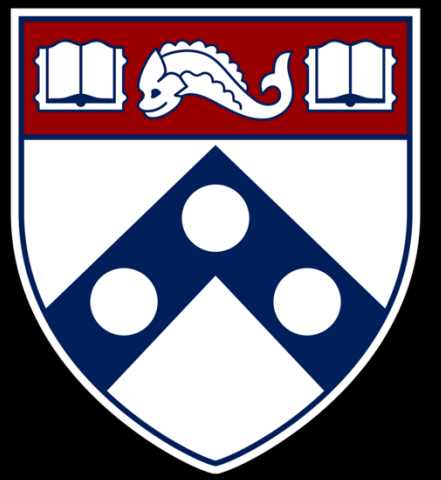
Secure Systems Engineering and Management

A Data-driven Approach



Introduction

Michael Hicks

UPenn CIS 7000-003
Spring 2026

How would you answer this question?

In the last decade, has the **security of computer systems**, generally,

- **improved**,
- **declined**, or
- **stayed the same**?

To answer it, we need data

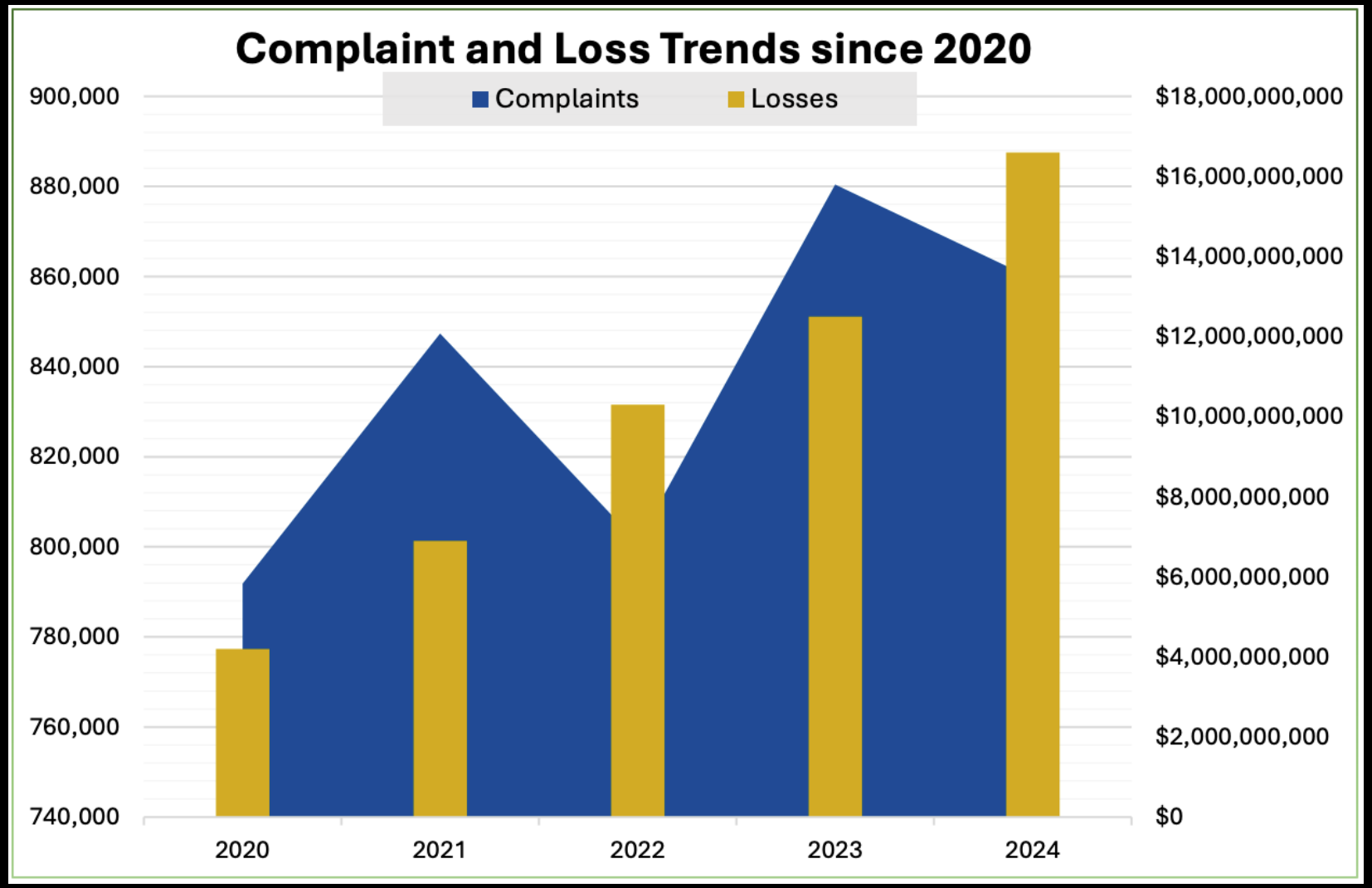
- Cybersecurity is improving if costs due to attacks are going down
  - Direct costs of the attack (e.g., lost or stolen assets)
  - Indirect costs of the attack (e.g., costs of downtime, recovery)
- Should also consider costs of defenses
  - Extra personnel and equipment, introduced inefficiencies (slower logins, dealing with false alarms, etc.)

Cybercrime $$: Overall estimate

- “It is estimated that the cost of cybercrime will grow from an annual sum of **$3 trillion in 2015** to **$6 trillion in 2021**” – cited 2016 report by Cybersecurity Ventures
- Evolve Security blog post (written 2023) agrees with those numbers, estimates **$20 trillion cost by 2026**

Cybercrime $$: FBI IC3 direct data

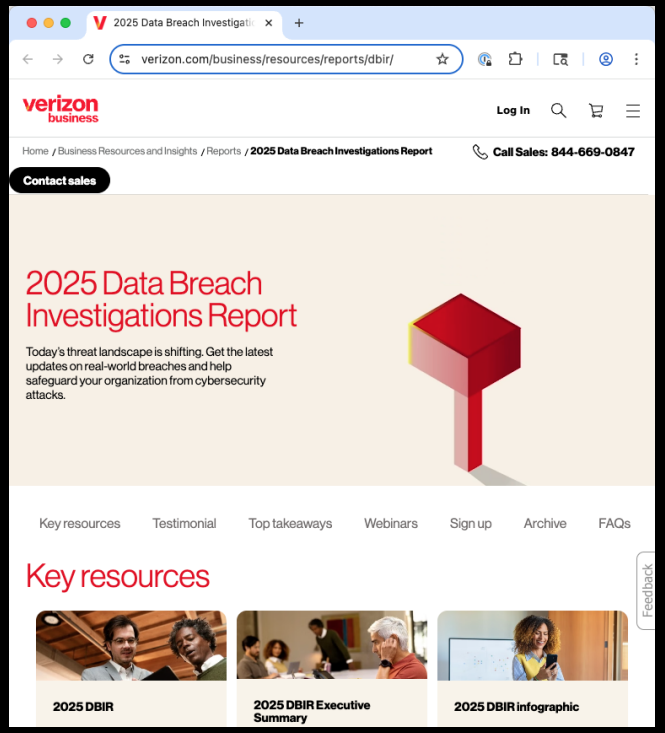
- Increasing cumulative cost
- Generally increasing number of incidents
- Varying trend on cost per incident



How should we protect ourselves?

- **Prioritize** vectors of attack based on **risk**
- Work backwards from an attack chain: **Consider methods** that break a link
  - Example: Dev tools to address vulnerabilities
  - Example: Patch 3p vulnerabilities quickly
  - Example: Train users to avoid the phish
- **Assess** against direct and indirect measures of effectiveness
  - Local: Experiments, proofs, arguments
  - Global: Prevalence of types of exploitation

Consider attack vectors in data breaches



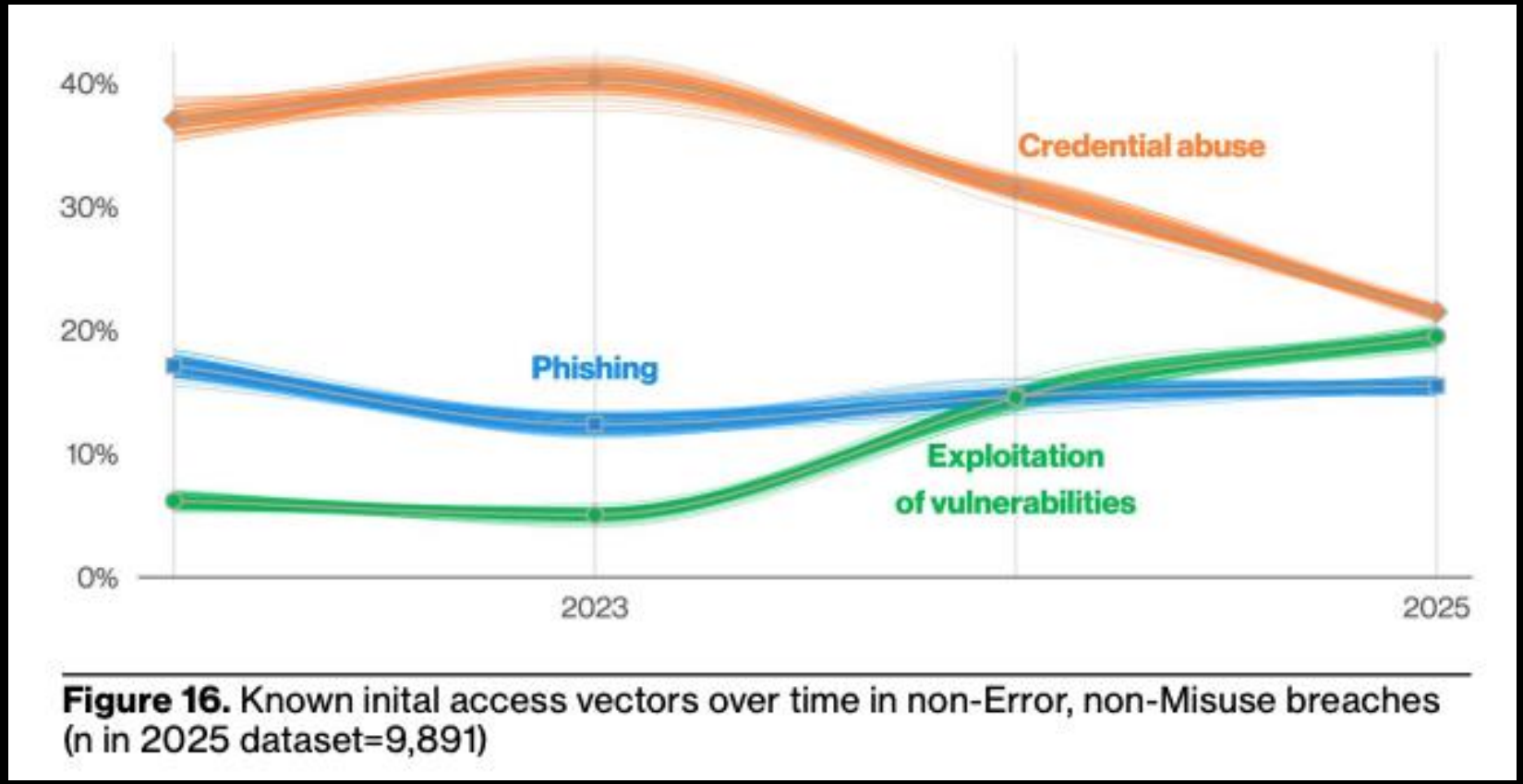


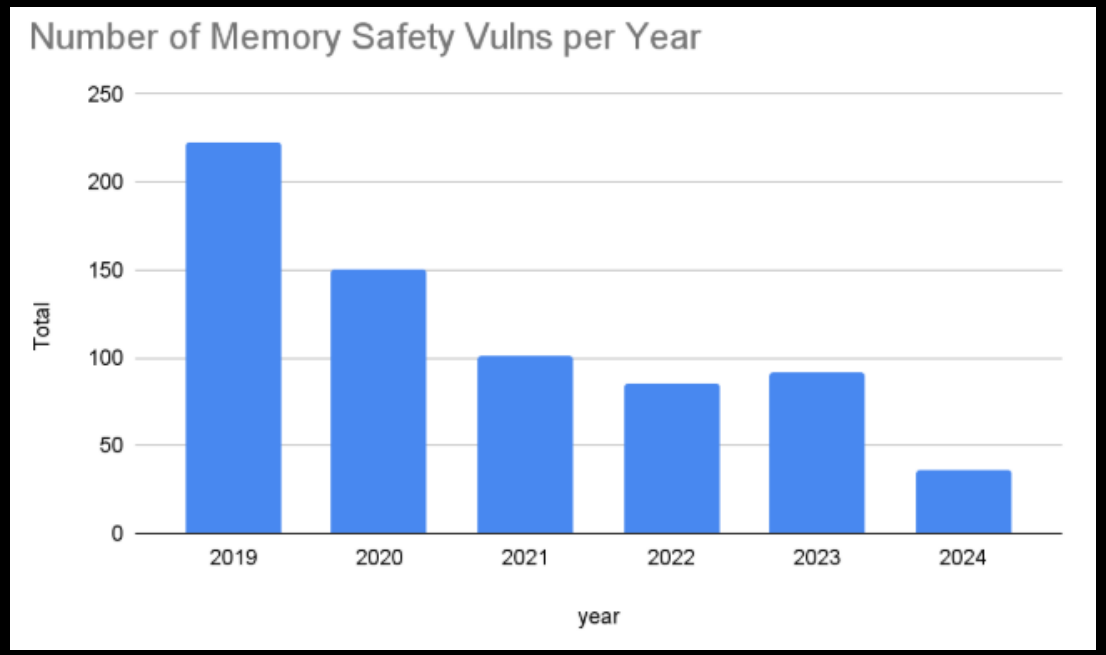
Figure 16. Known inital access vectors over time in non-Error, non-Misuse breaches (n in 2025 dataset=9,891)

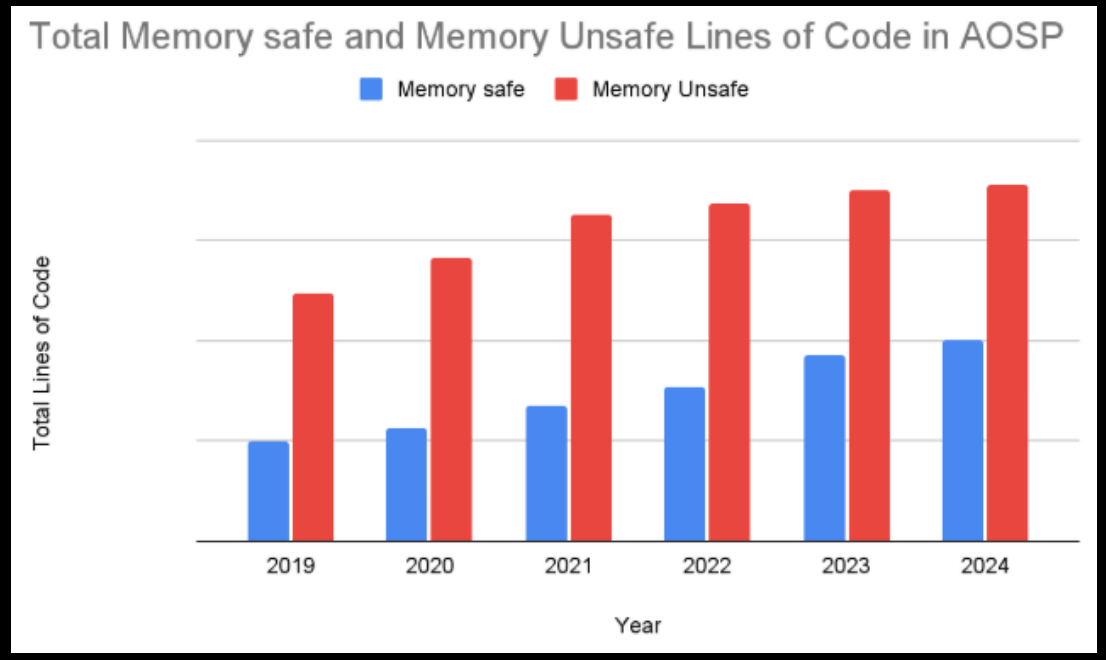
Source: 2025 Verizon Data Breach Investigations Report

Method: Use PL that prevents vulnerabilities

Android is writing most new code in Rust, and fixing vulns in its C/C++.

Resuilt: A roughly exponential drop in vulnerabilities reported

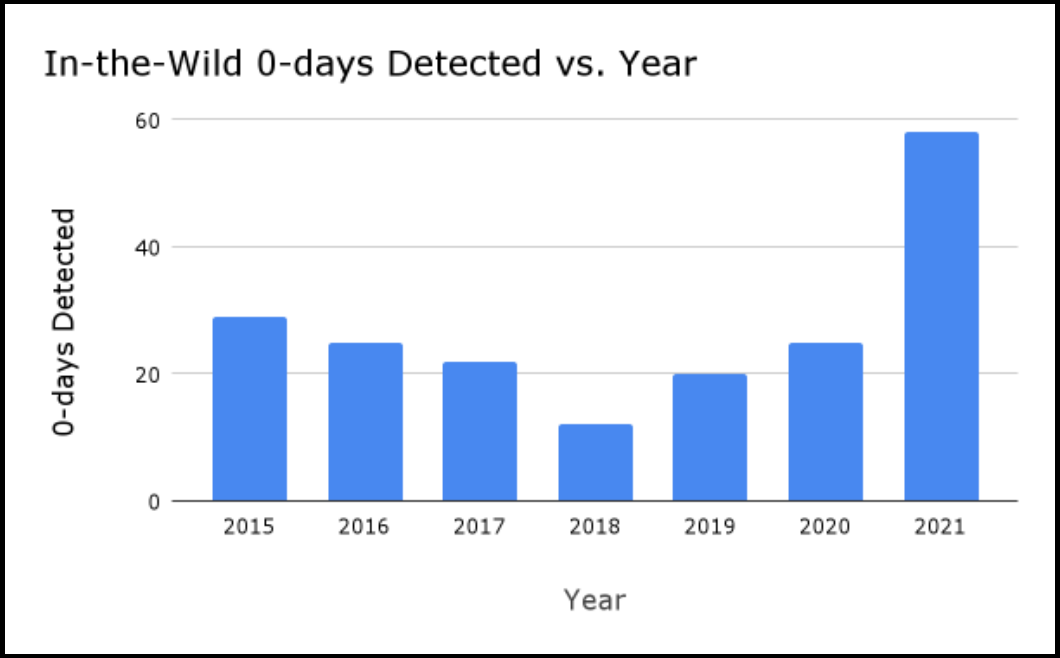




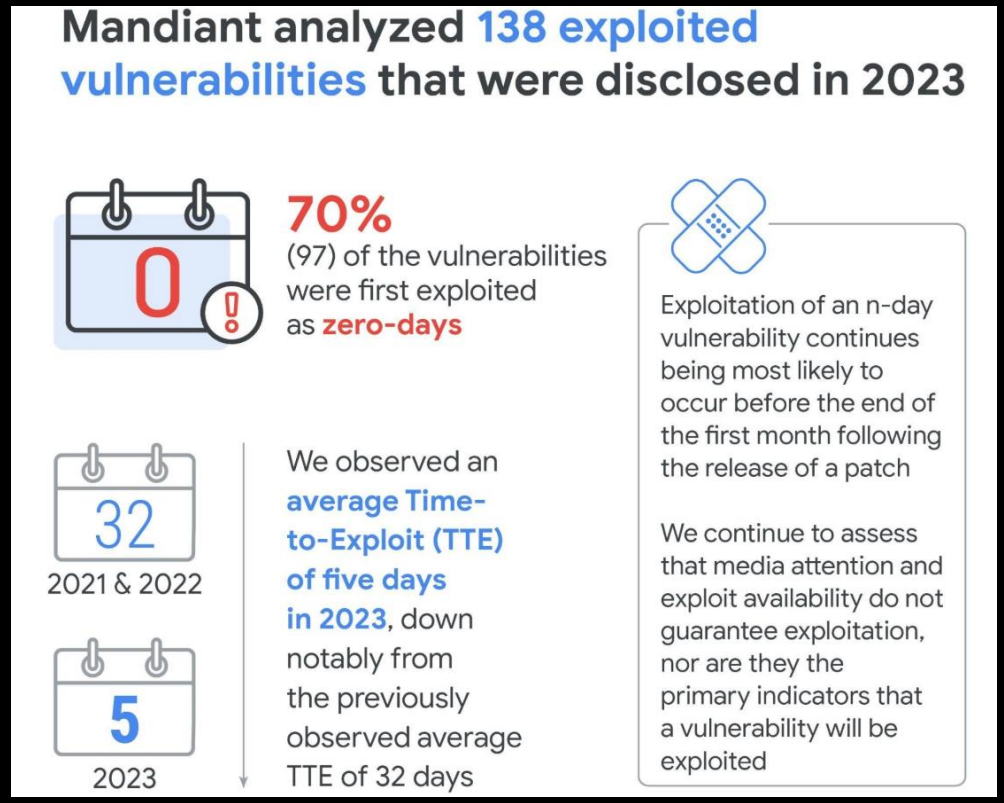
Source: Google Security Blog

Method: Patch bugs faster

So: Attackers cannot keep exploiting the same vulnerabilities year after year



Source: Google Project Zero



Source: Mandiant 2024 Threat Intelligence Report

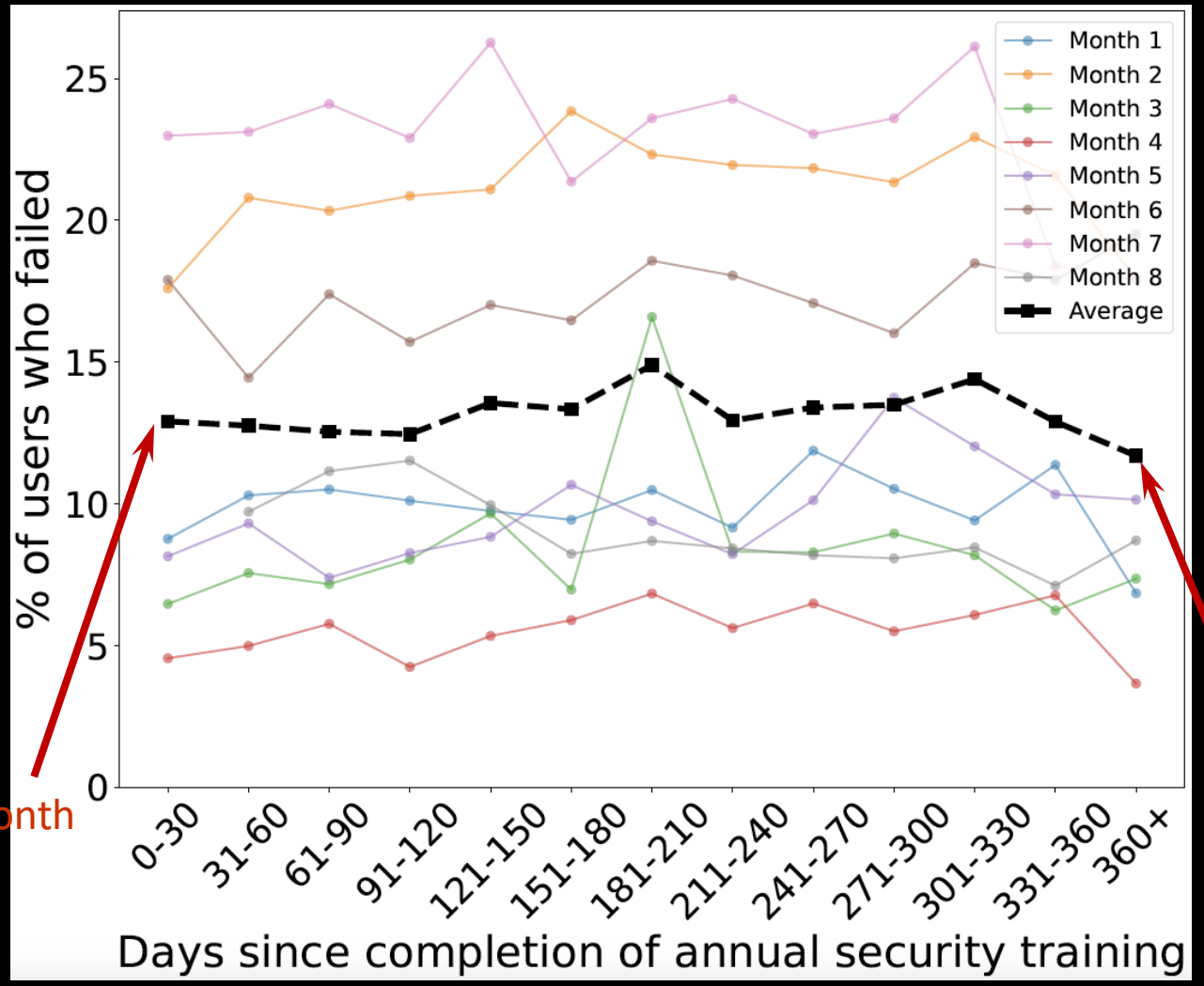
Evidence: Adversaries shifting to greater use of 0-day vulnerabilities and exploiting them sooner

Method: Annual security awareness training

Can help address prevalent phishing attacks?

Evidence: Data and statistical models (GLME) find ***no association*** between:

1. how long ago a user completed training (KnowBe4) and
2. their likelihood of failing a phishing simulation



Source: Ho et al, “Understanding the efficacy of phishing training in practice.” S&P 2025.

Evidence-based security

Outcomes
- Data breached,
- Vulnerabilities exploited,
- Revenue lost,
- …

Analysis
- What are the (still) successful vectors of attack?
- Where is risk (still) greatest?
- What interventions could be deployed cost-effectively?

Intervention
- Engineering,
- Operations,
- Policy,
- Education, …

If it's working, we should see:
- A decline in successful attacks, according to a consistent data collection system
- Updated best practices to remove demonstrably ineffective techniques, like password rotation

Course goals: You will be able to

- **Understand cybersecurity from a data-driven and economic perspective**, learning to make decisions based on empirical evidence, following good science
- **Identify key vulnerabilities and threats**, especially when considering the **impact of humans**, both when they are attack targets and when the play a role in ensuring a system's security
- **Follow a well-designed process for secure systems construction**, from threat modeling to building to testing to maintenance
- **Manage security operations** – preventing, detecting, mitigating, and recovering from incidents – and gather data to improve future posture
- **Make risk-informed decisions**: Assess designs and technologies according to how they mitigate security risk, while leveraging **insurance** and responding to **regulation**
- **Communicate effectively and with empathy** to key stakeholders about security options and recommendations

All while taking a data-informed approach

Schedule

Today – 12 Feb (weeks 1-6)

- Threat review: vulnerabilities and social engineering
- Speaking and writing well
- Empirical cybersecurity
  - Economics of cybersecurity
  - Cybersecurity as a scientific pursuit
  - Measuring and analyzing security

17 Feb – 24 Mar (weeks 6-10)

- Secure software development
  - Threat modeling
  - Secure system design
  - Programming (memory safety!)
  - Pen testing (fuzzing)
  - Supply chain, patching, vulnerability remediation

Schedule

17 Feb – 24 Mar (weeks 6-10)

- Secure software development
  - Threat modeling
  - Secure system design
  - Programming (memory safety!)
  - Pen testing (fuzzing)
  - Supply chain, patching, vulnerability remediation

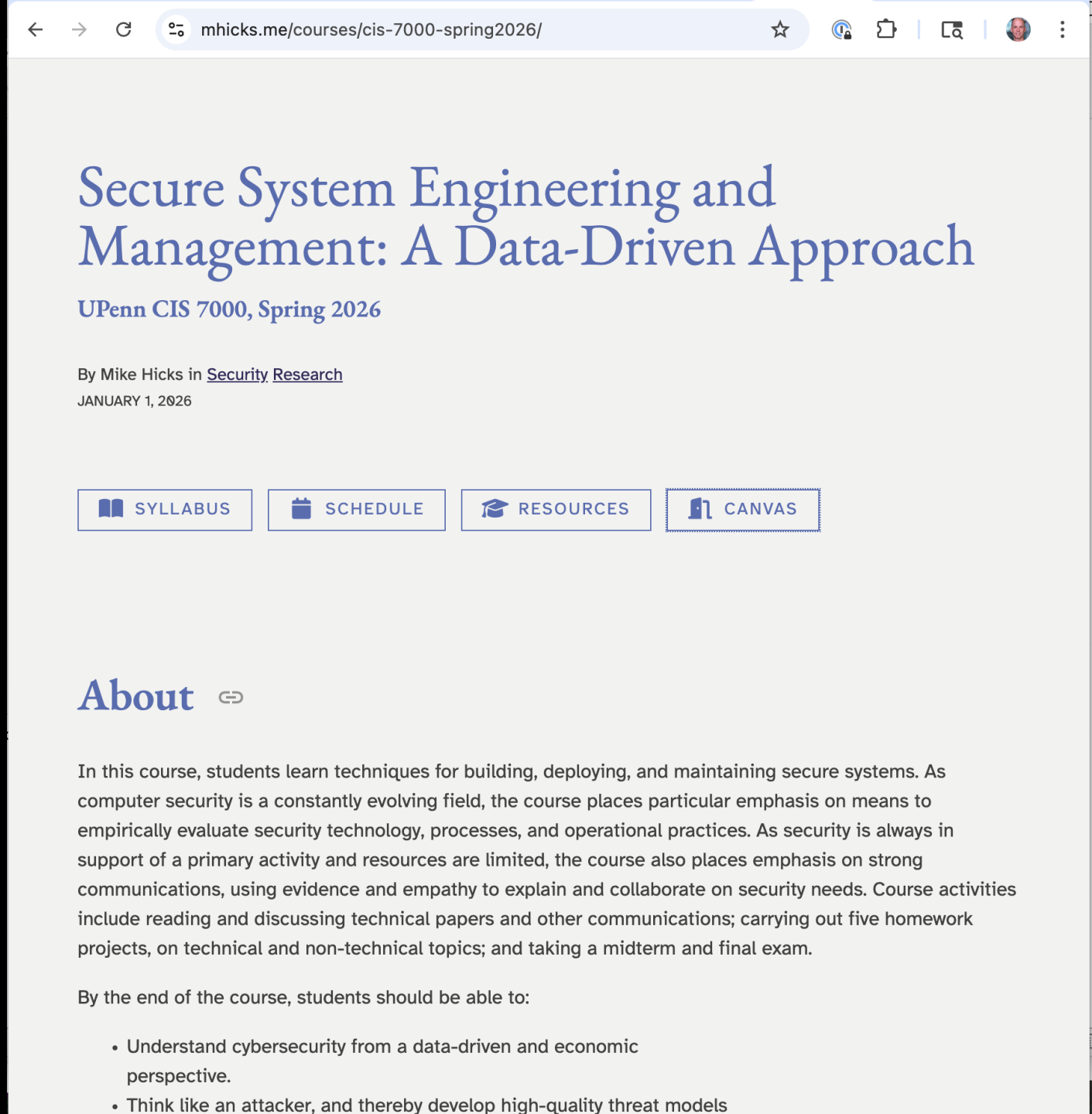
26 Mar – 28 Apr (weeks 10-14)

- Security operations
  - Incident detection and response
  - Management
  - Making risk informed decisions
  - The role and activities of the CISO
- Cyber regulation and insurance

Bonus content week 10 (and throughout): Impact of AI/ML on security

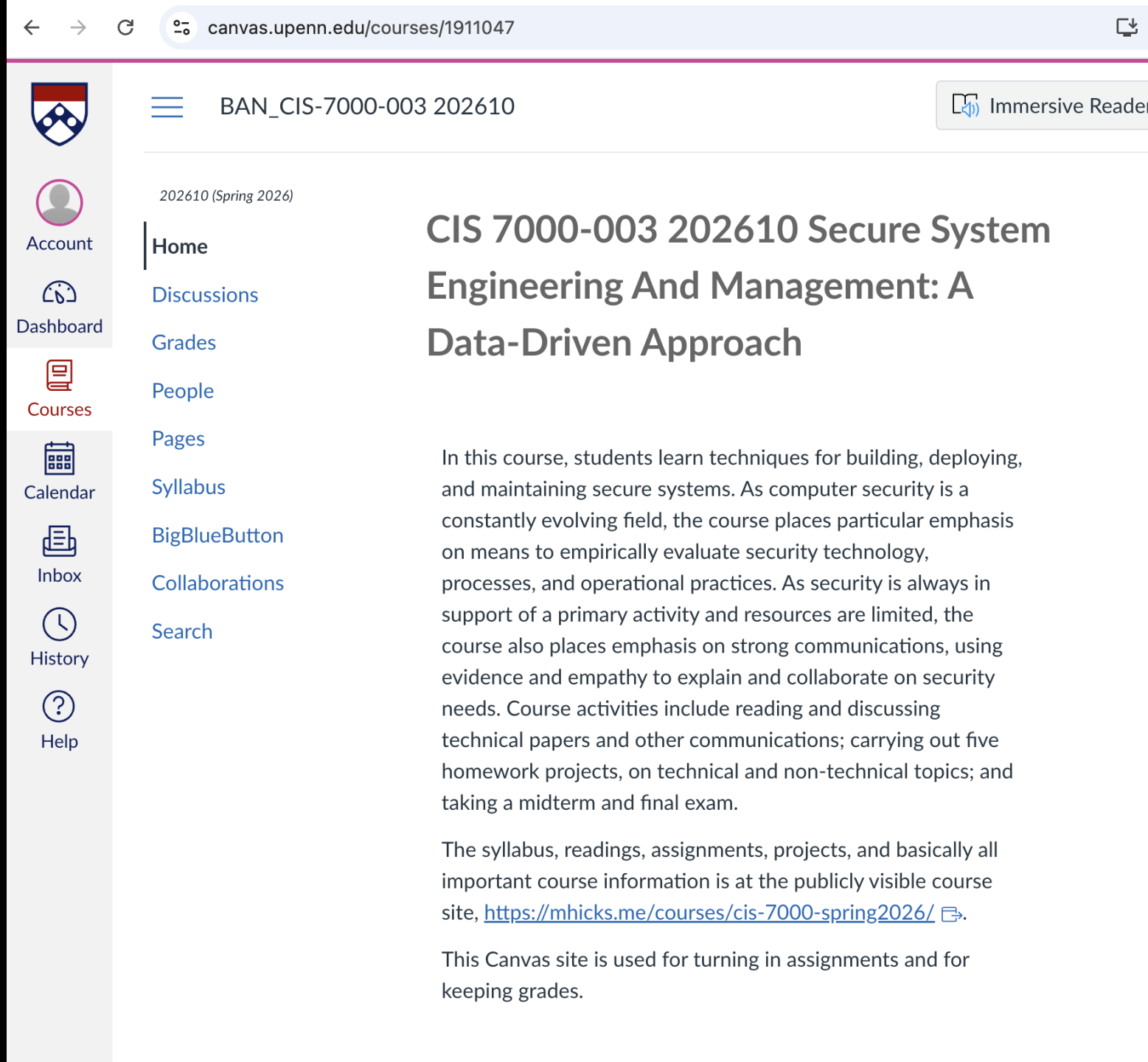
Main WWW site

https://mhicks.me/courses/cis-7000-spring2026/



Canvas site

https://canvas.upenn.edu/courses/1911047



Graded activities

- Read and critically review research papers, other sources
- Discuss them, and course topics generally, in class
- Do 5 (solo) projects
  - Communication
  - Data analysis
  - Threat modeling
  - Fuzzing
  - SecOps
- Take 2 exams (midterm and final)

Read papers: Question, understand, improve

Good practice for the future!

How to Read a Paper

S. Keshav
David R. Cheriton School of Computer Science, University of Waterloo
Waterloo, ON, Canada
keshav@uwaterloo.ca

ABSTRACT

Researchers spend a great deal of time reading research papers. However, this skill is rarely taught, leading to much wasted effort. This article outlines a practical and efficient *three-pass method* for reading research papers. I also describe how to use this method to do a literature survey.

Categories and Subject Descriptors: A.1 [Introductory and Survey]

General Terms: Documentation.

Keywords: Paper, Reading, Hints.

1. INTRODUCTION

Researchers must read papers for several reasons: to review them for a conference or a class, to keep current in their field, or for a literature survey of a new field. A typical researcher will likely spend hundreds of hours every year reading papers.

Learning to efficiently read a paper is a critical but rarely taught skill. Beginning graduate students, therefore, must learn on their own using trial and error. Students waste much effort in the process and are frequently driven to frus-

4. Glance over the references, mentally ticking off the ones you've already read

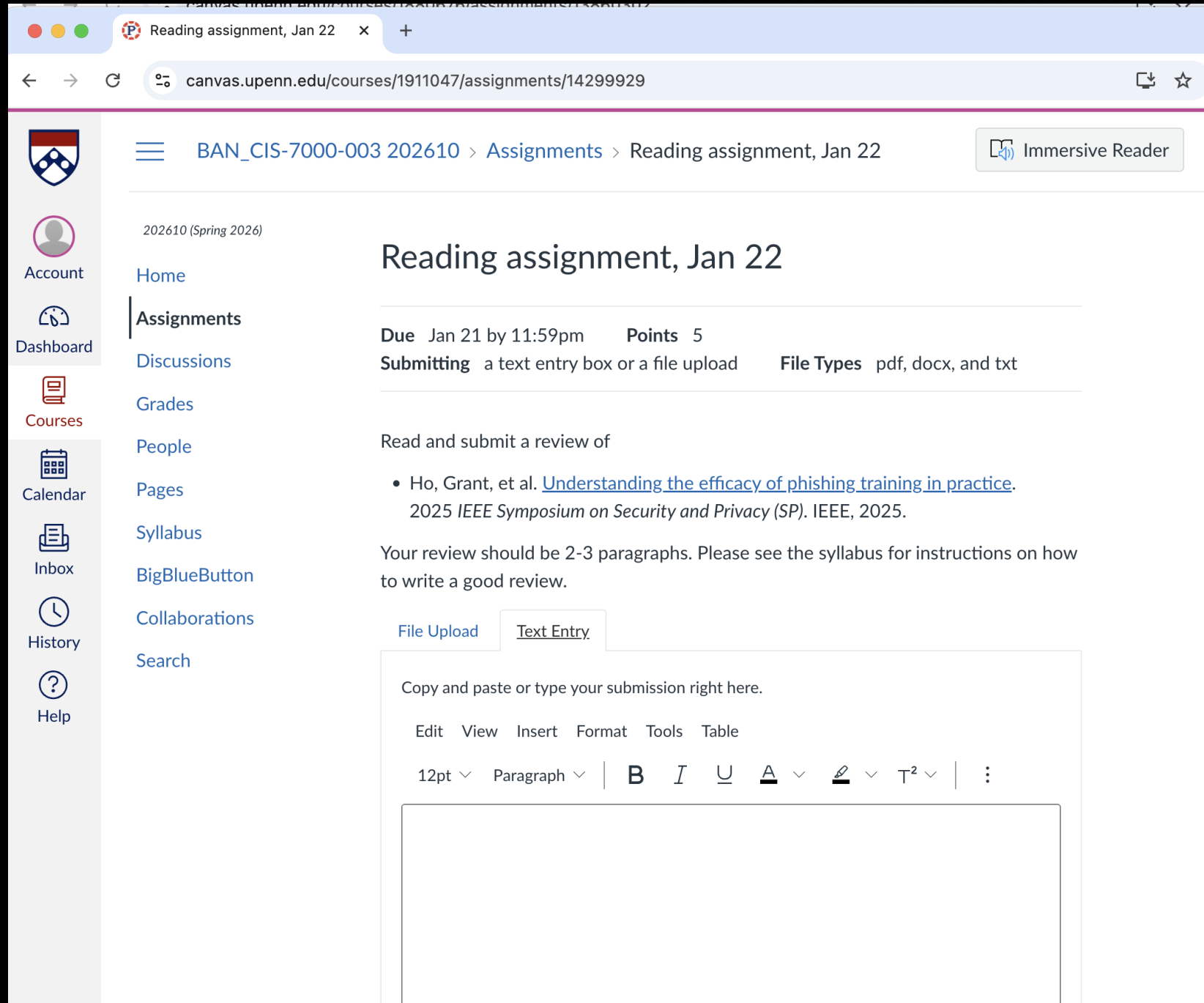
At the end of the first pass, you should be able to answer the *five Cs*:

1. *Category*: What type of paper is this? A measurement paper? An analysis of an existing system? A description of a research prototype?
2. *Context*: Which other papers is it related to? Which theoretical bases were used to analyze the problem?
3. *Correctness*: Do the assumptions appear to be valid?
4. *Contributions*: What are the paper's main contributions?
5. *Clarity*: Is the paper well written?

Using this information, you may choose not to read further. This could be because the paper doesn't interest you, or you don't know enough about the area to understand the paper, or that the authors make invalid assumptions. The

Class prep

- Read the paper(s). Submit a 2-3 paragraph review the day before

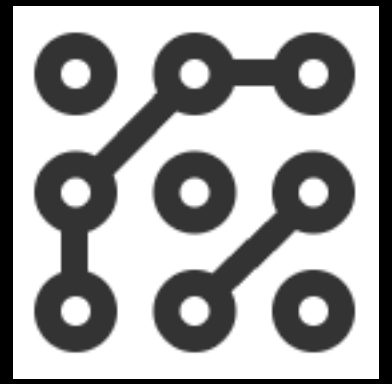


Personnel

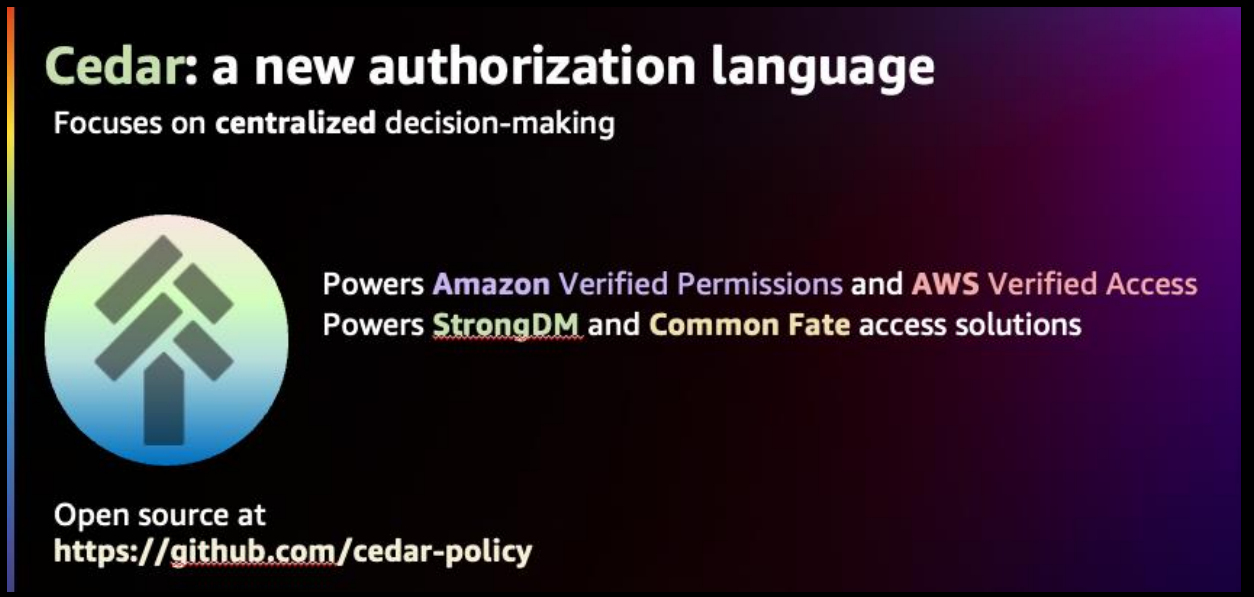
Professor in CIS
Director of Schlein Center for Cybersecurity

Teaching assistant
PhD student in CIS, focus on cybersecurity

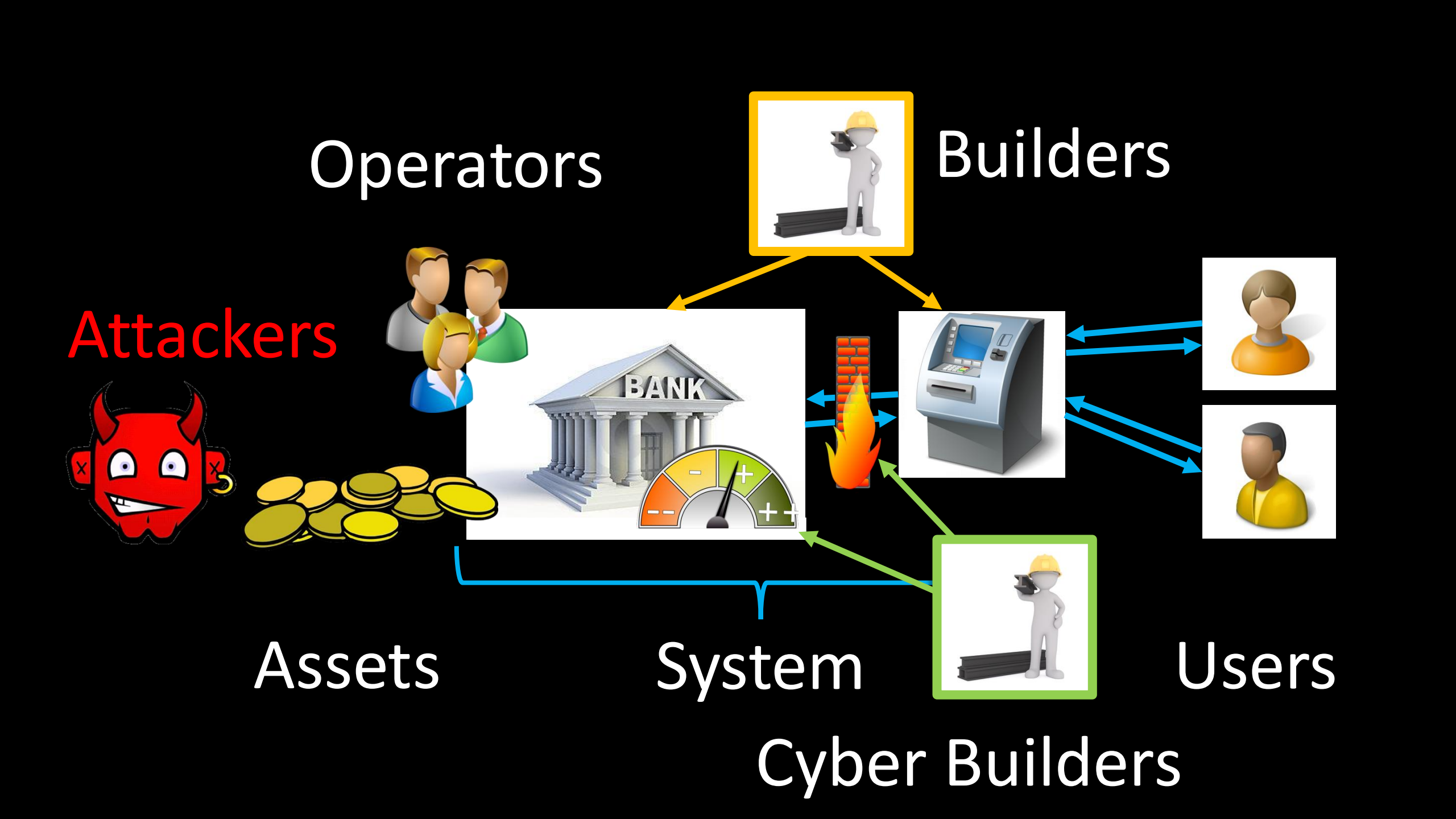
About me

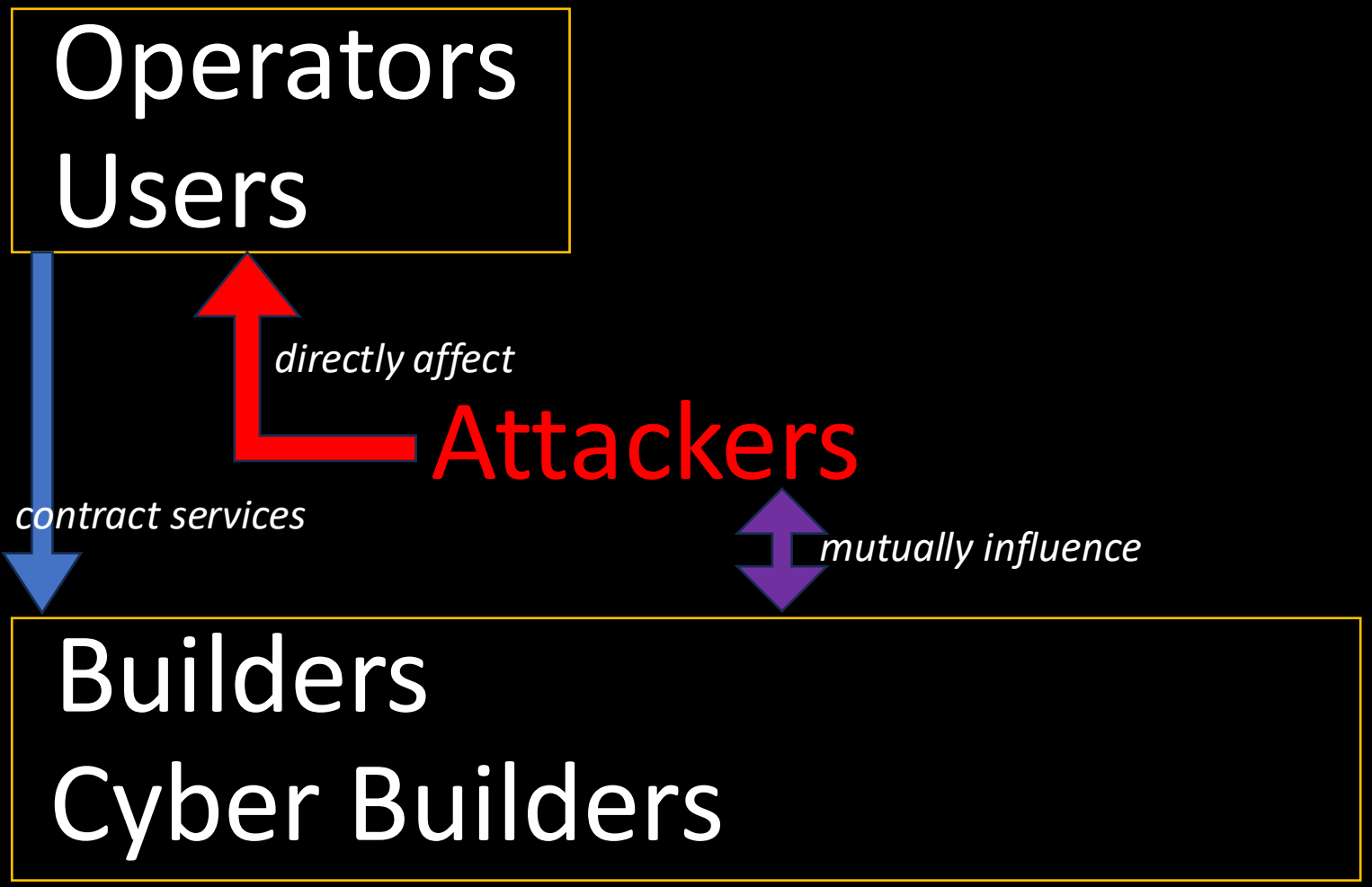


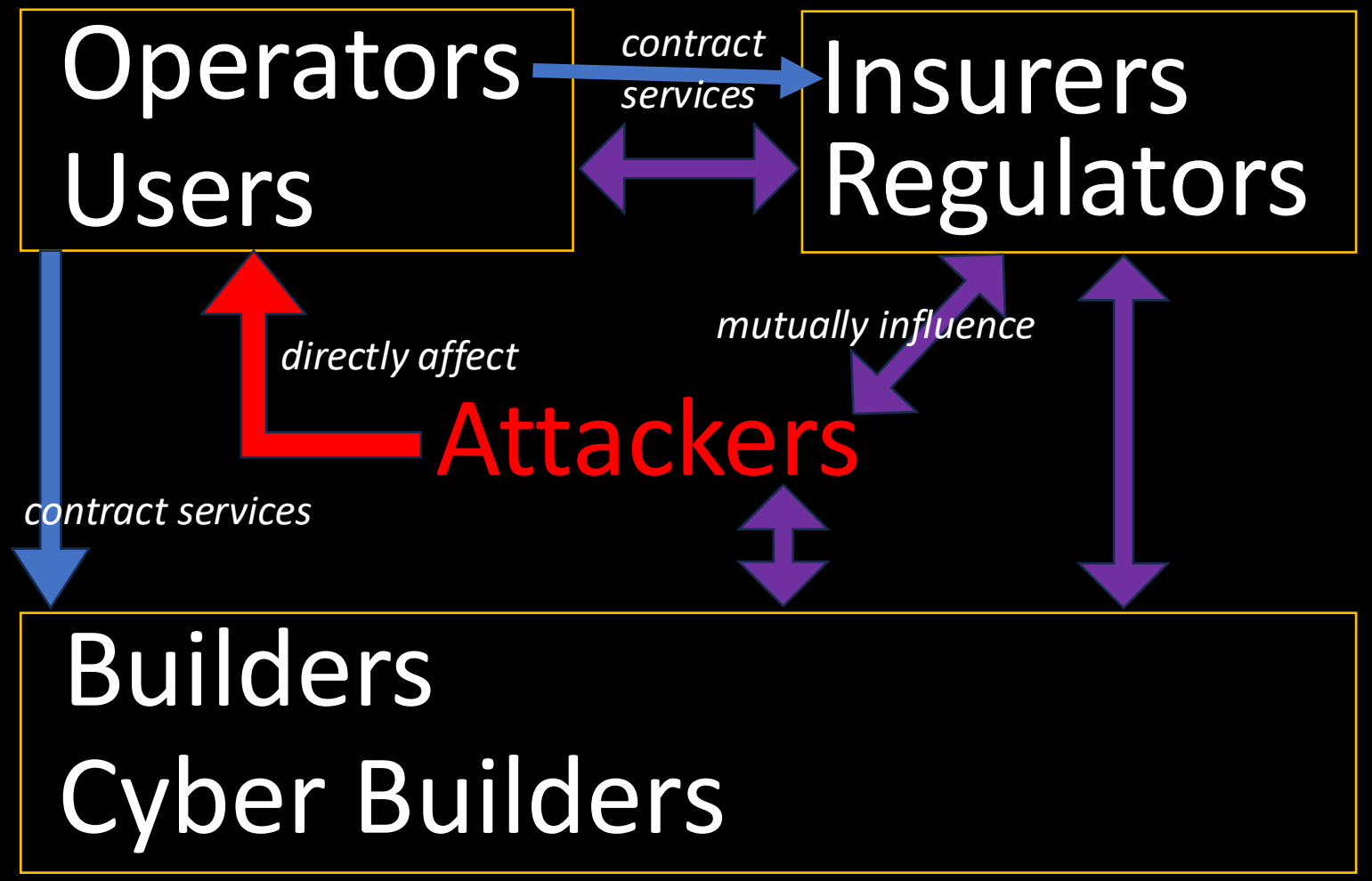
- Prof (2002-present): Research in **Software Security**, Programming Languages, Software Engineering, Usability, Cryptography
- Startup (2018-2021): Building tools for secure software development
  - Binary analysis
  - Migration to memory-safe C
- AWS (2022-2025)
  - Cedar authorization language
  - Fuzzing/automated test generation
  - Formal/mechanized proofs of security



Course overview


Operators
Builders
Attackers
BANK
Assets
System
Users
Cyber Builders


Operators
Users
directly affect
Attackers
contract services
mutually influence
Builders
Cyber Builders



Consider:

- Relationships induce incentives
- (Cyber)builders are users and operators too!

Attackers

- Overwhelming reason for attacks: Cybercrime
  - But also: national-state activities, such as espionage and cyber-war
- Value proposition: Is the expected cost of developing and carrying out the attack worth the expected reward?
  - Costs and benefits are both monetary and non-monetary
  - As the world has become more cyber-enabled, the rewards have increased
  - But defenses have made **carrying out attacks much harder**, too!

Ukraine power grid attack (2015)

In the Ukraine power grid cyber attack,

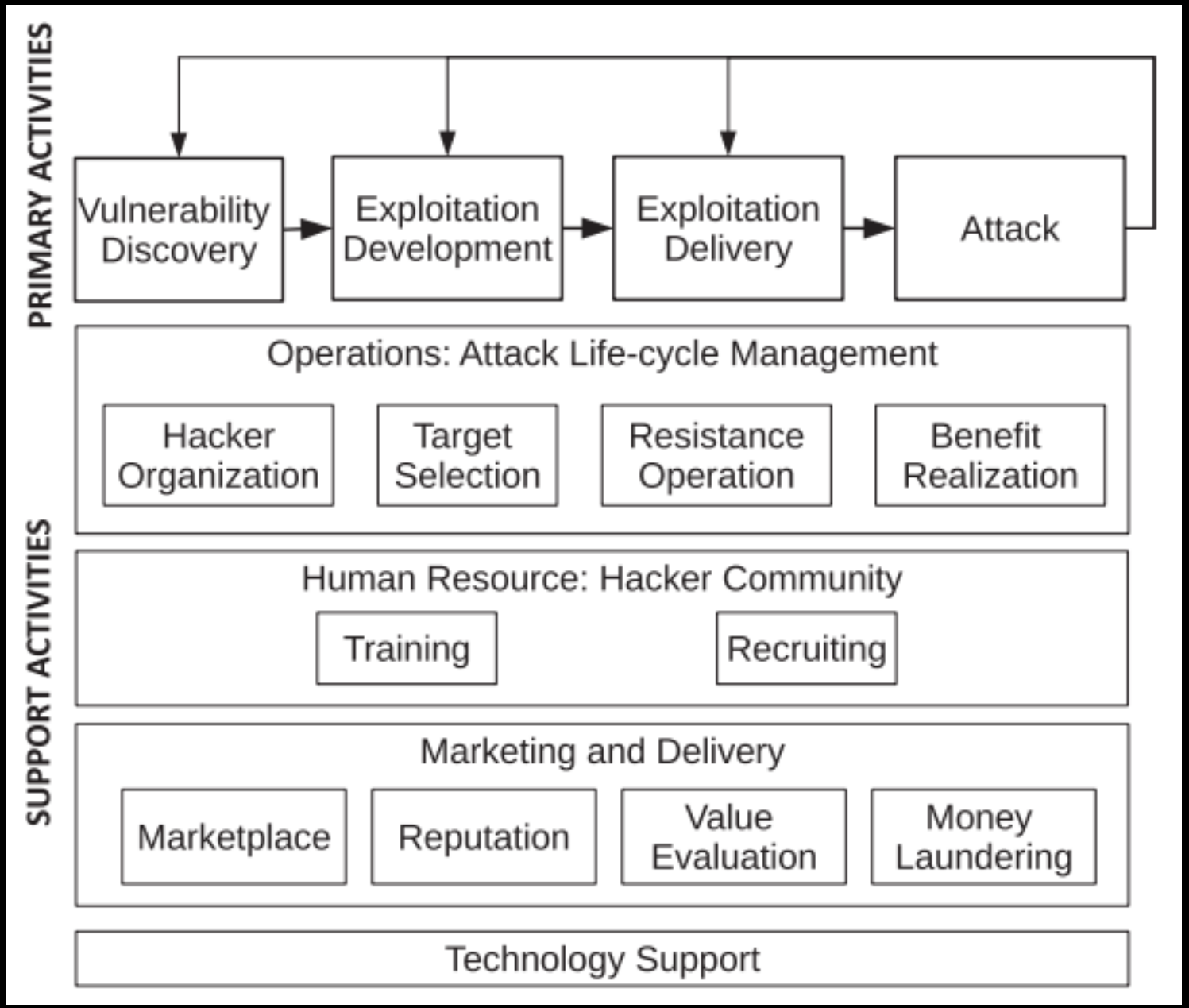
- **spear-fishing emails**,
- an **exploit kit targeting vulnerabilities**,
- the KillDisk, a destructive **data-wiping utility**, and
- an **SSH backdoor** to maintain persistent access,

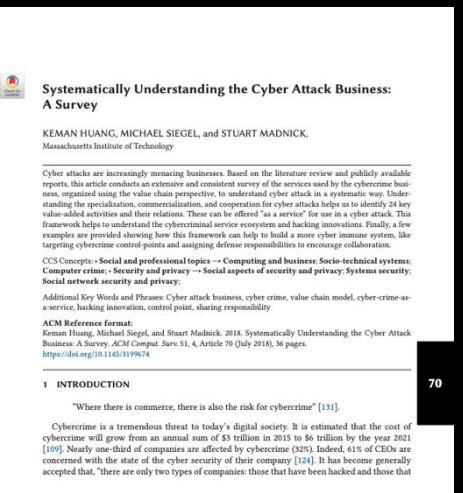
were used in tandem to successfully break into the system.

In the second step of the same attack, **malicious firmware** developed based on domain knowledge collected from the distribution management system and was tested by the **simulated power grid system**,

was uploaded to the system and to attack the ICS components.

Cybercriminal Value Chain Model





Systematically Understanding the Cyber Attack Business: A Survey

KEMAN HUANG, MICHAEL SIEGEL, and STUART MADNICK,
Massachusetts Institute of Technology

Cyber attacks are increasingly menacing businesses. Based on the literature review and publicly available reports, this article conducts an extensive and consistent survey of the services used by the cybercrime business, organized using the value chain perspective, to understand cyber attack in a systematic way. Understanding the specialization, commercialization, and cooperation for cyber attacks helps us to identify 24 key value-added activities and their relations. These can be offered "as a service" for use in a cyber attack. This framework helps to understand the cybercriminal service ecosystem and hacking innovations. Finally, a few examples are provided showing how this framework can help to build a more cyber immune system, like targeting cybercrime control-points and assigning defense responsibilities to encourage collaboration.

CCS Concepts: • **Social and professional topics → Computing and business**; **Socio-technical systems**; **Computer crime**; • **Security and privacy → Social aspects of security and privacy**; **Systems security**; **Social network security and privacy**;

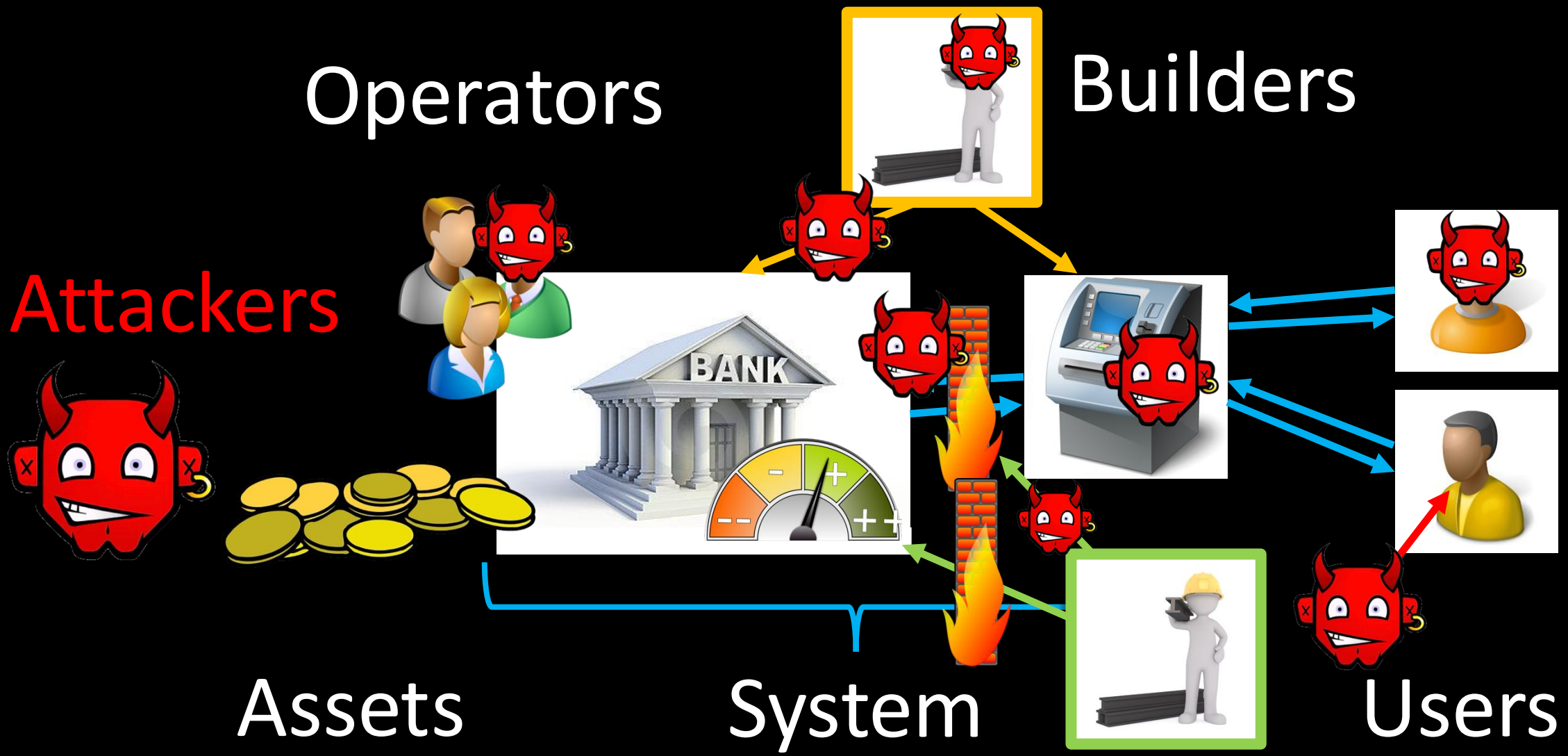
Additional Key Words and Phrases: Cyber attack business, cyber crime, value chain model, cyber-crime-as-a-service, hacking innovation, control point, sharing responsibility

ACM Reference format:
Keman Huang, Michael Siegel, and Stuart Madnick. 2018. Systematically Understanding the Cyber Attack Business: A Survey. *ACM Comput. Surv.* 51, 4, Article 70 (July 2018), 36 pages.
https://doi.org/10.1145/3199674

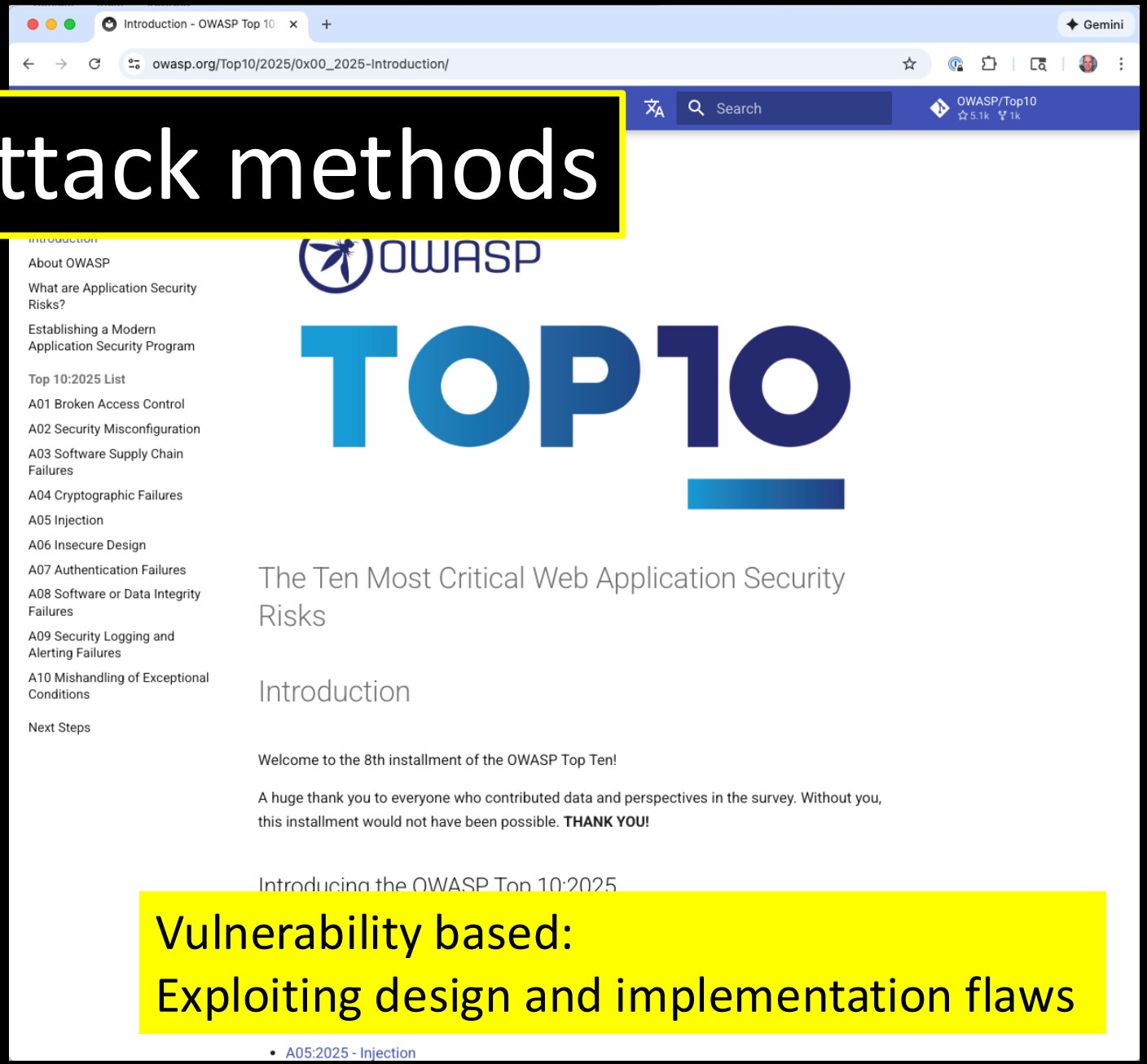
1 INTRODUCTION

"Where there is commerce, there is also the risk for cybercrime" [131].

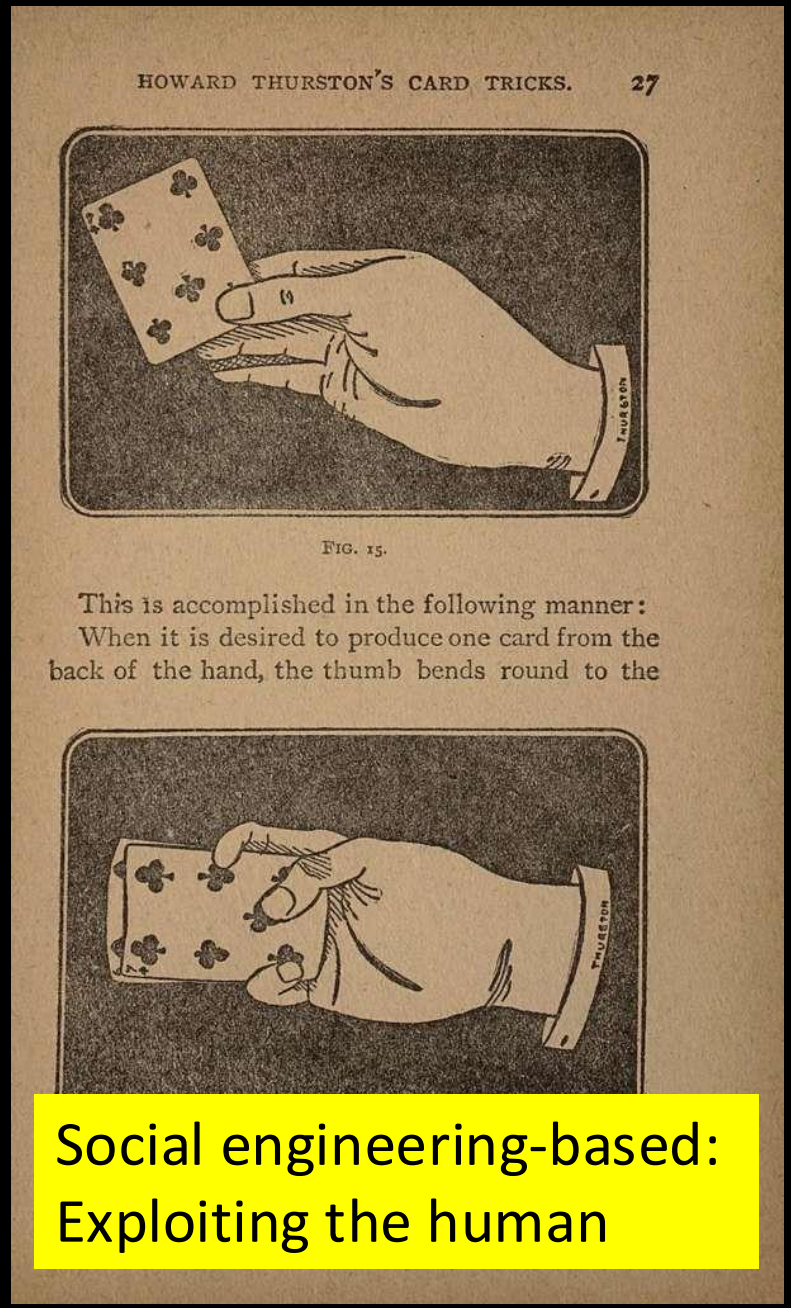
Cybercrime is a tremendous threat to today's digital society. It is estimated that the cost of cybercrime will grow from an annual sum of $3 trillion in 2015 to $6 trillion by the year 2021 [109]. Nearly one-third of companies are affected by cybercrime (32%). Indeed, 61% of CEOs are concerned with the state of the cyber security of their company [124]. It has become generally accepted that, "there are only two types of companies: those that have been hacked and those that


Operators
Builders
Attackers
BANK
Assets
System
Users
Cyber Builders

Attack methods



Vulnerability based:
Exploiting design and implementation flaws



Social engineering-based:
Exploiting the human

Builders

Operators

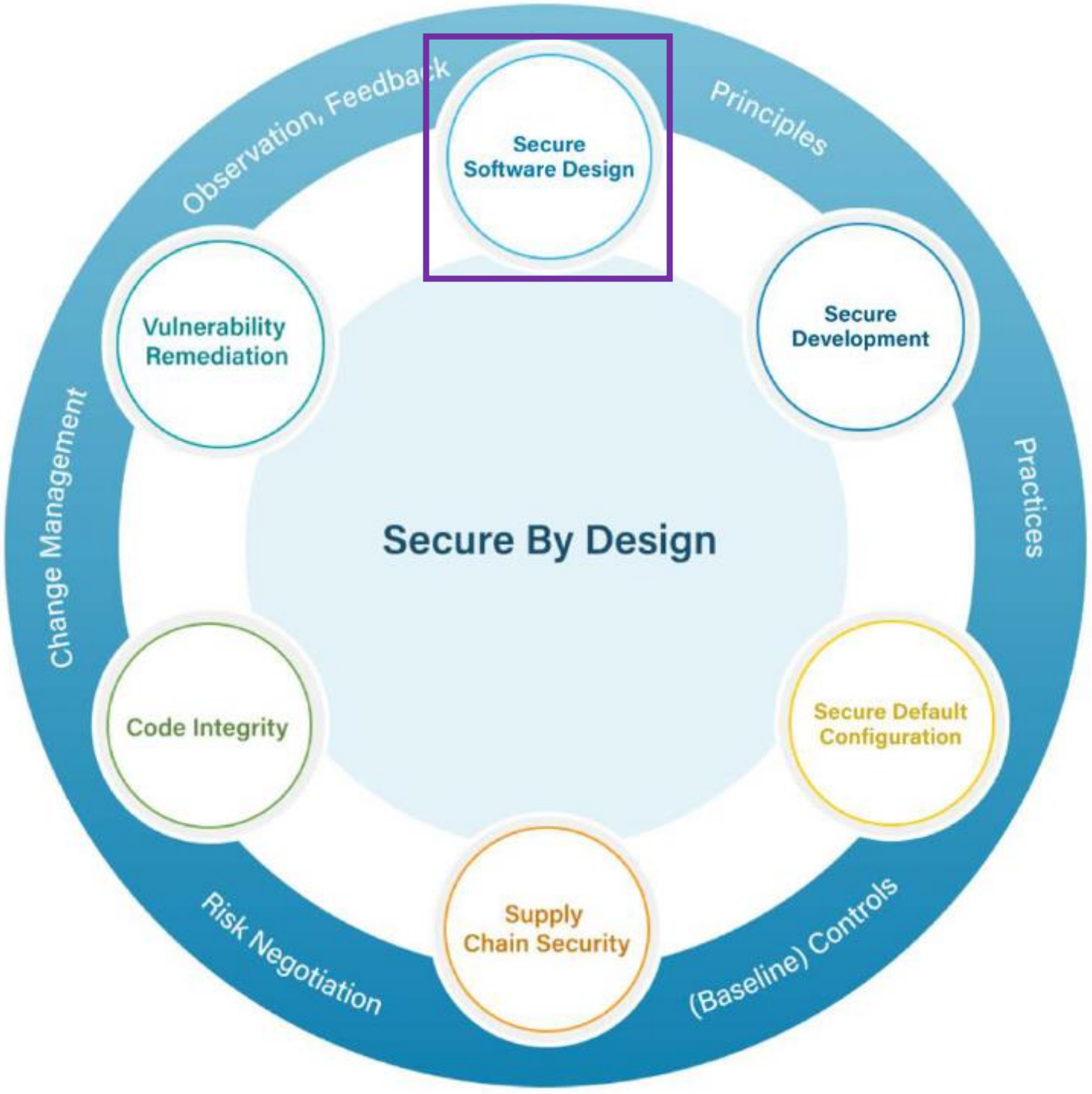
Users

Cyber Builders

(also provides
tools for Builders)

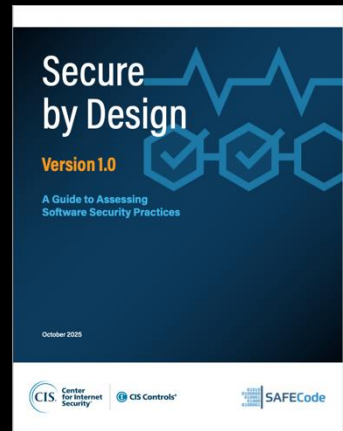
Builders

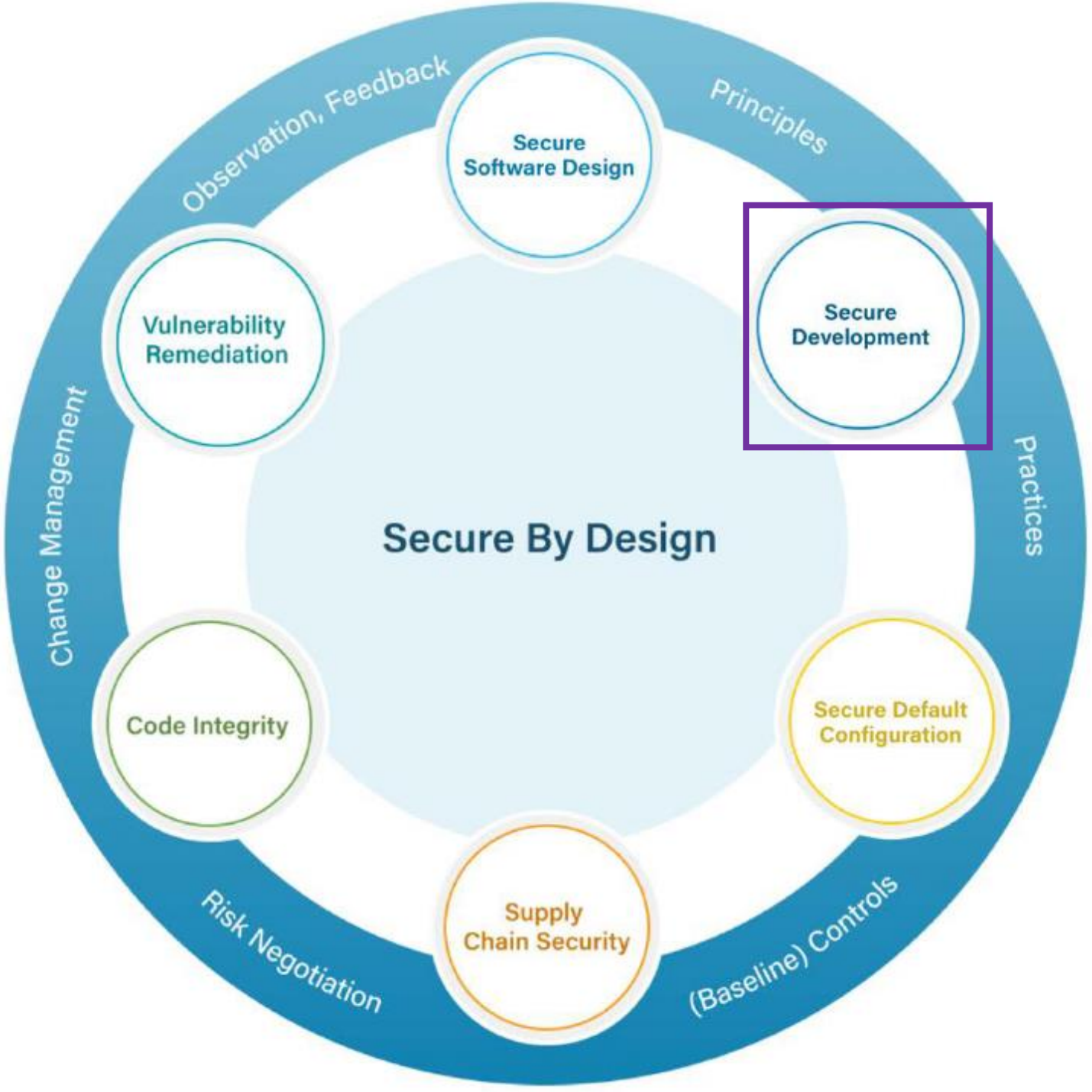
- Create bespoke (first-party) software and services …
  - Developers within a bank, e.g., Capitol One
- … and commodity (third party) software or services
  - Software: Android OS, Linux, Google Chrome, Microsoft Word, …
  - Services: AWS, Azure, Workday, Google Suite, …
- Responsible for the product, and its security
  - Often rely on collaborating dev and security engineering teams



Builders

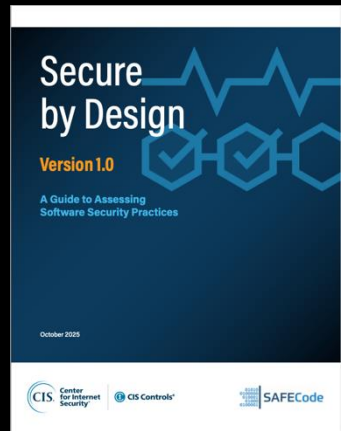
- Threat modeling
- Secure architectural design

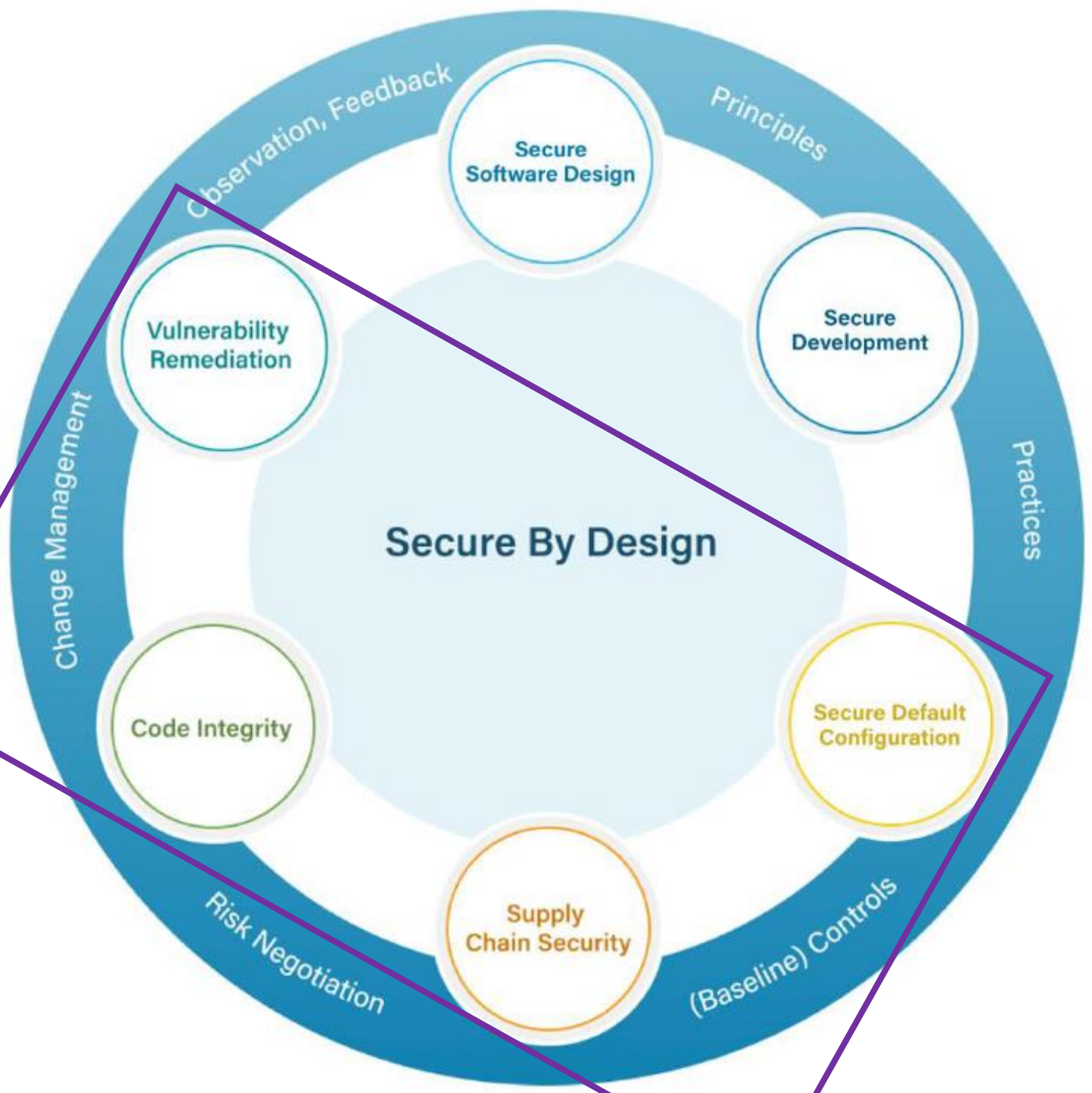




Builders

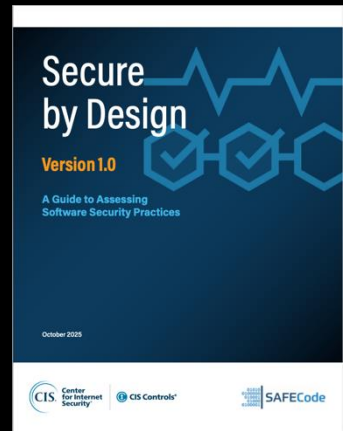
- Threat modeling
- Secure architectural design
- Secure programming
- Security testing (e.g., fuzzing)





Builders

- Threat modeling
- Secure architectural design
- Secure programming
- Security testing (e.g., fuzzing)
- Secure deployment and management



Builders

Operators

Users

Cyber Builders

(also provides
tools for Builders)

Users

- Target (direct or indirect) of attackers
- Participant in system security
  - Setting passwords, setting a security policy, not clicking phishing links, …
- … but not necessarily motivated or capable
  - May share or reuse passwords, set over-permissive policies, click suspicious links, ignore security training, …

Passwords: Security v. memorability

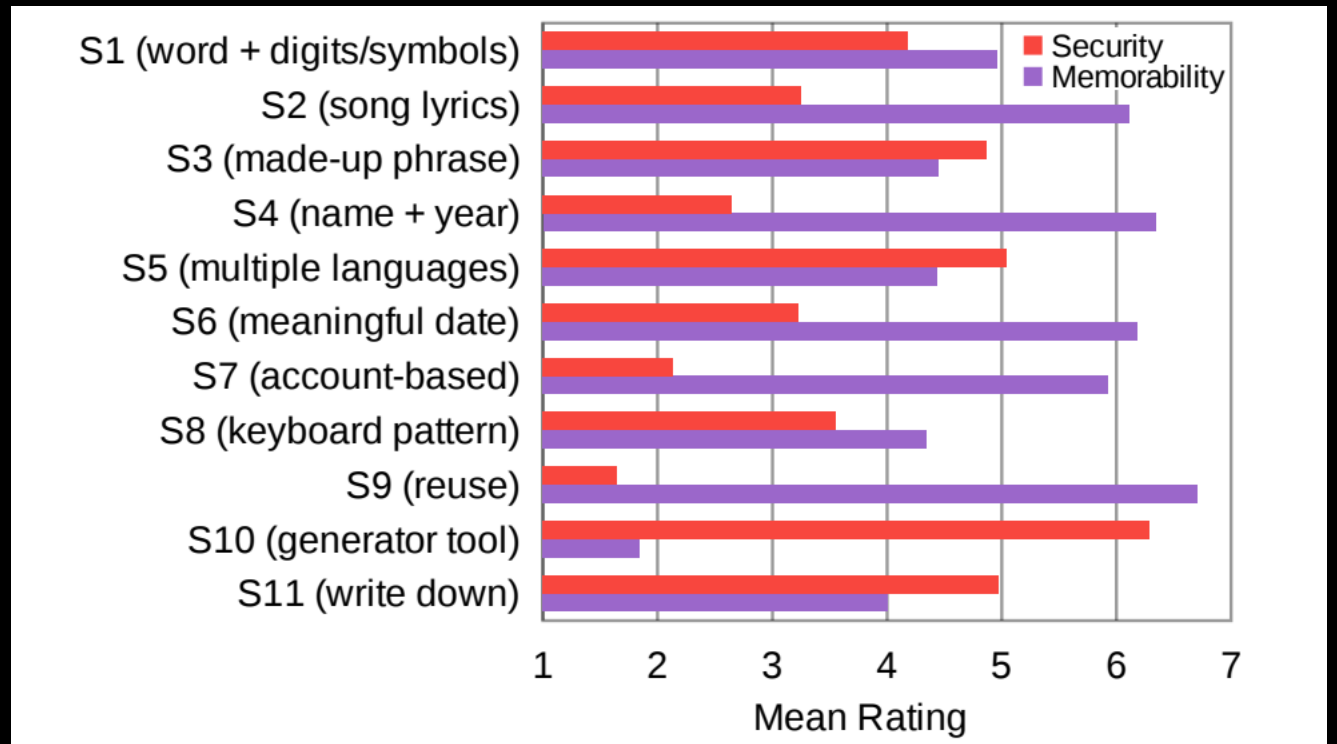
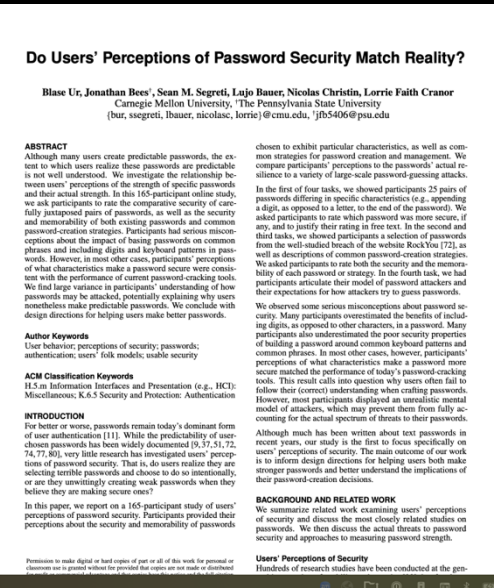


Figure 3. Mean ratings for the security and memorability of the 11 password-creation strategies.



Do Users' Perceptions of Password Security Match Reality?

Blase Ur, Jonathan Bees[†], Sean M. Segreti, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor
Carnegie Mellon University, [†]The Pennsylvania State University
{bur, ssegreti, lbauer, nicolasc, lorrie}@cmu.edu, [†]jfb5406@psu.edu

ABSTRACT

Although many users create predictable passwords, the extent to which users realize these passwords are predictable is not well understood. We investigate the relationship between users' perceptions of the strength of specific passwords and their actual strength. In this 165-participant online study, we ask participants to rate the comparative security of carefully juxtaposed pairs of passwords, as well as the security and memorability of both existing passwords and common password-creation strategies. Participants had serious misconceptions about the impact of basing passwords on common phrases and including digits and keyboard patterns in passwords. However, in most other cases, participants' perceptions of what characteristics make a password secure were consistent with the performance of current password-cracking tools. We find large variance in participants' understanding of how passwords may be attacked, potentially explaining why users nonetheless make predictable passwords. We conclude with design directions for helping users make better passwords.

Author Keywords

User behavior; perceptions of security; passwords; authentication; users' folk models; usable security

ACM Classification Keywords

H.5.m Information Interfaces and Presentation (e.g., HCI): Miscellaneous; K.6.5 Security and Protection: Authentication

INTRODUCTION

For better or worse, passwords remain today's dominant form of user authentication [11]. While the predictability of user-chosen passwords has been widely documented [9,37,51,72, 74,77,80], very little research has investigated users' perceptions of password security. That is, do users realize they are selecting terrible passwords and choose to do so intentionally, or are they unwittingly creating weak passwords when they believe they are making secure ones?

In this paper, we report on a 165-participant study of users' perceptions of password security. Participants provided their perceptions about the security and memorability of passwords chosen to exhibit particular characteristics, as well as common strategies for password creation and management. We compare participants' perceptions to the passwords' actual resilience to a variety of large-scale password-guessing attacks.

In the first of four tasks, we showed participants 25 pairs of passwords differing in specific characteristics (e.g., appending a digit, as opposed to a letter, to the end of the password). We asked participants to rate which password was more secure, if any, and to justify their rating in free text. In the second and third tasks, we showed participants a selection of passwords from the well-studied breach of the website RockYou [72], as well as descriptions of common password-creation strategies. We asked participants to rate both the security and the memorability of each password or strategy. In the fourth task, we had participants articulate their model of password attackers and their expectations for how attackers try to guess passwords.

We observed some serious misconceptions about password security. Many participants overestimated the benefits of including digits, as opposed to other characters, in a password. Many participants also underestimated the poor security properties of building a password around common keyboard patterns and common phrases. In most other cases, however, participants' perceptions of what characteristics make a password more secure matched the performance of today's password-cracking tools. This result calls into question why users often fail to follow their (correct) understanding when crafting passwords. However, most participants displayed an unrealistic mental model of attackers, which may prevent them from fully accounting for the actual spectrum of threats to their passwords.

Although much has been written about text passwords in recent years, our study is the first to focus specifically on users' perceptions of security. The main outcome of our work is to inform design directions for helping users both make stronger passwords and better understand the implications of their password-creation decisions.

BACKGROUND AND RELATED WORK

We summarize related work examining users' perceptions of security and discuss the most closely related studies on passwords. We then discuss the actual threats to password security and approaches to measuring password strength.

Users' Perceptions of Security

Hundreds of research studies have been conducted at the gen-

Password reuse: Vector of attack

- Guessed 32% of passwords in historical DB by leveraging reuse
  - As compared to 6.5% without considering reuse
  - 35.5% of valid guesses were for current passwords
- Of those guessed by reuse
  - 54.7% were verbatim reuse, vs. 45.3% based on tweaks
- Vulnerability is real
  - Some historical observed exploits seemed to coincide with data breaches
  - Passwords were vulnerable for long after a breach (median of 5 years)

A Two-Decade Retrospective Analysis of a University's Vulnerability to Attacks Exploiting Reused Passwords

Alexandra Nisenoff[†*], Maximilian Golla[†‡], Miranda Wei[†*], Juliette Hainline[†], Hayley Szymanek[†], Annika Braun[†], Annika Hildebrandt[†], Blair Christensen[†], David Langenberg[†], Blase Ur[†]

† University of Chicago, * Carnegie Mellon University, ‡ Max Planck Institute for Security and Privacy, ⋆ University of Washington

Abstract

Credential-guessing attacks often exploit passwords that were reused across a user's online accounts. To learn how organizations can better protect users, we retrospectively analyzed our university's vulnerability to credential-guessing attacks across twenty years. Given a list of university usernames, we searched for matches in both data breaches from hundreds of websites and a dozen large compilations of breaches. After cracking hashed passwords and tweaking guesses, we successfully guessed passwords for 32.0% of accounts matched to a university email address in a data breach, as well as 6.5% of accounts where the username (but not necessarily the domain) matched. Many of these accounts remained vulnerable for years after the breached data was leaked, and passwords found verbatim in breaches were nearly four times as likely to have been exploited (i.e., suspicious account activity was observed) than tweaked guesses. Over 70 different data breaches and various username-matching strategies bootstrapped correct guesses. In surveys of 40 users whose passwords we guessed, many users were unaware of the risks to their university account or that their credentials had been breached. This analysis of password reuse at our university provides pragmatic advice for organizations to protect accounts.

1 Introduction

Despite their disadvantages, passwords remain widely used for authentication [7]. Organizations must protect against large-scale attacks on users' passwords. An adversary may leverage **reused passwords**—when the same individual picks similar or identical passwords for different services [10, 80] to cope with having to remember numerous passwords [16]. If any one of these services suffers a data breach, attackers typically try to log into another service with the same email address alongside a password that is either the same as the leaked password, or tweaked in small ways. Such credential-stuffing attacks are this paper's focus. Additionally, attackers

The ability to conduct attacks that exploit reused password has increased as hundreds of websites have had their password databases stolen and leaked over the last decade [34]. We term the breach of a single service an **individual service breach**. In recent years, hackers have also packaged credentials from many different services into **breach compilations** containing hundreds of millions or even billions of credentials [24].

To protect an organization against attacks exploiting common passwords, system administrators can institute straightforward blocklists [25, 70]. Protecting an organization from reused passwords, however, is far more complex. A vulnerable password is specific to one user based on their credentials on other sites at any past or future time. Furthermore, prospective attackers often have far more information than system administrators. Attackers may know about a successful breach that system administrators may not hear about for years, or ever. Further, attackers may pool resources to crack hashes and reveal the plaintext needed for an attack, while the system administrator may be left only with uncracked hashes [9].

In recent years, researchers and practitioners have developed compromised-credential-checking tools to try to defend users. For instance, Chrome [73], Firefox [55], and Safari [12] notify users if their passwords appear in a data breach. The Have I Been Pwned (**HIBP**) service [32], itself integrated with 1Password [13], enables users to check for their appearance in a data breach. Supporting these efforts, academic work has proposed protocols that underpin compromised-credential-checking tools [40, 43, 44, 59, 82, 83] and sought to improve the usability of data breach notifications [22, 31, 53, 79, 90, 92].

Despite prior work, many questions remain for system administrators trying to protect their organizations from attacks exploiting reused passwords. For what amount of time are accounts vulnerable? Out of hundreds of data breaches, how important is it to account for them all? Should defenders devote resources to trying to crack hashes to protect users? Is it sufficient to look for matching email addresses, or should they also search for matching usernames? How often do attackers appear to have exploited reused passwords, and what factors

Builders

Operators

Users

Cyber Builders

(also provides
tools for Builders)

Operators

- Manage and operate systems for a user community
  - Examples: Companies that have an online presence, nonprofits such as universities and social services, and on-line service providers like Workday
  - In addition to core services/systems they may provide, they maintain internal network, email, personnel and financial records, etc.
- Ultimately responsible for cybersecurity: prevention, detection, mitigation, response, recovery
  - Many technologies for these. Challenge: How to decide which to use in an evidence-based manner?

A naïve model relating loss to security level

- Simple regression (blue line): more security implies more losses?!
- Problem: Confounding variables (especially threat level)

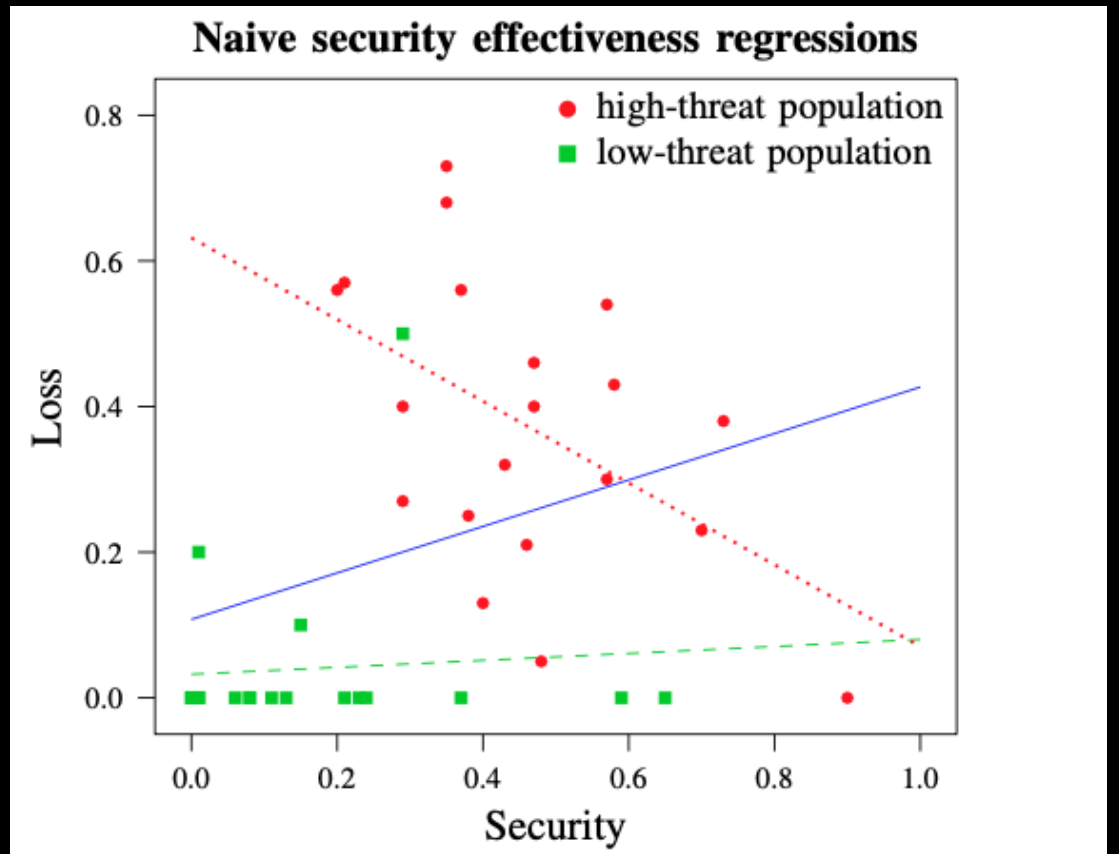
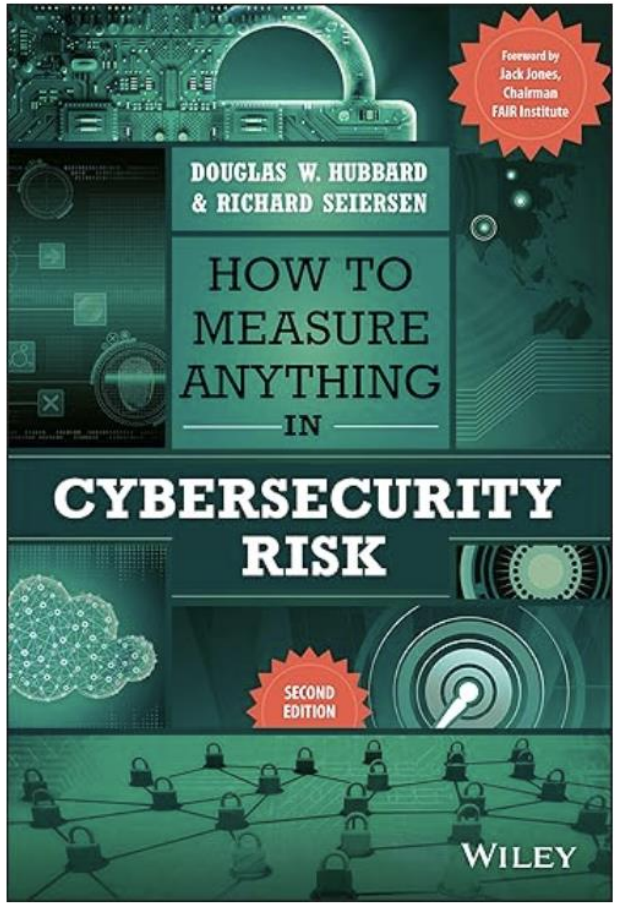


Fig. 1. The solid blue line fails to account for threat level, which may lead the high-threat population to under estimate the effectiveness of security.

SoK: Quantifying Cyber Risk

Daniel W. Woods
University of Innsbruck
Innsbruck, Austria

Rainer Böhme
University of Innsbruck
Innsbruck, Austria



How to Measure Anything in Cybersecurity Risk 2nd Edition

by Douglas W. Hubbard (Author), Richard Seiersen (Author)

4.5 ★★★★½ 88 ratings

See all formats and editions

A start-to-finish guide for realistically measuring cybersecurity risk

In the newly revised *How to Measure Anything in Cybersecurity Risk, Second Edition*, a pioneering information security professional and a leader in quantitative analysis methods delivers yet another eye-opening text applying the quantitative language of risk analysis to cybersecurity. In the book, the authors demonstrate how to quantify uncertainty and shed light on how to measure seemingly intangible goals. It's a practical guide to improving risk assessment with a straightforward and simple framework.

Advanced methods and detailed advice for a variety of use cases round out the book, which also includes:

- A new "Rapid Risk Audit" for a first quick quantitative risk assessment.
- New research on the real impact of reputation damage

with other portfolios. The aggregation process is typically some form of invented mathematics unfamiliar to actuaries, statisticians, and mathematicians.

Just over 50% of respondents plot risks on a two-dimensional matrix. In this approach, "likelihood" and "impact" will be rated subjectively, perhaps on a 1 to 5 scale, and those two values will be used to plot a particular risk on a matrix (variously called a "risk matrix," "heat map," "risk map," etc.). The matrix—similar to the one shown in Figure 1.1—is then often further divided into sections of low, medium, and high risk. Events with high likelihood and high impact would be in the upper-right "high risk" corner, while those with low likelihood and low impact would be in the opposite "low risk" corner. The idea is that the higher the score, the more important something is and the sooner you should address it. You may intuitively think such an approach is reasonable, and if you thought so, you would be in good company.

Various versions of scores and risk maps are endorsed and promoted by several major organizations, standards, and frameworks such as the National Institute of Standards and Technology (NIST), the International Standards Organization (ISO), MITRE.org, and the Open Web Application Security Project (OWASP). Most organizations with a cybersecurity function claim at least one of these as part of their framework for assessing risk. In fact, most major software organizations such as Oracle, Microsoft, and Adobe rate their vulnerabilities using a NIST-supported scoring system called the "Common Vulnerability Scoring System" (CVSS). Many security solutions also include CVSS ratings, be it for vulnerability and/or attack related. While the control recommendations made by many of these frameworks are good,

			Impact				
			Negligible	Minor	Moderate	Critical	Catastrophic
			1	2	3	4	5
Likelihood	Frequent	5	Medium	Medium	High	High	High
	Likely	4	Medium	Medium	Medium	High	High
	Occasional	3	Low	Medium	Medium	Medium	High
	Seldom	2	Low	Low	Medium	Medium	Medium
	Improbable	1	Low	Low	Low	Medium	Medium

FIGURE 1.1 The Familiar Risk Matrix (aka Heat Map or Risk Map)

it's how we are guided to prioritize risk management on an enterprise scale that is amplifying risk.

Literally hundreds of security vendors and even standards bodies have come to adopt some form of scoring system including the risk matrix. Indeed, scoring approaches and risk matrices are at the core of the security industry's risk management approaches.

In all cases, they are based on the idea that such methods are beneficial to some degree. That is, they are assumed to be at least an improvement over not using such a method. As one of the standards organizations has put it, rating risk this way is adequate:

> *Once the tester has identified a potential risk and wants to figure out how serious it is, the first step is to estimate the* likelihood. *At the highest level, this is a rough measure of how likely this particular vulnerability is to be uncovered and exploited by an attacker. It is not necessary to be over-precise in this estimate.* Generally, identifying whether the likelihood is low, medium, or high is sufficient. *(emphasis added)*
>
> —OWASP[20]

Does this last phrase, stating "low, medium, or high is sufficient," need to be taken on faith? Considering the critical nature of the decisions such methods will guide, we argue that it should not. This is a testable hypothesis, and it actually *has been* tested in many different ways. The growing trends of cybersecurity attacks alone indicate it might be high time to try something else.

So, let's be clear about our position on current methods: *They are a failure. They do not work.* A thorough investigation of the research on these methods and decision-making methods in general indicates the following (all of this will be discussed in detail in later chapters):

- There is no evidence that the types of scoring and risk matrix methods widely used in cybersecurity improve judgment.
- On the contrary, there is evidence these methods add noise and error to the judgment process. One researcher we will discuss more—Tony Cox—goes as far as to say they can be "worse than random."
- Any appearance of "working" is probably a type of "analysis placebo." That is, a method may make you feel better even though the activity provides no measurable improvement in estimating risks (or even adds error).
- There is overwhelming evidence in published research that quantitative, probabilistic methods are an improvement over unaided expert intuition.

Builders

Operators

Users

Cyber Builders

(also provides
tools for Builders)

Cyber Builders

- Build products and services to enhance cybersecurity
  - Usability is extremely important: May reduce security benefits to increase it!
- Customer: Operators
  - Firewalls, EDR (Endpoint Detection and Response), email security, pen testing services, threat intelligence, …
- Customer: Users
  - Password managers, antivirus, cloud-hosted encrypted backups, …
- Customer: Builders
  - Code management, dependency tracking, code analysis, automated testing, …
- Customer: Attacker (!!)
  - Use builder services (code analysis), learn from defenses (malware scans)

For Operators: Key technologies and activities

- Preventing and detecting attacks
  - Antivirus
  - Firewalls
  - Host-based intrusion detection/prevention (HIDS/HIPS)
  - Endpoint detection and response (EDR)
  - Security Information and Event Management (SIEM)
- Mitigating effects of an attack
  - Containerization, cloud backups, MAC
- Threat intelligence: leverage security researchers, CERTs and ISACs, media/journalists

For Users: Password managers

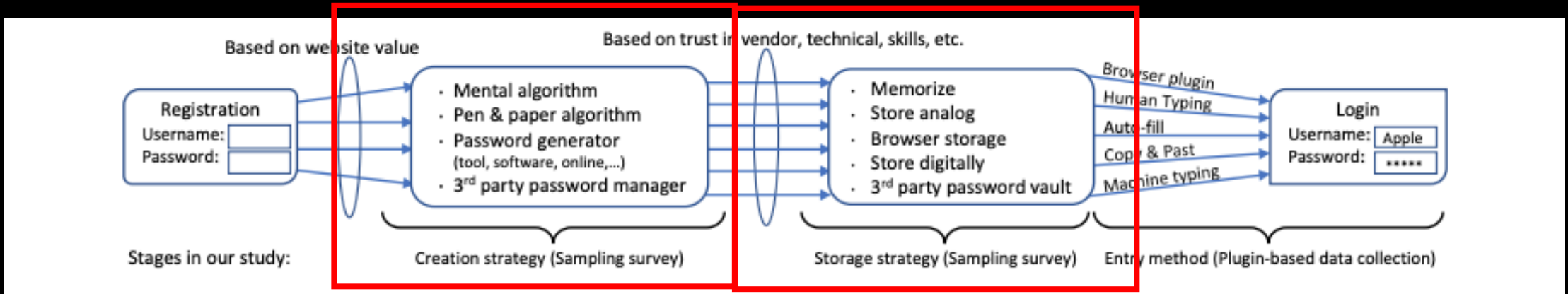


Figure 1: Users' strategies for password creation and storage plus the stages of our study to investigate managers' influence.

- Password **creation** – with algorithm, by hand, using service, …
- Password **storage** – memorized, in file, in service, …
- Password **entry** – typed, cut&paste, auto-filled, …

Better managed than memorized?
Studying the Impact of Managers on Password Strength and Reuse

Sanam Ghorbani Lyastani, *CISPA, Saarland University*
Michael Schilling, *Saarland University*
Sascha Fahl, *Ruhr-University Bochum*
Michael Backes, *CISPA Helmholtz Center i.G.*
Sven Bugiel, *CISPA Helmholtz Center i.G.*

Abstract

Despite their well-known security problems, passwords are still the incumbent authentication method for virtually all online services. To remedy the situation, users are very often referred to password managers as a solution to the password reuse and weakness problems. However, to date the actual impact of password managers on password strength and reuse has not been studied systematically.

We provide the first large-scale study of the password managers' influence on users' real-life passwords. By combining qualitative data on users' password creation and management strategies, collected from 476 participants of an online survey, with quantitative data (incl. password metrics and entry methods) collected in situ with a browser plugin from 170 users, we were able to gain a more complete picture of the factors that influence our participants' password strength and reuse. Our approach allows us to quantify for the first time that password managers indeed influence the password security, however, whether this influence is beneficial or aggravating existing problems depends on the users' strategies and how well the manager supports the users' password management right from the time of password creation. Given our results, we think research should further investigate how managers can better support users' password strategies in order to improve password security as well as stop aggravating the existing problems.

1 Introduction

For several decades passwords prevail as the default authentication scheme for virtually all online services [44, 11, 30]. At the same time, research has again and again demonstrated that passwords perform extremely poor in terms of security [48]. For instance, various attacks exploit that humans fail to create strong passwords themselves [10, 19, 45, 31, 34]. Even worse, there is an observable trend towards an increasing number of online services that users register to. This increasing number of required passwords in combination with the limited human capacity to remember passwords leads to the bad practice of re-using passwords across accounts [26, 51, 16, 66].

In the past, different solutions have been implemented to help users creating stronger passwords, such as password meters and policies, which are also still subject of active research [41, 54, 17, 45, 68]. Among the most often recommended solutions [28, 59, 53, 62, 56] to these problems for end-users is technical support in the form of password management software. Those password managers come built-in to our browsers, as a browser plugin, or as separate applications. Password managers are being recommended as a solution because they fulfill important usability and security aspects at the same time: They store all the users' passwords so the users do not have to memorize them; they can also help users entering their passwords by automatically filling them into log-in forms; and they can also offer help in creating unique, random passwords. By today, there are several examples of third party password managers that fit this description, such as Lastpass [5], 1Password [1], and even seemingly unrelated security software, such as anti-virus [4] solutions.

Unfortunately, it has not been sufficiently studied in the past whether password managers fulfill their promise and indeed have a positive influence on password security or not? To break this question down, we are interested in *1) whether password managers actually store strong passwords that are likely auto-generated by, for instance, password generators, or if they really are just storage where users save their self-made, likely weak passwords?* Further, we are interested whether *2) users, despite using password managers, still reuse passwords across different websites or if do they use the managers' support to maintain a large set of unique passwords for every distinct service?* Prior works [66, 51] that studied password reuse and strength in sites have also considered password managers as factors, but did not find an influence by managers and could not conclusively answer those questions.

Study: Password managers may not help!

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2.62	0.45	5.80	<0.001
em:chrome	0.46	0.16	2.81	<0.01
em:copy/paste	-2.68	0.41	-6.54	<0.001
em:lastpass	-1.05	0.37	-2.86	<0.01
em:unknownplugin	0.76	0.51	1.51	0.13
in-situ:value	-0.13	0.06	-2.01	<0.05
in-situ:strength	-0.21	0.08	-2.50	<0.05
user:entries	0.06	0.02	2.67	<0.01
q9:generator	-1.31	0.40	-3.24	<0.01
q14:memorize	0.22	0.25	0.88	0.38
q14:analog	-0.48	0.24	-1.98	<0.05
q14:digital	-0.18	0.26	-0.70	0.48
q14:pwm	-0.07	0.24	-0.30	0.76

em: Entry method; q9: Creation strategy; q14: Storage strategy; in-situ: Plugin questionnaire

Table 8: Logistic multi-level regression model predicting reuse. Estimates are in relation to manually entered passwords by a human and refer to the corresponding logit transformed odds ratios. Statistically significant predictors are shaded.

- Reuse was significantly influenced by the entry method of the password
  - Odds for reuse were 2.85 times *lower* by LastPass, 14.29 times lower if C&P
  - odds for reuse were 1.65 times *higher* by Chrome auto-fill
- Creation by alg: odds of non-reuse 3.70 times higher
- More passwords → greater odds of reuse
- Higher-value website → lower odds of reuse

Better managed than memorized?
Studying the Impact of Managers on Password Strength and Reuse

Sanam Ghorbani Lyastani, *CISPA, Saarland University*
Michael Schilling, *Saarland University*
Sascha Fahl, *Ruhr-University Bochum*
Michael Backes, *CISPA Helmholtz Center i.G.*
Sven Bugiel, *CISPA Helmholtz Center i.G.*

Abstract

Despite their well-known security problems, passwords are still the incumbent authentication method for virtually all online services. To remedy the situation, users are very often referred to password managers as a solution to the password reuse and weakness problems. However, to date the actual impact of password managers on password strength and reuse has not been studied systematically.

We provide the first large-scale study of the password managers' influence on users' real-life passwords. By combining qualitative data on users' password creation and management strategies, collected from 476 participants of an online survey, with quantitative data (incl. password metrics and entry methods) collected in situ with a browser plugin from 170 users, we were able to gain a more complete picture of the factors that influence our participants' password strength and reuse. Our approach allows us to quantify for the first time that password managers indeed influence the password security, however, whether this influence is beneficial or aggravating existing problems depends on the users' strategies and how well the manager supports the users' password management right from the time of password creation. Given our results, we think research should further investigate how managers can better support users' password strategies in order to improve password security as well as stop aggravating the existing problems.

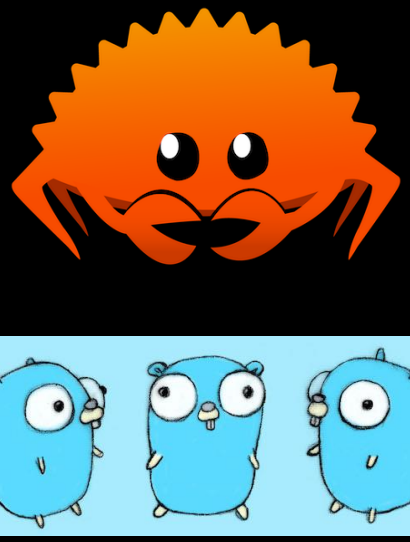
1 Introduction

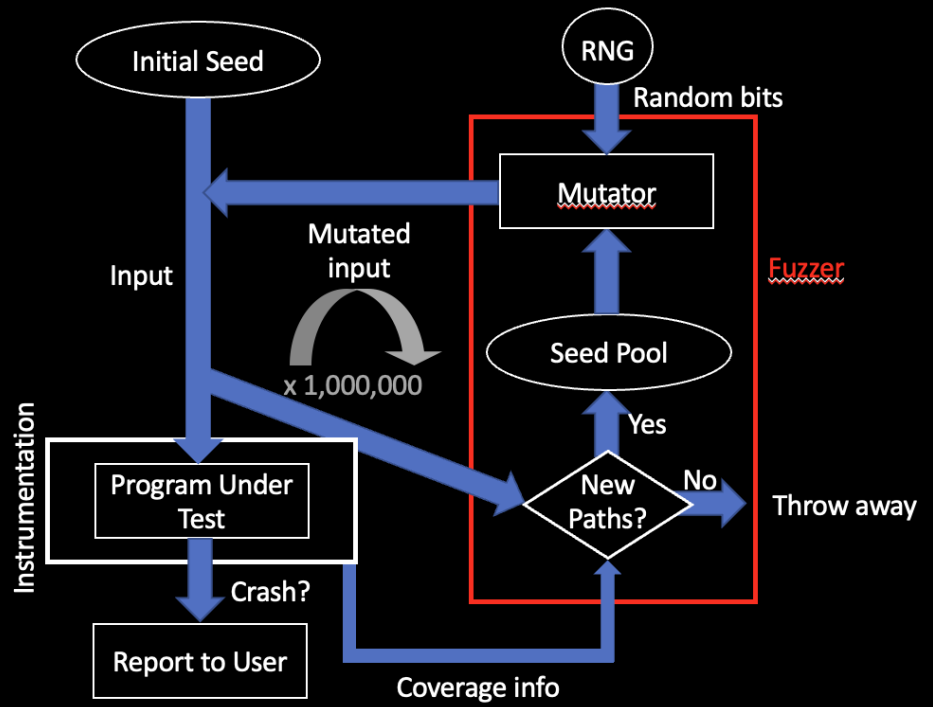
For several decades passwords prevail as the default authentication scheme for virtually all online services [44, 11, 30]. At the same time, research has again and again demonstrated that passwords perform extremely poor in terms of security [48]. For instance, various attacks exploit that humans fail to create strong passwords themselves [10, 19, 45, 31, 34]. Even worse, there is an observable trend towards an increasing number of online services that users register to. This increasing number of required passwords in combination with the limited human capacity to remember passwords leads to the bad practice of re-using passwords across accounts [26, 51, 16, 66].

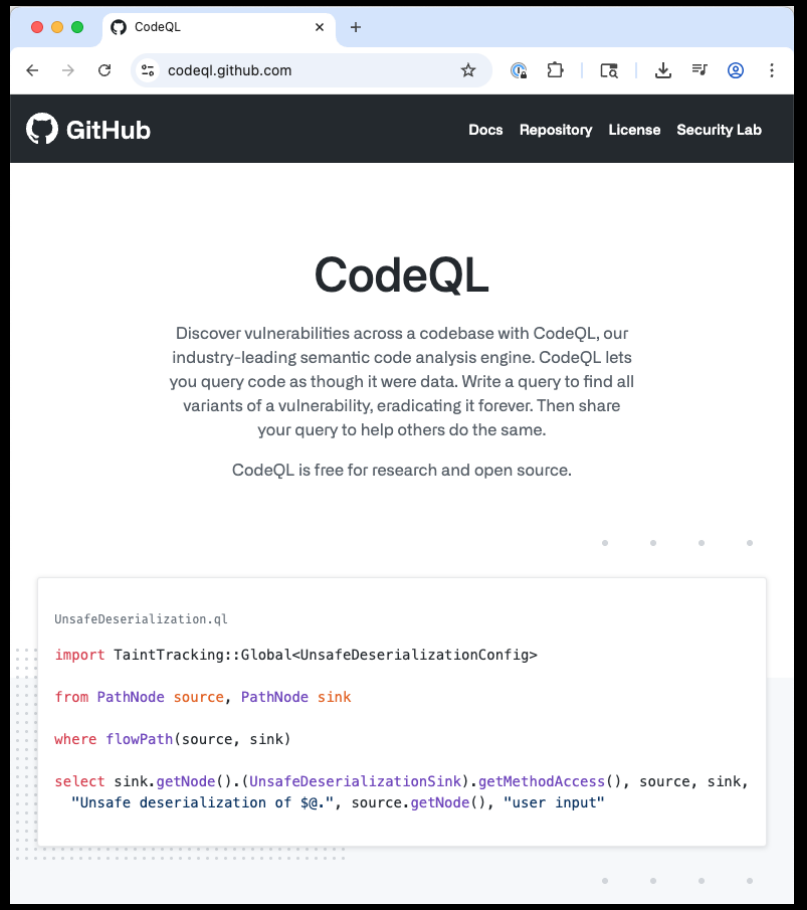
In the past, different solutions have been implemented to help users creating stronger passwords, such as password meters and policies, which are also still subject of active research [41, 54, 17, 45, 68]. Among the most often recommended solutions [28, 59, 53, 62, 56] to these problems for end-users is technical support in the form of password management software. Those password managers come built-in to our browsers, as a browser plugin, or as separate applications. Password managers are being recommended as a solution because they fulfill important usability and security aspects at the same time: They store all the users' passwords so the users do not have to memorize them; they can also help users entering their passwords by automatically filling them into log-in forms; and they can also offer help in creating unique, random passwords. By today, there are several examples of third party password managers that fit this description, such as Lastpass [5], 1Password [1], and even seemingly unrelated security software, such as anti-virus [4] solutions.

Unfortunately, it has not been sufficiently studied in the past whether password managers fulfill their promise and indeed have a positive influence on password security or not? To break this question down, we are interested in *1) whether password managers actually store strong passwords that are likely auto-generated by, for instance, password generators, or if they really are just storage where users save their self-made, likely weak passwords?* Further, we are interested whether *2) users, despite using password managers, still reuse passwords across different websites or if do they use the managers' support to maintain a large set of unique passwords for every distinct service?* Prior works [66, 51] that studied password reuse and strength in situ have also considered password managers as factors, but did not find an influence by managers and could not conclusively answer those questions.

For Builders: Safe PLs, fuzzers, analyzers, …

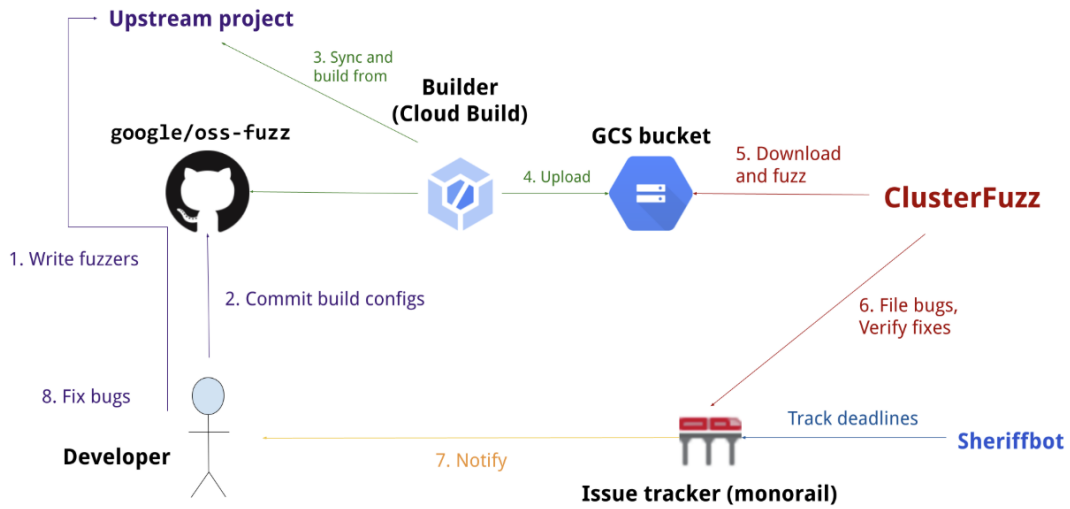






OSS-Fuzz

Overview



Documentation

Read our detailed documentation to learn how to use OSS-Fuzz.

Trophies

As of May 2025, OSS-Fuzz has helped identify and fix over 13,000 vulnerabilities and 50,000 bugs across 1,000 projects.

OSS-Fuzz | Documentation fo

Gemini

google.github.io/oss-fuzz/

OSS-Fuzz

Search OSS-Fuzz

OSS-Fuzz on GitHub

OSS-Fuzz

cture

arted

topics

ading

guidance

es Just the Docs, a
tion theme for Jekyll.

OSS-Fuzz

Fuzz testing is a well-known technique for uncovering programming errors in software. Many of these detectable errors, like buffer overflow, can have serious security implications. Google has found thousands of security vulnerabilities and stability bugs by deploying guided in-process fuzzing of Chrome components, and we now want to share that service with the open source community.

In cooperation with the Core Infrastructure Initiative and the OpenSSF, OSS-Fuzz aims to make common open source software more secure and stable by combining modern fuzzing techniques with scalable, distributed execution. Projects that do not qualify for OSS-Fuzz (e.g. closed source) can run their own instances of ClusterFuzz or ClusterFuzzLite.

We support the libFuzzer, AFL++, Honggfuzz, and Centipede fuzzing engines in combination with Sanitizers, as well as ClusterFuzz, a distributed fuzzer execution environment and reporting tool.

Currently, OSS-Fuzz supports C/C++, Rust, Go, Python and Java/JVM code. Other languages supported by LLVM may work too. OSS-Fuzz supports fuzzing x86_64 and i386 builds.

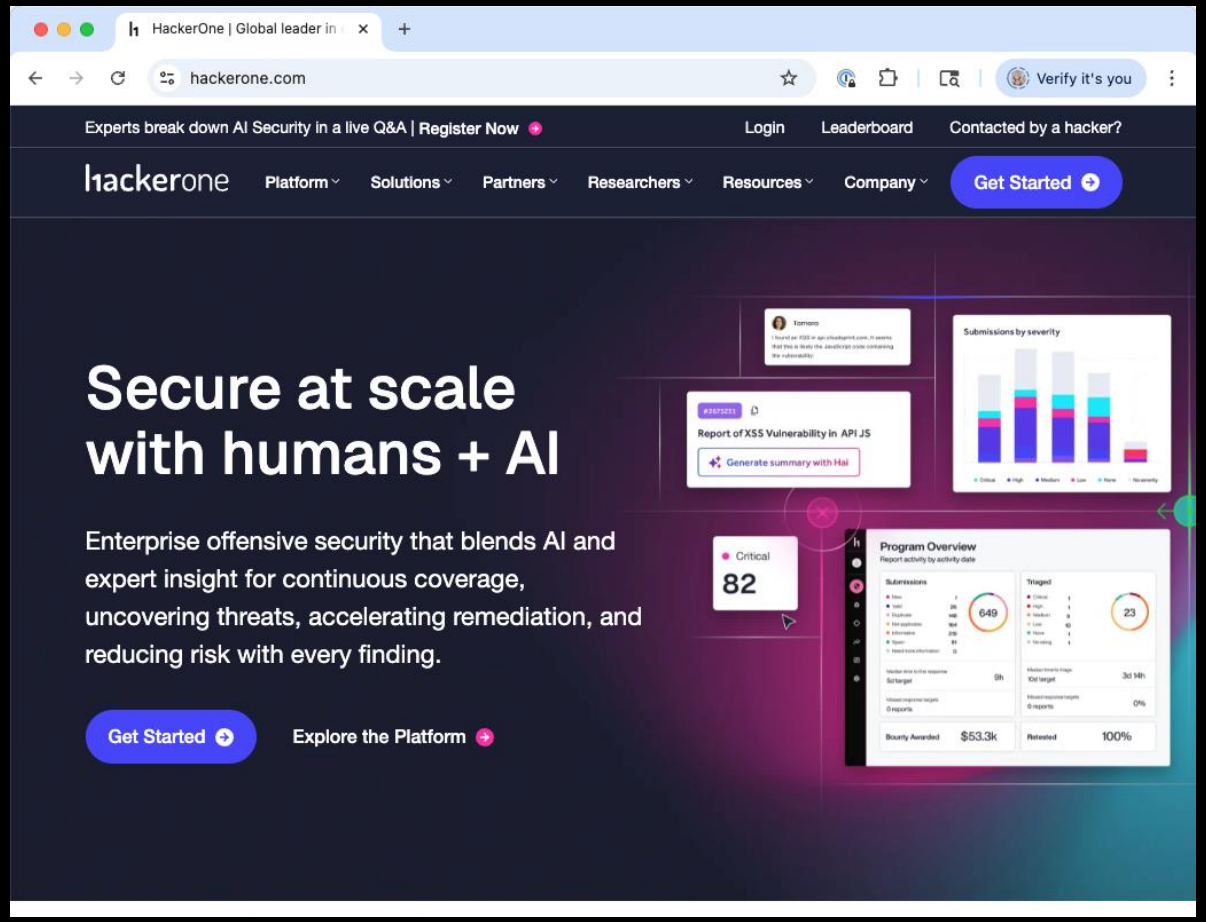
Project history

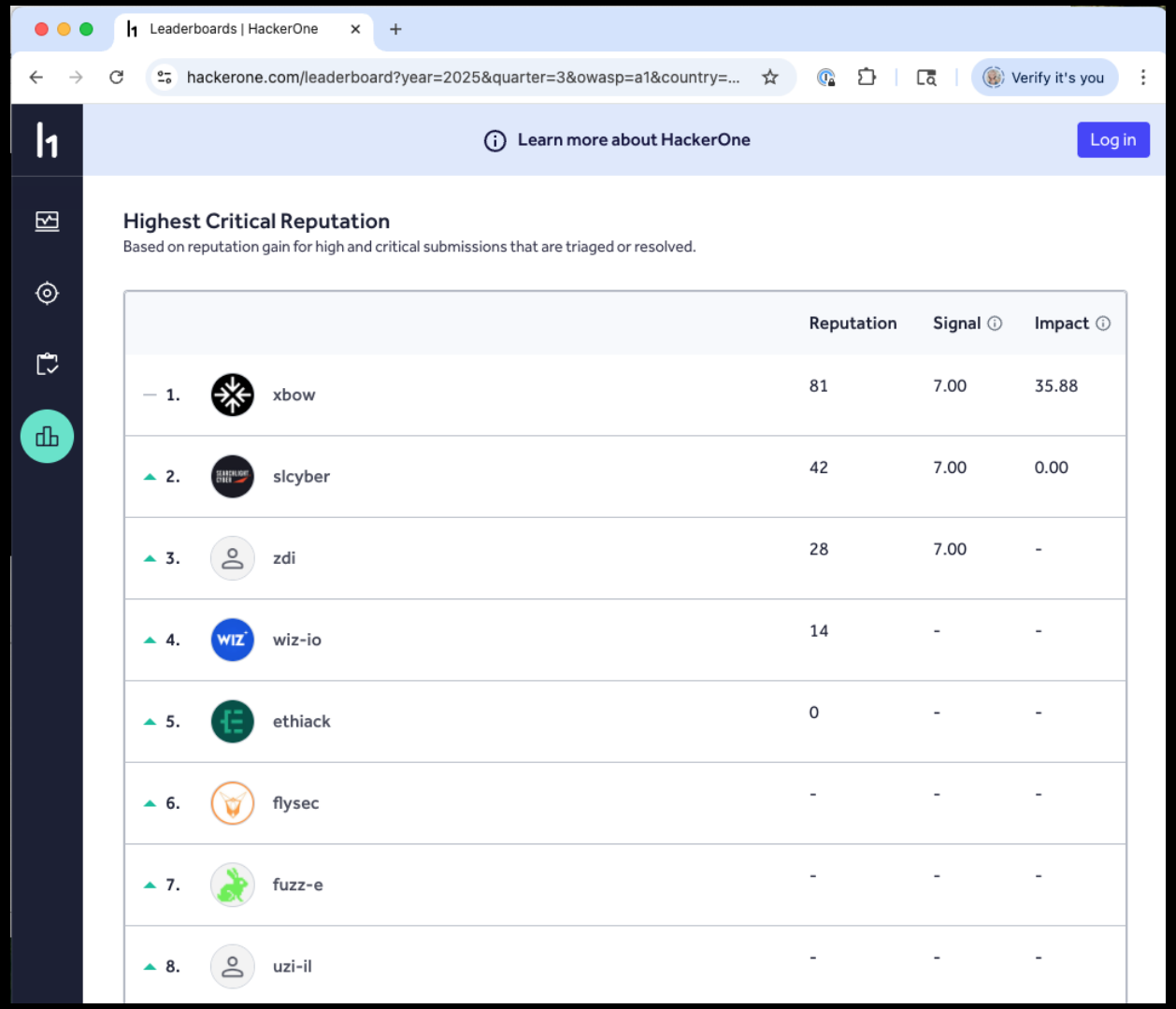
OSS-Fuzz was launched in 2016 in response to the Heartbleed vulnerability, discovered in OpenSSL, one of the most popular open source projects for encrypting web traffic. The vulnerability had the potential to affect almost every internet user, yet was caused by a relatively simple memory buffer overflow bug that could have been detected by fuzzing—that is, by running the code on randomized inputs to intentionally cause unexpected behaviors or crashes. At the time, though, fuzzing was not widely used and was cumbersome for developers, requiring extensive manual effort.

Google created OSS-Fuzz to fill this gap: it's a free service that runs fuzzers for open source

https://en.wikipedia.org/wiki/Buffer_overflow

LLMs and GenAI: Game changers



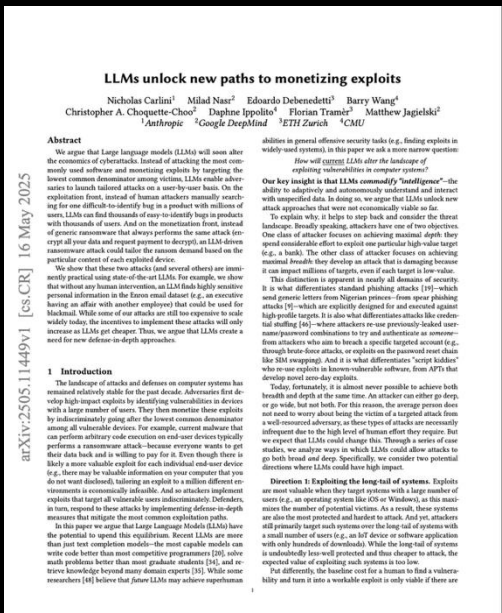


Value proposition to attacker, with LLMs

value = (profit per exploit) * (number impacted)
- (cost to find vulnerability + cost to develop attack)

So, can we use LLMs to do any of the following?

- Increase expected profit
- Increase the number of expected users
- Decrease the cost to find a vulnerability
- Decrease the cost to develop an attack with it



arXiv:2505.11449v1 [cs.CR] 16 May 2025

LLMs unlock new paths to monetizing exploits

Nicholas Carlini[1] Milad Nasr[2] Edoardo Debenedetti[3] Barry Wang[4]
Christopher A. Choquette-Choo[2] Daphne Ippolito[4] Florian Tramèr[3] Matthew Jagielski[2]
[1]Anthropic [2]Google DeepMind [3]ETH Zurich [4]CMU

Abstract

We argue that Large language models (LLMs) will soon alter the economics of cyberattacks. Instead of attacking the most commonly used software and monetizing exploits by targeting the lowest common denominator among victims, LLMs enable adversaries to launch tailored attacks on a user-by-user basis. On the exploitation front, instead of human attackers manually searching for one difficult-to-identify bug in a product with millions of users, LLMs can find thousands of easy-to-identify bugs in products with thousands of users. And on the monetization front, instead of generic ransomware that always performs the same attack (encrypt all your data and request payment to decrypt), an LLM-driven ransomware attack could tailor the ransom demand based on the particular content of each exploited device.

We show that these two attacks (and several others) are imminently practical using state-of-the-art LLMs. For example, we show that without any human intervention, an LLM finds highly sensitive personal information in the Enron email dataset (e.g., an executive having an affair with another employee) that could be used for blackmail. While some of our attacks are still too expensive to scale widely today, the incentives to implement these attacks will only increase as LLMs get cheaper. Thus, we argue that LLMs create a need for new defense-in-depth approaches.

1 Introduction

The landscape of attacks and defenses on computer systems has remained relatively stable for the past decade. Adversaries first develop high-impact exploits by identifying vulnerabilities in devices with a large number of users. They then monetize these exploits by indiscriminately going after the lowest common denominator among all vulnerable devices. For example, current malware that can perform arbitrary code execution on end-user devices typically performs a ransomware attack—because everyone wants to get their data back and is willing to pay for it. Even though there is likely a more valuable exploit for each individual end-user device (e.g., there may be valuable information on your computer that you do not want disclosed), tailoring an exploit to a million different environments is economically infeasible. And so attackers implement exploits that target all vulnerable users indiscriminately. Defenders, in turn, respond to these attacks by implementing defense-in-depth measures that mitigate the most common exploitation paths.

In this paper we argue that Large Language Models (LLMs) have the potential to upend this equilibrium. Recent LLMs are more than just text completion models—the most capable models can write code better than most competitive programmers [20], solve math problems better than most graduate students [34], and retrieve knowledge beyond many domain experts [35]. While some researchers [48] believe that *future* LLMs may achieve superhuman abilities in general offensive security tasks (e.g., finding exploits in widely-used systems), in this paper we ask a more narrow question:

How will current LLMs alter the landscape of exploiting vulnerabilities in computer systems?

Our key insight is that LLMs *commodify "intelligence"*—the ability to adaptively and autonomously understand and interact with unspecified data. In doing so, we argue that LLMs unlock new attack approaches that were not economically viable so far.

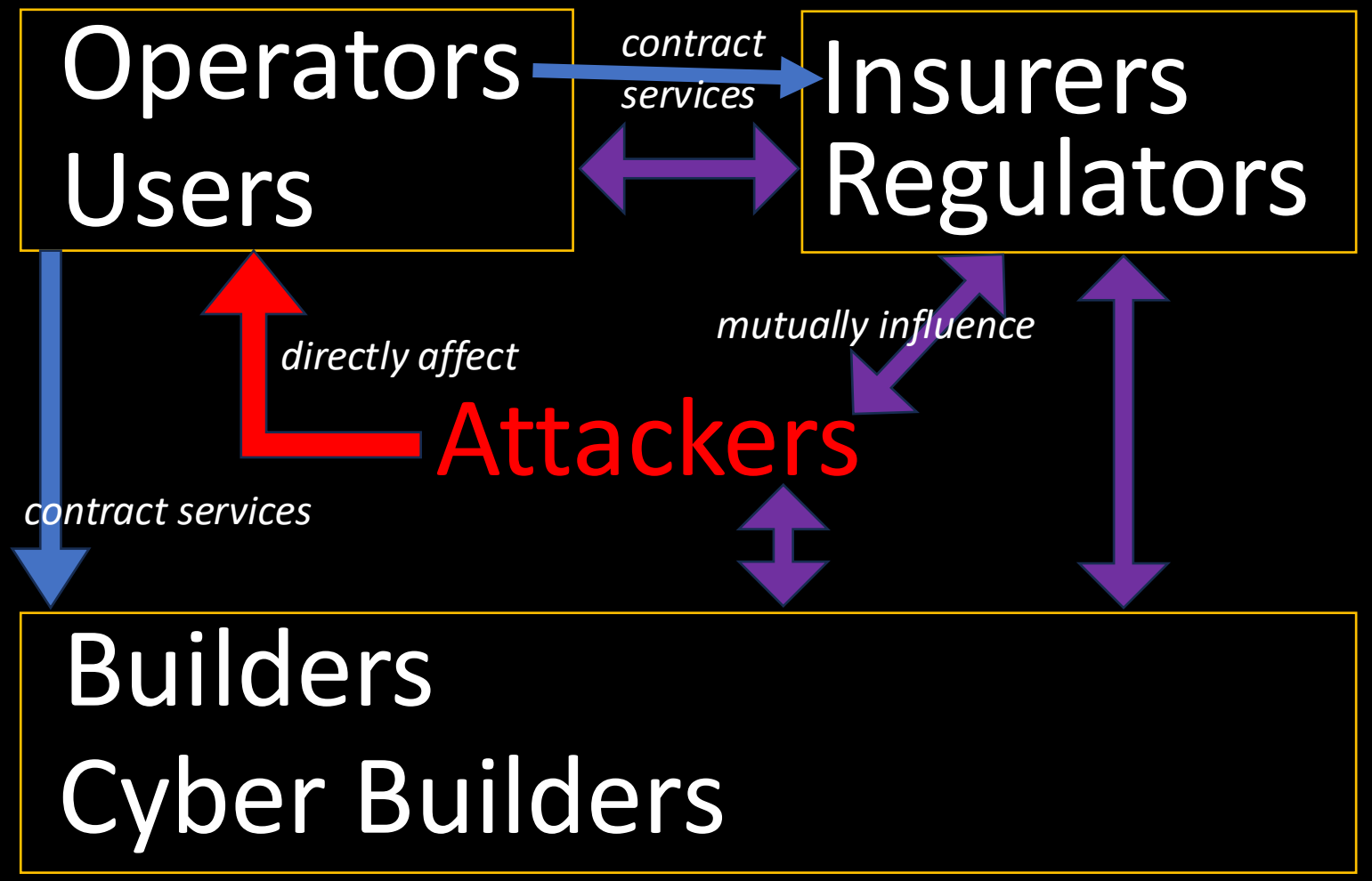
To explain why, it helps to step back and consider the threat landscape. Broadly speaking, attackers have one of two objectives. One class of attacker focuses on achieving maximal *depth*: they spend considerable effort to exploit one particular high-value target (e.g., a bank). The other class of attacker focuses on achieving maximal *breadth*: they develop an attack that is damaging because it can impact millions of targets, even if each target is low-value.

This distinction is apparent in nearly all domains of security. It is what differentiates standard phishing attacks [19]—which send generic letters from Nigerian princes—from spear phishing attacks [9]—which are explicitly designed for and executed against high-profile targets. It is also what differentiates attacks like credential stuffing [46]—where attackers re-use previously-leaked username/password combinations to try and authenticate as *someone*—from attackers who aim to breach a specific targeted account (e.g., through brute-force attacks, or exploits on the password reset chain like SIM swapping). And it is what differentiates "script kiddies" who re-use exploits in known-vulnerable software, from APTs that develop novel zero-day exploits.

Today, fortunately, it is almost never possible to achieve both breadth and depth at the same time. An attacker can either go deep, or go wide, but not both. For this reason, the average person does not need to worry about being the victim of a targeted attack from a well-resourced adversary, as these types of attacks are necessarily infrequent due to the high level of human effort they require. But we expect that LLMs could change this. Through a series of case studies, we analyze ways in which LLMs could allow attacks to go both broad *and* deep. Specifically, we consider two potential directions where LLMs could have high impact.

Direction 1: Exploiting the long-tail of systems. Exploits are most valuable when they target systems with a large number of users (e.g., an operating system like iOS or Windows), as this maximizes the number of potential victims. As a result, these systems are also the most protected and hardest to attack. And yet, attackers still primarily target such systems over the long-tail of systems with a small number of users (e.g., an IoT device or software application with only hundreds of downloads). While the long-tail of systems is undoubtedly less-well protected and thus cheaper to attack, the expected value of exploiting such systems is too low.

Put differently, the baseline cost for a human to find a vulnerability and turn it into a workable exploit is only viable if there are



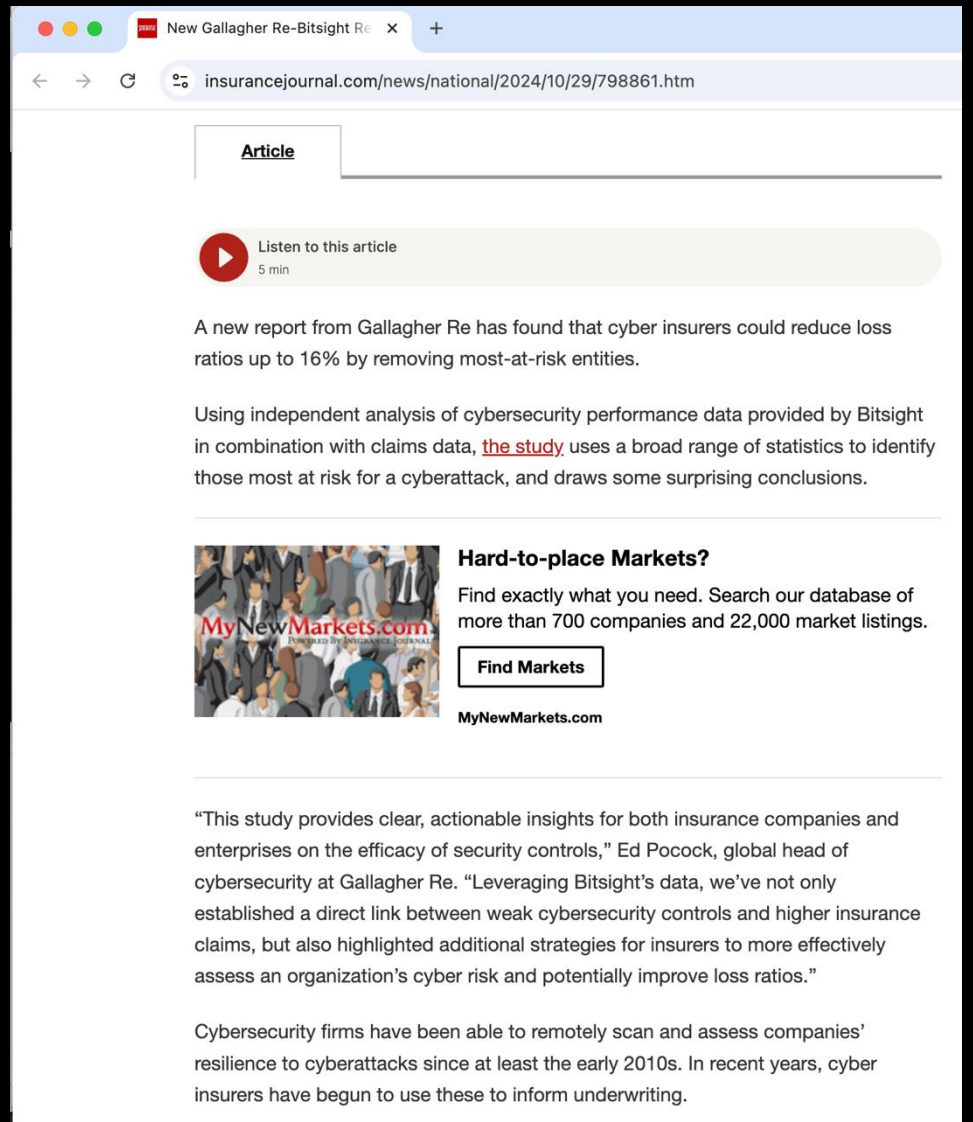
Notes

- Relationships induce incentives
- (Cyber)builders are users and operators too!

Regulators, Insurers

- Governance organizations **define and enforce the rules of the game**
  - Government regulators (FTC, SEC, CISA, EU regulatory bodies …)
  - Standards organizations (NIST, ISO, OWASP, …)
  - Laws: HIPAA, FERPA, PCI, …
- Insurance and financial intermediaries help **manage cyber risk** (effectively setting rules of their own)
  - Cyber insurance providers (Chubb, etc.)
  - Credit rating agencies
  - Investment firms (which may take into account cybersecurity posture)

Cyber insurance: Elevating evidence?



New Gallagher Re-Bitsight Re

insurancejournal.com/news/national/2024/10/29/798861.htm

Article

Listen to this article
5 min

A new report from Gallagher Re has found that cyber insurers could reduce loss ratios up to 16% by removing most-at-risk entities.

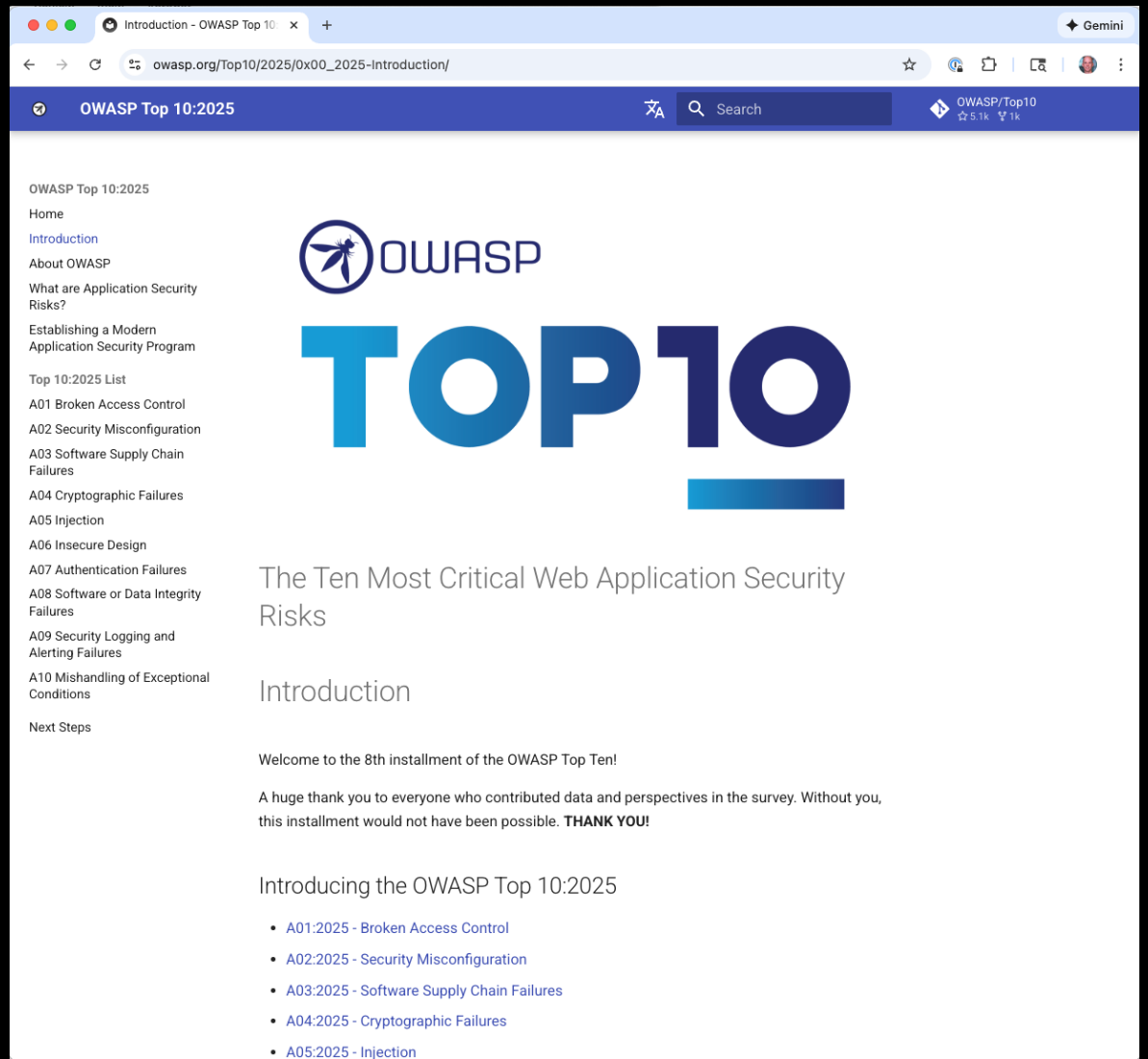
Using independent analysis of cybersecurity performance data provided by Bitsight in combination with claims data, the study uses a broad range of statistics to identify those most at risk for a cyberattack, and draws some surprising conclusions.

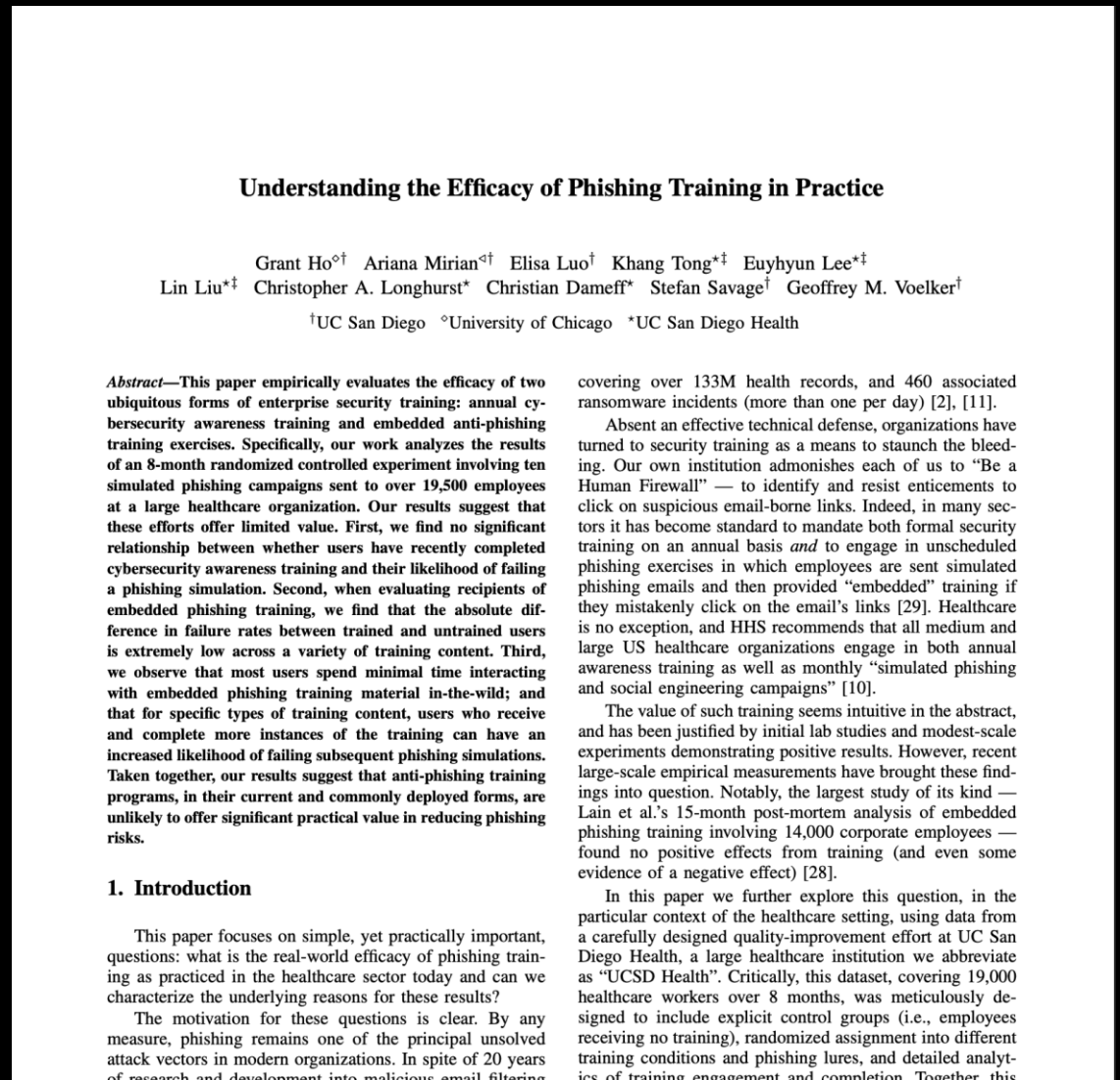
"This study provides clear, actionable insights for both insurance companies and enterprises on the efficacy of security controls," Ed Pocock, global head of cybersecurity at Gallagher Re. "Leveraging Bitsight's data, we've not only established a direct link between weak cybersecurity controls and higher insurance claims, but also highlighted additional strategies for insurers to more effectively assess an organization's cyber risk and potentially improve loss ratios."

Cybersecurity firms have been able to remotely scan and assess companies' resilience to cyberattacks since at least the early 2010s. In recent years, cyber insurers have begun to use these to inform underwriting.

Readings for next week

Plus: “How to Read a Paper?”





Understanding the Efficacy of Phishing Training in Practice

Grant Ho[◇†] Ariana Mirian[◁†] Elisa Luo[†] Khang Tong[*‡] Euyhyun Lee[*‡]
Lin Liu[*‡] Christopher A. Longhurst[*] Christian Dameff[*] Stefan Savage[†] Geoffrey M. Voelker[†]

[†]UC San Diego [◇]University of Chicago [*]UC San Diego Health

***Abstract*—This paper empirically evaluates the efficacy of two ubiquitous forms of enterprise security training: annual cybersecurity awareness training and embedded anti-phishing training exercises. Specifically, our work analyzes the results of an 8-month randomized controlled experiment involving ten simulated phishing campaigns sent to over 19,500 employees at a large healthcare organization. Our results suggest that these efforts offer limited value. First, we find no significant relationship between whether users have recently completed cybersecurity awareness training and their likelihood of failing a phishing simulation. Second, when evaluating recipients of embedded phishing training, we find that the absolute difference in failure rates between trained and untrained users is extremely low across a variety of training content. Third, we observe that most users spend minimal time interacting with embedded phishing training material in-the-wild; and that for specific types of training content, users who receive and complete more instances of the training can have an increased likelihood of failing subsequent phishing simulations. Taken together, our results suggest that anti-phishing training programs, in their current and commonly deployed forms, are unlikely to offer significant practical value in reducing phishing risks.**

1. Introduction

This paper focuses on simple, yet practically important, questions: what is the real-world efficacy of phishing training as practiced in the healthcare sector today and can we characterize the underlying reasons for these results?

The motivation for these questions is clear. By any measure, phishing remains one of the principal unsolved attack vectors in modern organizations. In spite of 20 years of research and development into malicious email filtering, covering over 133M health records, and 460 associated ransomware incidents (more than one per day) [2], [11].

Absent an effective technical defense, organizations have turned to security training as a means to staunch the bleeding. Our own institution admonishes each of us to “Be a Human Firewall” — to identify and resist enticements to click on suspicious email-borne links. Indeed, in many sectors it has become standard to mandate both formal security training on an annual basis *and* to engage in unscheduled phishing exercises in which employees are sent simulated phishing emails and then provided “embedded” training if they mistakenly click on the email’s links [29]. Healthcare is no exception, and HHS recommends that all medium and large US healthcare organizations engage in both annual awareness training as well as monthly “simulated phishing and social engineering campaigns” [10].

The value of such training seems intuitive in the abstract, and has been justified by initial lab studies and modest-scale experiments demonstrating positive results. However, recent large-scale empirical measurements have brought these findings into question. Notably, the largest study of its kind — Lain et al.’s 15-month post-mortem analysis of embedded phishing training involving 14,000 corporate employees — found no positive effects from training (and even some evidence of a negative effect) [28].

In this paper we further explore this question, in the particular context of the healthcare setting, using data from a carefully designed quality-improvement effort at UC San Diego Health, a large healthcare institution we abbreviate as “UCSD Health”. Critically, this dataset, covering 19,000 healthcare workers over 8 months, was meticulously designed to include explicit control groups (i.e., employees receiving no training), randomized assignment into different training conditions and phishing lures, and detailed analyt-