



Yet Another Introduction to Usable Security

aka: A never ending story of finding the enemy

COINS Summer School 2021

Prof. Dr.-Ing.

Luigi Lo Iacono

Data and Application Security Group

Hochschule Bonn-Rhein-Sieg




Security objectives

Threats

Liability

Accountability

tracked by  **CMS**
law-tax-future

GDPR Enforcement Tracker

The CMS.Law GDPR Enforcement Tracker is an overview of fines and penalties which data protection authorities within the EU have imposed under the EU General Data Protection Regulation (GDPR, DSGVO). Our aim is to keep this list as up-to-date as possible. Since not all fines are made public, this list can of course never be complete, which is why we appreciate any [indication of further GDPR fines and penalties](#). Please note that we do not list any fines imposed under national / non-European laws, under non-data protection laws (e.g. competition laws / electronic communication laws) and under "old" pre-GDPR-laws.

New features: "ETid" and "Direct URL"
 We have assigned a unique and permanent ID to each fine in our database, which makes it possible to precisely address fines, e.g. in publications. Once an "ETid" has been assigned to a fine, it remains the same, even if the fine is overturned or amended by courts at a later date, or if we add fines that were issued chronologically before. The "Direct URL" (click "+" on a specific ETid to view details of a fine) can be used to share fines online, e.g. on Twitter or other media.

Show entries Search:

ETid	Country	Date of Decision	Fine [€]	Controller/Processor	Quoted Art.	Type	Source
ETid-694	SPAIN	2021-05-25	100,000	Vodafone España, SAU	Art. 28 GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-689	IRELAND	2021-03-23	90,000	Irish Credit Bureau DAC	Art. 5 (2) GDPR, Art. 24 (1) GDPR, Art. 25 (1) GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-687	THE NETHERLANDS	2020-03-24	15,000	CP&A	Art. 9 GDPR, Art. 32 GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-686	ITALY	2021-04-15	40,000	Comune di Palermo	Art. 5 (1) f) GDPR, Art. 25 GDPR, Art. 32 GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-681	POLAND	2021-04-22	245,000	Cyrowy Polsat S.A.	Art. 24 (1) GDPR, Art. 32 (1), (2) GDPR, Art. 34 (1) GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-669	ICELAND	2021-04-29	23,100	InfoMentor ehf	Art. 5 (1) f) GDPR, Art. 32 (1) b), d) GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-668	ROMANIA	2021-05-07	2000	World Class România S.A.	Art. 32 GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-666	HUNGARY	2021-03-24	27,700	Budapest Főváros Kormányhivatala XI. kerületi Hivatalát (11th District Public Health Department of the Government Office of the Capital City Budapest)	Art. 32 (1) a), b) GDPR, Art. 32 (2) GDPR, Art. 33 (1) GDPR, Art. 34 (1) GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-664	BELGIUM	2021-04-26	100,000	Financial company	Art. 5 (1) f) GDPR, Art. 32 GDPR	Insufficient technical and organisational measures to ensure information security	link
ETid-639	CZECH REPUBLIC	2020	387	Private healthcare provider	Art. 24 GDPR, Art. 32 (1) GDPR	Insufficient technical and organisational measures to ensure information security	link

Showing 1 to 10 of 151 entries (filtered from 701 total entries) Previous 2 3 4 5 ... 16 Next

<https://www.enforcementtracker.com/>

Threats

Integrity

Availability



Bloomberg

Subscribe

Cybersecurity

Gas Stations Run Dry as Pipeline Races to Recover From Hacking

By [Joe Carroll](#), [Andres Guerra Luz](#), and [Jill R Shah](#)

May 9, 2021, 3:43 AM GMT+2

Updated on May 11, 2021, 9:59 AM GMT+2

- ▶ Biden says Russia has “some responsibility” to address attack
- ▶ Gas stations from Alabama to South Carolina run out of fuel

Source: <https://www.bloomberg.com/news/articles/2021-05-09/u-s-fuel-sellers-scramble-for-alternatives-to-hacked-pipeline>

Threats

Access control

THE SUN, A NEWS UK COMPANY

THE Sun

< MONEY | DEAR DEIDRE | TECH | TRAVEL | MOTORS | PUZZLES | SUN

< Money News | Shopping | Money Tips | Property | Business

Money > News Money

NOT APPY Klarna app 'bug' let users log in to other shoppers' accounts

Lynsey Barber

18:54, 27 May 2021 | Updated: 17:00, 28 May 2021

Source: <https://www.thesun.co.uk/money/15085000/klarna-app-down-reports-logging-other-shoppers-accounts/>

Threats

Authenticity



The screenshot shows the EUIPO website with a news article. The header includes the EUIPO logo and navigation links for Home, Trade marks, Designs, and Law & practice. The article title is "Payment Service Directive 2, Increased Security for Online Payments" and is dated December 08, 2020. Below the title is an image of a hand holding a blue credit card. The article text at the bottom states: "New security requirements for online payments will come into effect in the European Economic Area in January 2021".

Source: <https://euiipo.europa.eu/ohimportal/en/news/-/action/view/8410243>

What are the reasons for security incidents?

BUSINESS TECHNOLOGY

HUMAN ERROR IS STILL THE NUMBER ONE CAUSE OF MOST DATA BREACHES IN 2021

Companies are not getting message across to staff.

11. 95% of cybersecurity breaches are **due to human error**

Cyber-criminals and hackers will infiltrate your company through your weakest link, [which is almost never in the IT department.](#)

7 Data Breaches Caused by **Human Error**: Did Enc Play a Role?

Why **Human Error** is #1 Cyber Security Threat to Businesses in 2021

 February 04, 2021  The Hacker News

[Home](#) > [Privacy & Security](#)

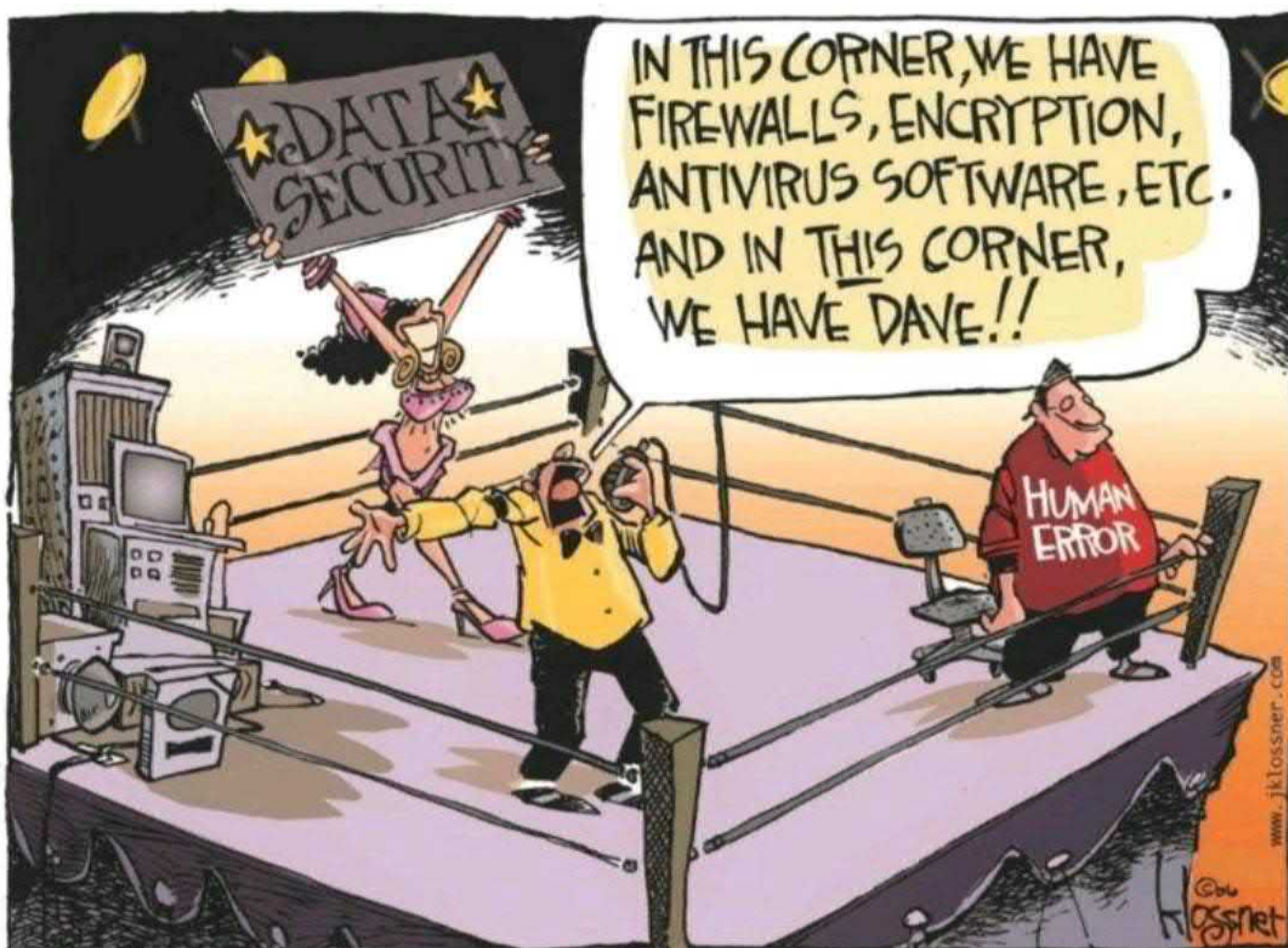
[Privacy & Security](#)

The bulk of data breaches have a **human element**, says Verizon report

By [Howard Solomon](#) - May 14, 2021

 503  0

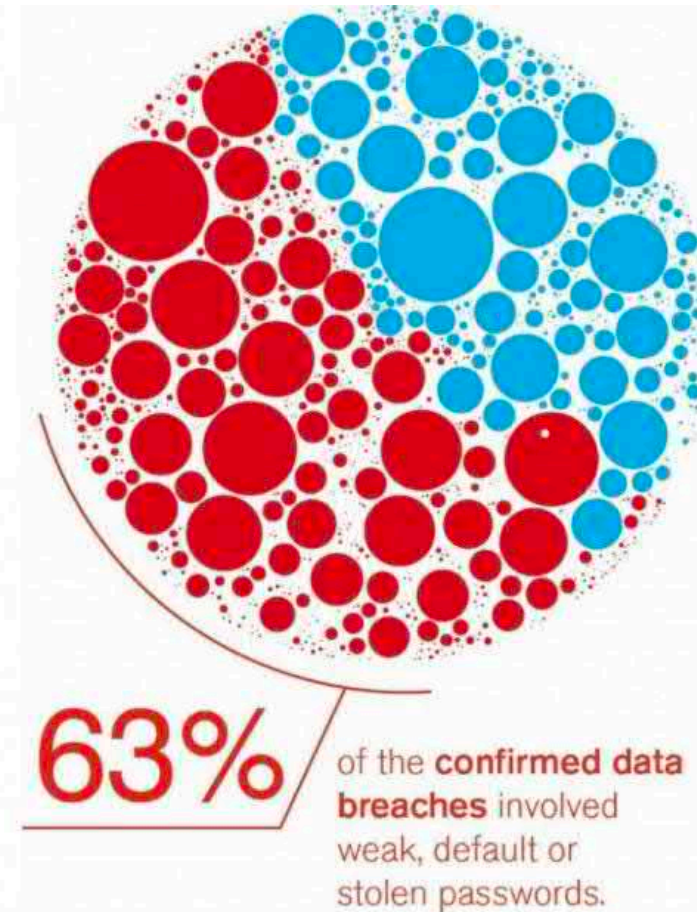
What-Who are the reasons for security incidents?



- “Dave” is also known as:
- Johnny
 - Jane
 - “the user”



Users are the weakest link in security

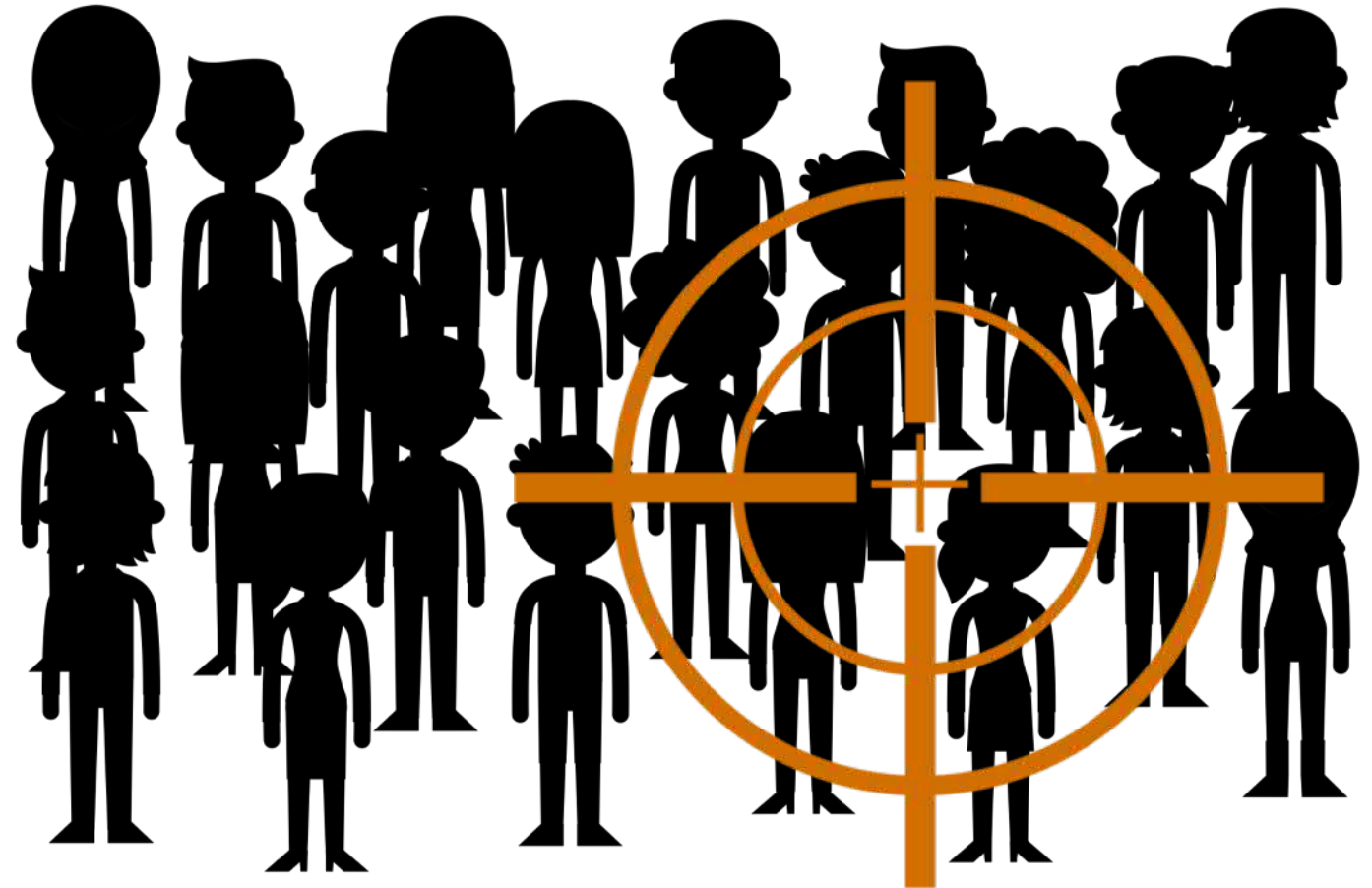


Source: Verizon Data Breach Investigations Report 2016

<https://www.verizon.com/about/news/verizons-2016-data-breach-investigations-report-finds-cybercriminals-are-exploiting-human>

Are Users The Enemy?

~~Users Are The Weakest Link~~



Ciaran Martin

” So, let's get serious about understanding the human being in all this. Let's stop talking nonsense about humans being the weakest link in cyber security: it's a bit like saying the weakest link in a sports team is all the players.”



Ciaran Martin,
NCSC Chief Executive Officer
Foto: NCSC

Source: NCSC, Ciaran Martin's speech to CBI, 13 September 2017
<https://www.ncsc.gov.uk/speech/ciaran-martins-speech-cbi>

Users Are The Weakest Link

1999

USERS ARE NOT THE ENEMY

Why users compromise computer security mechanisms and how to take remedial measures

Confidentiality is an important aspect of computer security. It depends on authentication mechanisms, such as passwords, to safeguard access to information [9]. Traditionally, authentication procedures are divided into two stages: *identification* (User ID), to identify the user; and *authentication*, to verify that the user is the legitimate owner of the ID. It is the latter stage that requires a secret password. To date, research on password security has focused on designing technical mechanisms to protect

access to systems; the usability of these mechanisms has rarely been investigated. Hitchings [8] and Davis and Price [4] argue that this narrow perspective has produced security mechanisms that are, in practice, less effective than they are generally assumed to be. Since security mechanisms are designed, implemented, applied and breached by people, human factors should be considered in their design. It seems that currently, hackers pay more attention to the human link in the security chain than security designers do, for example, by using social engineering techniques to obtain passwords.

The key element in password security is the crackability of a password combination. Davies and Ganesan [3] argue that an adversary's ability

do not have to write them down). The U.S. Federal Information Processing Standards [5] suggest several criteria for assuring different levels of password security. *Password composition*, for example, relates the size of a character set from which a password has been chosen to its level of security.

ANNE ADAMS AND
MARTINA ANGELA SASSE

An alphanumeric password is therefore more secure than one composed of letters alone. Short *password lifetime*—changing passwords frequently—is suggested as reducing the risk associated with undetected compromised passwords. Finally, *password ownership*, in particular individual ownership, is recommended to:

- Increase individual accountability;
- Reduce illicit usage;



Anne Adams



M. Angela Sasse



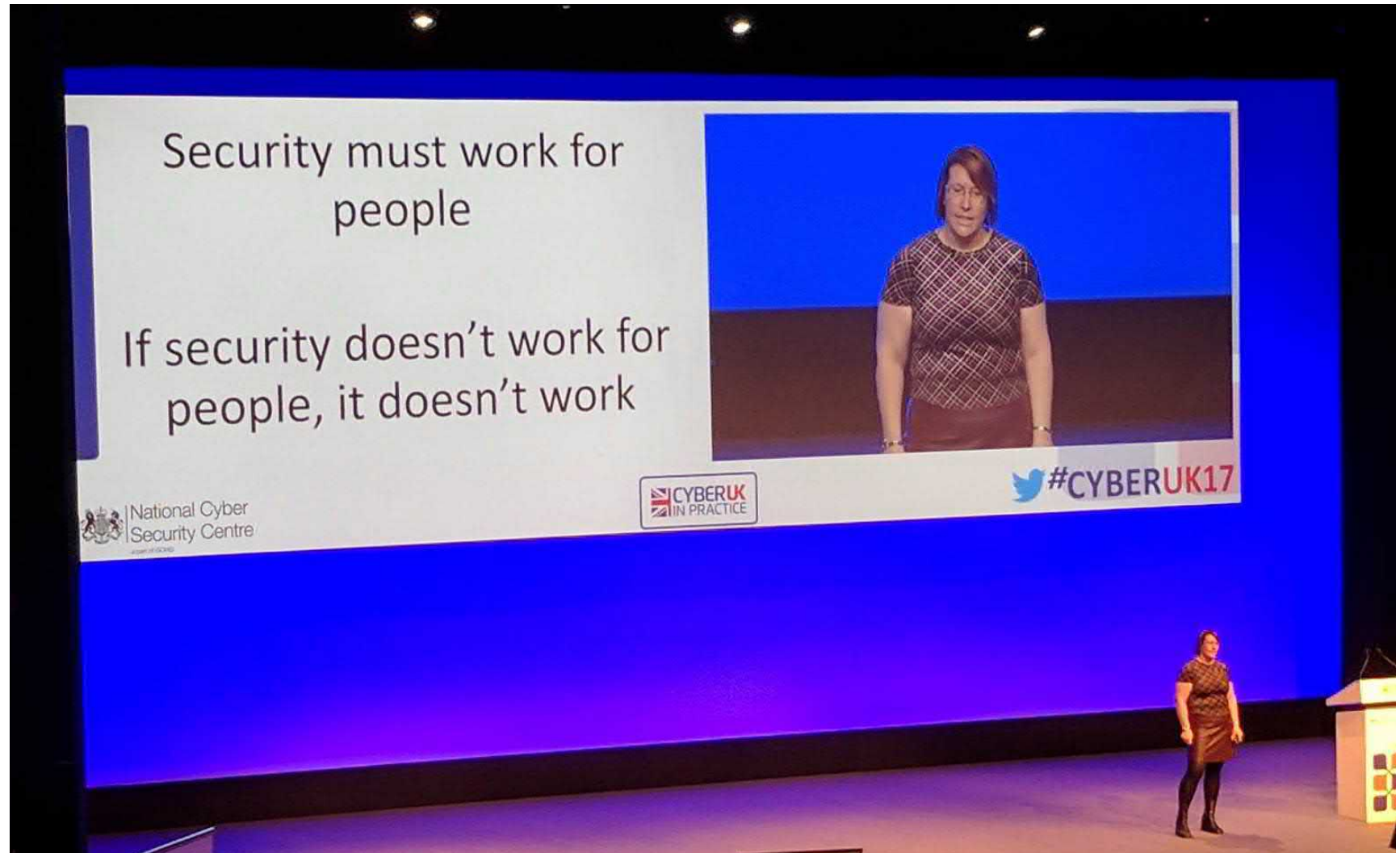
Designed by Freepik

Source: Communications of the ACM,
Volume 42 Issue 12, Dec. 1999

Prof. Dr.-Ing. Luigi Lo Iacono
Internet | COINS Summer School 2021

Source:
<https://iet.open.ac.uk/people/anne.adams>
<https://www.ei.ruhr-uni-bochum.de/fakultaet/personen/sasse/>

Emma W.



Source: Twitter, Harry Metcalfe @harrym, <https://twitter.com/harrym/status/841970258848157696>



Emma W.,
Commissioning Editor for Advice and Guidance
„People: The Strongest Link“, CyberUK In Practice 2017



Source: Superman - Secret Origin Vol 1 #3, January, 2010,
http://dc.wikia.com/wiki/File:Clark_Kent_001.jpg

People: The **S**trongest Link

~~Users Are Not The Enemy~~

~~Users Are The Weakest Link~~



Source: Superman - Secret Origin Vol 1 #3, January, 2010,
http://dc.wikia.com/wiki/File:Clark_Kent_001.jpg

Empower People to Become The a Strongest Link

~~Users Are Not The Enemy~~

~~Users Are The Weakest Link~~



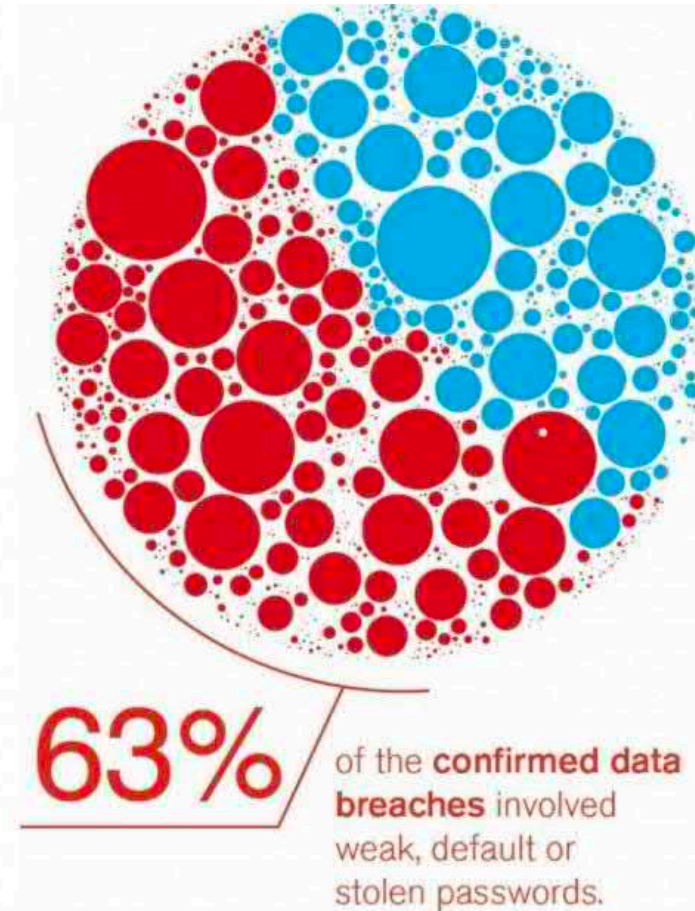
Usability Evaluation of Security Mechanisms



Usability Evaluation of Security Mechanisms



Users are the weakest link in security

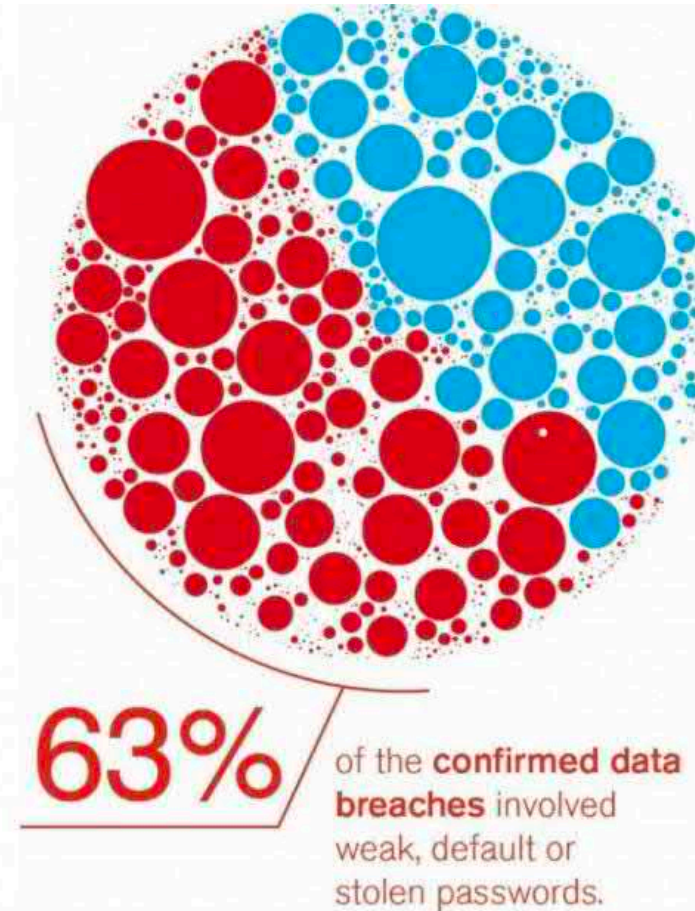


Source: Verizon Data Breach Investigations Report 2016

<https://www.verizon.com/about/news/verizons-2016-data-breach-investigations-report-finds-cybercriminals-are-exploiting-human>

verizon^v

~~Users are the
weakest link in
security~~

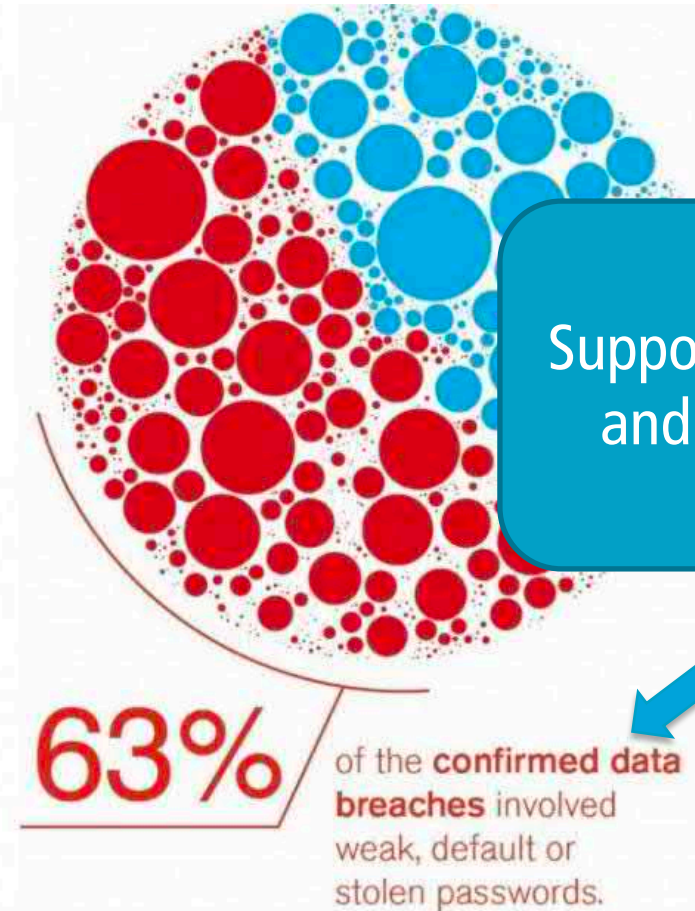


Source: Verizon Data Breach Investigations Report 2016

<https://www.verizon.com/about/news/verizons-2016-data-breach-investigations-report-finds-cybercriminals-are-exploiting-human>



~~Users are the
weakest link in
security~~



Support Johnny, Jane,
and all the users.

Source: Verizon Data Breach Investigations Report 2016
<https://www.verizon.com/about/news/verizons-2016-data-breach-investigations-report-finds-cybercriminals-are-exploiting-human>

Technological factors

Human factors



Usable Security



Major Themes (selected)

- User Authentication
- Email Security and PKI
- Phishing (Warnings)
- Storage
- Device Pairing
- Policy
- Mobile Security und Privacy
- Administrators
- Developers

Major Themes (selected)

- **User Authentication** (tomorrow)
- **Email Security and PKI**
- Phishing (Warnings)
- Storage
- Device Pairing
- Policy
- Mobile Security und Privacy
- Administrators
- **Developers**

Usable Email Security

– the example par excellence –



1971

First E-Mail



1981

RFC788 SMTP



1991

US government
wants to
establish
surveillance
program



1991

PGP is released



1995

S/MIME is releases

Usable Email Security

– the example par excellence –


Today

~ 14% of
German
email users
encrypt
messages.



Source: The internet. Adapted by Jan Tolsdorf

Usable Email Security

– the example par excellence –

CHALLENGE ACCEPTED



Why do users not encrypt their emails?



The encryption tools have poor usability:

- UIs
- Key Management

1999

**Why Johnny Can't Encrypt:
A Usability Evaluation of PGP 5.0**

Alma Whitten
*School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alma@cs.cmu.edu*

J. D. Tygar¹
*EECS and SIMS
University of California
Berkeley, CA 94720
tygar@cs.berkeley.edu*

Abstract

User errors cause or contribute to most computer security failures, yet user interfaces for security still tend to be clumsy, confusing, or near-nonexistent. Is this simply due to a failure to apply standard user interface design techniques to security? We argue that, on the contrary, effective security requires a different usability standard, and that it will not be achieved through the user interface design techniques appropriate to other types of consumer software.

To test this hypothesis, we performed a case study of a security program which does have a good user interface by general standards: PGP 5.0. Our case study used a cognitive walkthrough analysis together with a laboratory user test to evaluate whether PGP 5.0 can be successfully used by cryptography novices to achieve effective electronic mail security. The analysis found a number of user interface design flaws that may contribute to security failures, and the user test demonstrated that when our test participants were given 90 minutes in which to sign and encrypt a message using PGP 5.0, the majority of them were unable to do so successfully.

We conclude that PGP 5.0 is not usable enough to provide effective security for most computer users, despite its attractive graphical user interface, supporting our hypothesis that user interface design for effective security remains an open problem. We close with a brief description of our continuing work on the development and application of user interface design principles and techniques for security.

1 Introduction

Security mechanisms are only effective when used correctly. Strong cryptography, provably correct protocols, and bug-free code will not provide security if the people who use the software forget to click on the encrypt button when they need privacy, give up on a communication protocol because they are too confused about which cryptographic keys they need to use, or accidentally configure their access control mechanisms to make their private data world-readable. Problems such as these are already quite serious: at least one researcher [2] has claimed that configuration errors are the probable cause of more than 90% of all computer security failures. Since average citizens are now increasingly encouraged to make use of networked computers for private transactions, the need to make security manageable for even untrained users has become critical [4, 9].

This is inescapably a user interface design problem. Legal remedies, increased automation, and user training provide only limited solutions. Individual users may not have the resources to pursue an attacker legally, and may not even realize that an attack took place. Automation may work for securing a communications channel, but not for setting access control policy when a user wants to share some files and not others. Employees can be required to attend training sessions, but home computer users cannot.

Why, then, is there such a lack of good user interface design for security? Are existing general user interface design principles adequate for security? To answer these questions, we must first understand what kind of usability security requires in order to be

Usable Email Security

Error prevention

Users failed to turn on security, even if one asks them to turn it on:

- 10% (Ruoti et al., 2013)
- 17% (Robison et al., 2012)
- 25% (Whitten / Tygar, 1999)
- 100% (Sheng et al., 2006)

Usable Email Security

Flexibility and efficiency of use

- Users prefer integrated tools.
→ Security secondary goal

(Atwater et al., 2015; Ruoti et al., 2016; Ruoti et al., 2013)

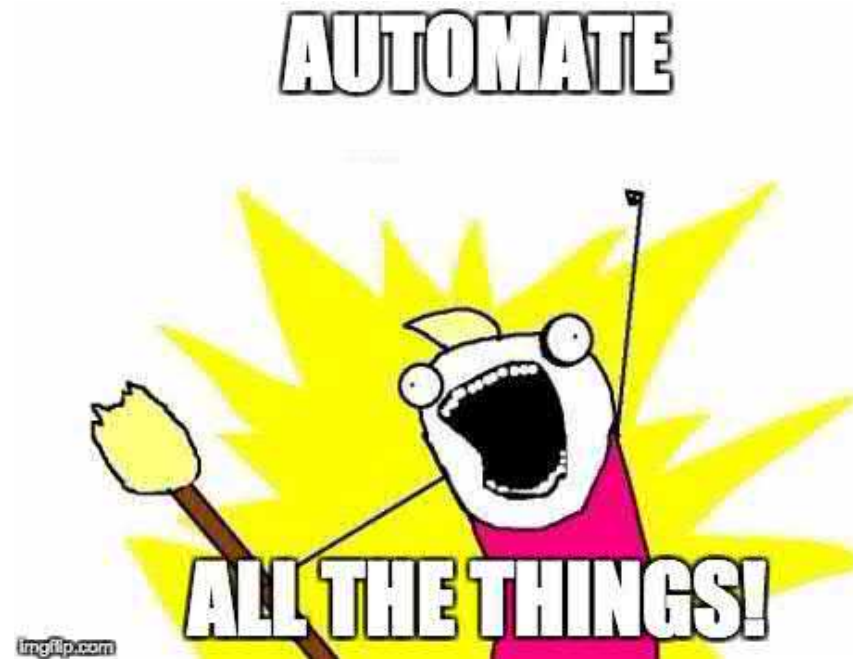
Usable Email Security

Match between system and real world

- Language and symbols used in email security do not support users in managing PKI.
 - Users prefer solutions with hidden cryptographic details. (Ruoti et al. 2018)
 - Users trust their providers with key management. (Bai et al. 2016)



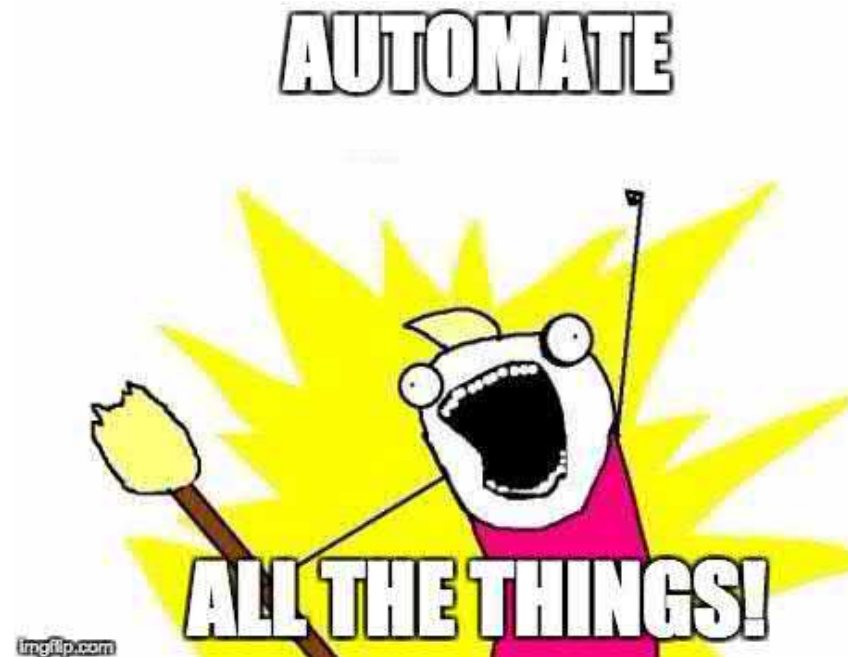
Usable Email Security





Usable Email Security

NO!



Usable Email Security

- **Opaque security:**
 - Causes errors;
 - Lowers trust.

(Atwater et al., 2015; Ruoti et al., 2016; Ruoti et al., 2013, Sheng et al., 2006)
- **Always on encryption:**
 - Does not fit users' expectations;
 - Is associated with paranoia;
 - Puts burden on receiver.

(Gaw et al., 2006; Hecht et al., 2015; Ruoti et al., 2013; Whitten, 2004)

Usable Email Security

- Requirements for Usable Email Tools
 - Tight integration;
 - Tutorials;
 - Streamline Onboarding;
 - Understandable and trustworthy design;
 - Easy-to-use key management.

Cf.: S. Ruoti and K. Seamons, "Johnny's Journey Toward Usable Secure Email," *IEEE Security Privacy*, vol. 17, no. 6, pp. 72–76, Nov. 2019

Usable Email Security

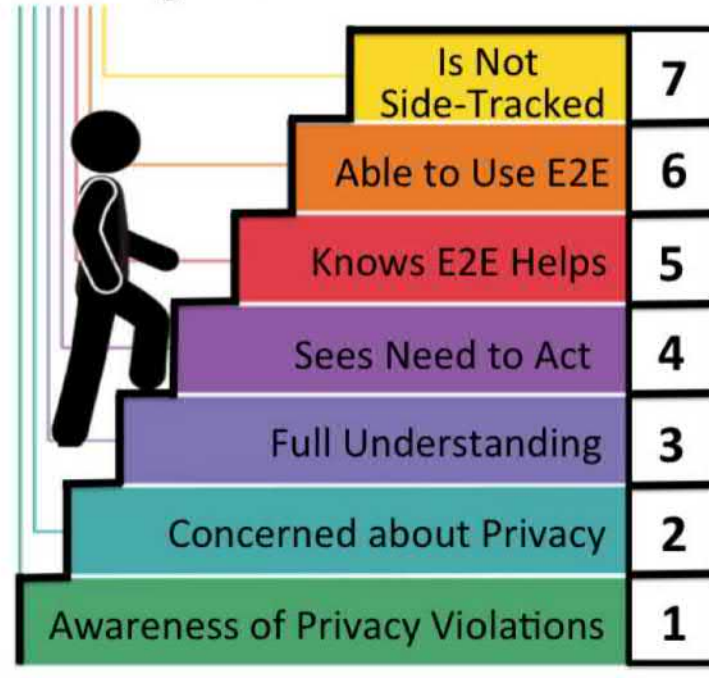
- Requirements for Usable Email Tools
 - Tight integration;
 - Tutorials;
 - Streamline Onboarding;
 - Understandable and trustworthy design;
 - Easy-to-use key management.

But users still do not use email encryption.

Cf.: S. Ruoti and K. Seamons, "Johnny's Journey Toward Usable Secure Email," *IEEE Security Privacy*, vol. 17, no. 6, pp. 72–76, Nov. 2019

Usable Email Security

Steps to using E2E



Impact of tools' usability

Other reasons

Figure: K. Renaud, M. Volkamer, and A. Renkema-Padmos, "Why Doesn't Jane Protect Her Privacy?," in *14th International Symposium on Privacy Enhancing Technologies (PETs)*, Cham, 2014, pp. 244–262.

Usable Messaging Security

- Most secure and usable messaging today:
 - Security by default and by design
 - Users still
 - Do not trust encryption;
 - Lack awareness;
 - Have misconceptions;
 - Do not feel targeted.



Source: <https://www.whatsapp.com/>



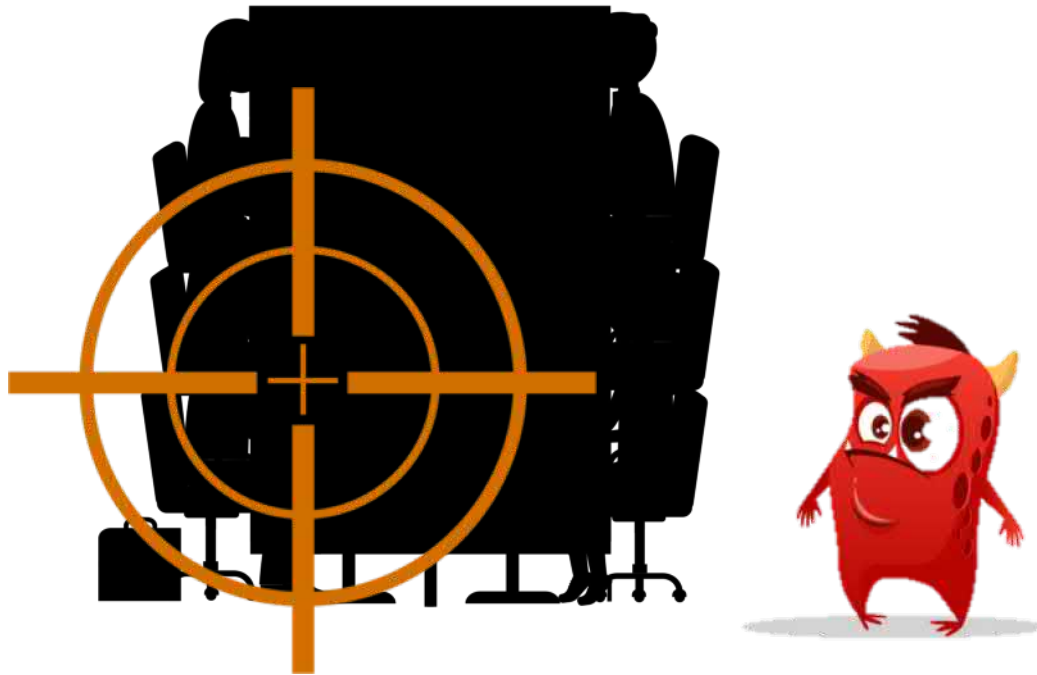
Source: <https://github.com/signalapp>

Cf.: S. Dechand, A. Naiakshina, A. Danilova, and M. Smith, "In Encryption We Don't Trust: The Effect of End-to-End Encryption to the Masses on User Perception," in *2019 IEEE European Symposium on Security and Privacy (EuroS P)*, Jun. 2019, pp. 401–415.



Users are not the enemy.

WHO IS THE ENEMY THEN?



The Developer is the Enemy

Glenn Wurster
Carleton Computer Security Lab
School of Computer Science
Carleton University, Canada
gwurster@scs.carleton.ca

P. C. van Oorschot
Carleton Computer Security Lab
School of Computer Science
Carleton University, Canada
paulv@scs.carleton.ca

ABSTRACT

We argue that application developers, while often viewed as allies in the effort to create software with fewer security vulnerabilities, are not reliable allies. They have varying skill sets which often do not include security. Moreover, we argue that it is inefficient and unrealistic to expect to be able to successfully teach all of the world's population of software developers to be security experts. We suggest more efficient and effective alternatives, focusing on those developers who produce core functionality used by other developers (e.g. those who develop popular APIs – *Application Programming Interfaces*). We discuss the benefits of designing APIs which can be easily used in a secure fashion to encourage security. We also introduce two straw-man proposals which integrate security into the work-flow of an application developer. Data tagging and unsuppressible warnings provide the basis for further work where the most natural use (path of least resistance) results in secure code. We believe there are benefits to co-opting developers into programming securely.

Categories and Subject Descriptors

D.4.6 [Software]: Security and Protection; D.2.3 [Software]: Coding Tools and Techniques; D.2.6 [Software]: Programming Environments

General Terms

Human Factors, Security

Keywords

software developers, human factors, software security, usability, education, development tools, persuasion

1. INTRODUCTION AND OVERVIEW

According to Adams et al. [1], many security policies are enforced on a need-to-know basis. This need-to-know mentality seems historically to have been based on the idea that

increased knowledge of security mechanisms and threats increases the potential for information leaks. The authors argue that this need-to-know mentality results in a situation where users are less motivated to work securely. In a related position, Vidyaraman et al. [46] argued that it can be beneficial to security to consider users as the enemy, in that their actions directly influence system security and they often perform tasks that actively reduce security.

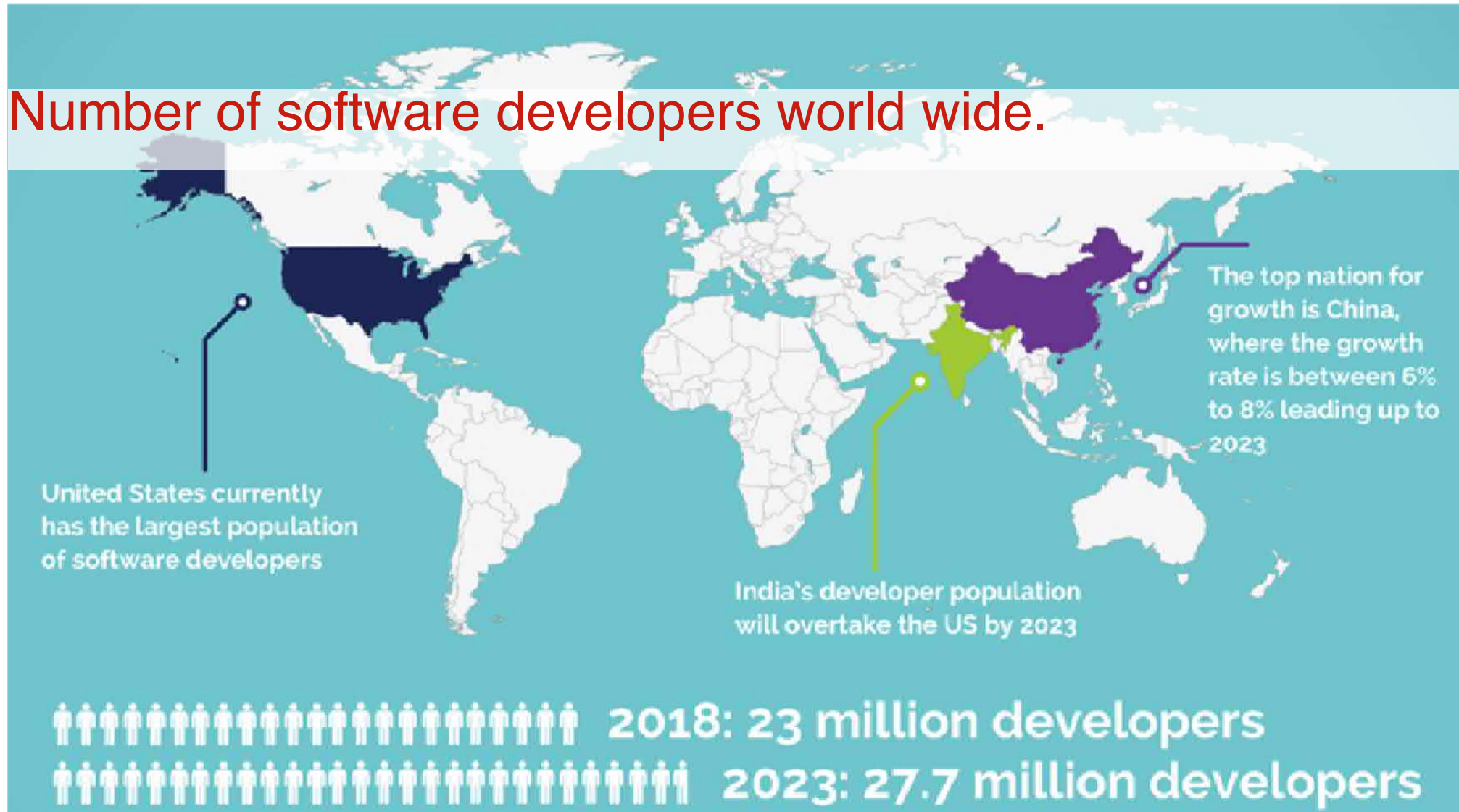
Application developers are currently treated different than users. Often, API (*Application Programming Interface*) developers provide functionality and application developers use this functionality to create applications. The security community relies on application developers to be knowledgeable and to understand how to use each API securely. In effect, we rely on all application developers to be security experts. In recent years, it has been widely acknowledged that software developers do not by any means have sufficient security expertise to make this model work [47, 6]. Consequently, some major players in industry have mounted substantial efforts towards increasing the security knowledge of general developers [26]. We argue that not only is relying on all developers individually to code securely doomed to failure, but that extensive training (even if it were possible) to give *all* software developers detailed security expertise is not the right approach. We suggest additional focus on providing development environments where even application developers without security expertise are less likely to make security errors; as developer skill sets become increasingly customized, requiring all developers to have security expertise as a core competency is too heavy a tax to pay.

It has often been said that *complexity is the enemy of security*. This complexity is present at the user interface level of software, in programming libraries and tools available to the developer of an application, as well as in system code.

The modern developer no longer builds applications from scratch. Instead, most developers essentially glue different libraries together to perform a task. Different developers are responsible for different parts of the resulting application. Given this situation, it is unreasonable to assume or require that all developers will be properly educated and proficient in security (e.g. creating

The Developer is the Enemy
People: The Strongest Link
~~Users Are Not The Enemy~~
~~Users Are The Weakest Link~~

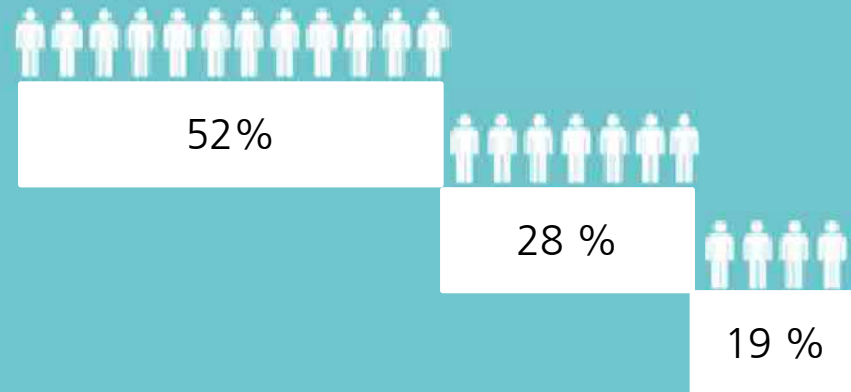
Number of software developers world wide.



Source: IDC's Worldwide Developer Census, 2018

Number of software developers world wide.

 2018: 23 million developers



11.65M **full-time** developers,
6.35M **part-time** developers
4.30M **nonprofessional** developers

Usable Security for Software Developers



Framework
Designers &
Security Experts



Software Developers

Key Role: Software Developers



Source: Holz T, Pohlmann N, Bodden E, Smitz M, Hoffmann J (2016)
Human-Centered Systems Security - IT-Sicherheit von Menschen für Menschen.

Software Developers – Human After All

” Imagine, hypothetically, just for a moment, that programmers are humans. No, don't laugh, I'm serious. [...] Now suppose for a moment, also for the sake of the argument, that their chief method of communicating and interacting with computers was with programming languages. What would we, as HCI people then do? Run screaming in the other direction, I hear you think.

Steven Pemberton (1997). „Views and Feelings: Programmers are Humans Too, 2“. In: SIGCHI Bulletin 29.3. url: <http://bulletin.sigchi.org/1997/july/news/views>.
Image: Vera de Kok,
https://en.wikipedia.org/wiki/Steven_Pemberton#/media/File:Steven_Pemberton_2017.jpg



2016

THE SECURITY-USABILITY TRADEOFF MYTH

Developers Are Not the Enemy!

The Need for Usable Security APIs

Matthew Green | Johns Hopkins University
Matthew Smith | University of Bonn and Fraunhofer FKIE

Source: IEEE Security & Privacy
Volume: 14, Issue: 5, Sept.-Oct. 2016

Modern security practice has created an adversarial relationship between security software designers and developers. But developers aren't the enemy. To strengthen security systems across the board, security professionals must focus on creating developer-friendly and developer-centric approaches.

The Developers' Role in Usable Security and Privacy

The usable security and privacy field studies end-user behavior, perceptions, problems, and wishes. Its researchers inform system administrators and software developers of the results and make concrete suggestions as to how developers and administrators can make their software and services more functional for end users. A classic example of usable security research is the study of users' password behavior, which has produced recommendations on how administrators should set policies that enable users to create strong yet memorable passwords.



Developers Are Not the Enemy!
~~The Developer is the Enemy~~
People: The Strongest Link
~~Users Are Not The Enemy~~
~~Users Are The Weakest Link~~

