.nspe.org/resources/licensure/what-pe.

References

Citations are by slide and footnote number: [slide# – citation#]

[26 – 1] 1 "8 Steps to a Strong Safety Culture." Accessed September 16, 2019. https://www.ishn.com/articles/91474-8-steps-to-a-strong-safety-culture.

[26 – 2] Cole, Kerstan, Susan Stevens-Adams, and Caren Wenner. "A Literature Review of Safety Culture.," March 1, 2013. https://doi.org/10.2172/1095959.

[26 – 3] Sexton, John B., Robert L. Helmreich, Torsten B. Neilands, Kathy Rowan, Keryn Vella, James Boyden, Peter R. Roberts, and Eric J. Thomas. "The Safety Attitudes Questionnaire: Psychometric Properties, Benchmarking Data, and Emerging Research." BMC Health Services Research 6, no. 1 (April 3, 2006): 44. https://doi.org/10.1186/1472-6963-6-44.

[26 – 4] Huang, David T., Gilles Clermont, Lan Kong, Lisa A. Weissfeld, J. Bryan Sexton, Kathy M. Rowan, and Derek C. Angus. "Intensive Care Unit Safety Culture and Outcomes: A US Multicenter Study." International Journal for Quality in Health Care 22, no. 3 (June 2010): 151–61. https://doi.org/10.1093/intqhc/mzq017.

[26 – 5] Curry, Leslie A., Marie A. Brault, Erika L. Linnander, Zahirah McNatt, Amanda L. Brewster, Emily Cherlin, Signe Peterson Flieger, Henry H. Ting, and Elizabeth H. Bradley. "Influencing Organisational Culture to Improve Hospital Performance in Care of Patients with Acute Myocardial Infarction: A Mixed-Methods Intervention Study." BMJ Quality & Safety 27, no. 3 (March 1, 2018): 207–17. https://doi.org/10.1136/bmjqs-2017-006989.

[26 – 6] Schein, E. H. (1996). Culture: The missing concept in organization studies. Administrative Science Quarterly, 41(2), 229–240