Serious Games for CyberSecurity

Raising security awareness of industrial software developers

Dr. Tiago Espinha Gasiba
tiago.gasiba@siemens.com
Siemens AG, Munich, Germany

IARIA

SIEMENS
Universität der Bundeswehr München
istar_iscte
Information Sciences, Technologies and Architecture Research Centre
fct
Fundação para a Ciência e a Tecnologia

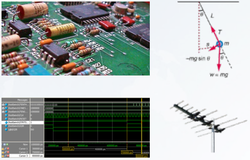
Bundesministerium für Bildung und Forschung
FKZ 13N16581
FKZ 13N16585
SIFO.de
contain

Presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 – October 03, 2024

© 2024 Siemens AG, Unrestricted

Who am I?

Background information…



Tech. at Escola Profissional D. Sancho II, Elvas, Portugal in 1994-1997
  Electrotechnical Professional
  Main Interest: **Electronics**

Eng. at FEUP, Porto, Portugal in 1997-2002
  Telecommunications, Electronics and Computers
  Main Interest: **Digital Systems**

FEUP
Universidade do Porto
Faculdade de Engenharia

M.Sc. at TUM, München, Germany in 2002-2004
  Communications Engineering
  Main Interest: **Forward Error-Correcting Codes**

Technical University of Munich
TUM

Researcher at Siemens Mobile in 2004-2007 ~~PhD~~
System Architect and Embedded Software Developer, Ericsson, in 2007-2013
System Design Engineer for JavaCard OS, NXP, Hamburg, in 2013-2014
PhD. at UniBwM, München, Germany in 2019-2021

Universität der Bundeswehr München

Fountain Codes

Tiago Gasiba
@2014

International Conference on Mobile and Ubiquitous Multimedia
10 Year Impact Award 2016

Senior Key Expert for Secure Software Development
Siemens AG, München, 2014-present

Certifications:
  CISSP, CISM, GXPN, SANS 642 Coin, …

SIEMENS

CSC: Takes you from ~~z~~ero to hero!
h

TEAM SECURITY CHAMPION WANTED
SUPER POWERS NOT REQUIRED
FULL SUPPORT PROVIDED BY ALL OF TEAM

SIEMENS
Ingenuity for life

DevSecOps calls for security heros…

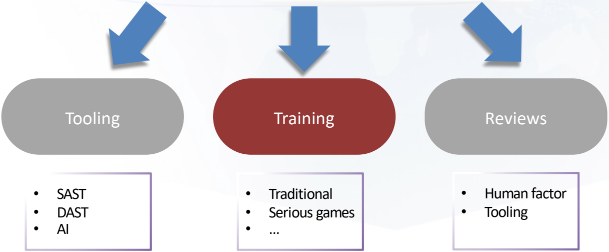
(Security) Ownership Culture

Corporate Technology

Overview of what is to come…

Agenda

How to address software vulnerabilities during the development phase?



Tooling

- SAST
- DAST
- AI

Training

- Traditional
- Serious games
- …

Reviews

- Human factor
- Tooling

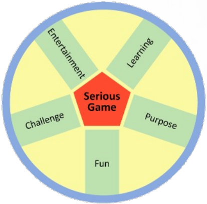
Agenda

1) What are serious games; what are the essential criteria for industrial serious games?
2) Examples of serious game design, developed through action design research.
3) (if we have time…) On the future of serious games using Artificial Intelligence

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Serious Games

What Constitutes a Serious Game? How are Serious Games different from Gamification?



Serious Games are not the same as Gamification

Purpose:

- **Serious Games**: Fully-fledged; medium to teach or achieve a purpose while engaging the player(s) in an experience
- **Gamification**: Adding game-like elements (e.g., points, badges, leaderboards) to a non-game event; it's more about enhancing existing processes

Experience:

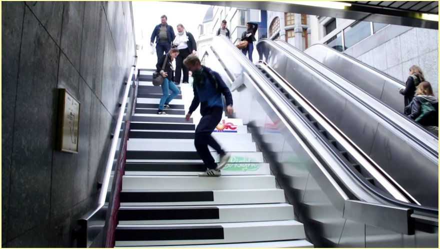
- **Serious Games**: Immersive environment for learning or skill development. Players might not even realize they are being educated or trained while playing.
- **Gamification**: Non-game-related core activity(ies); engagement achieved through elements such as rewards

Scope:

- **Serious Games**: Full game with mechanics, storylines, challenges, and goals tied to the purpose
- **Gamification**: Partial gaming elements, without building a complete game

Gamification

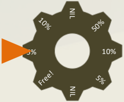
Examples of Gamification



Source: `https://www.straight.one/learning-base/mozart-stairs-in-stockholm-mehr-spass-am-treppensteigen/`

Employee of the Month

"Spin the Wheel and Win"

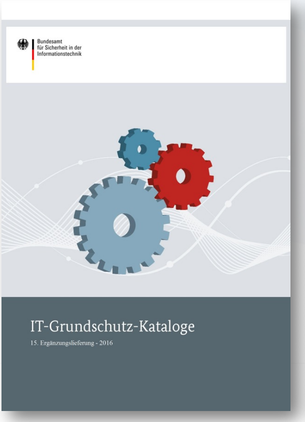


GitHub Profile Badges

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Serious Games

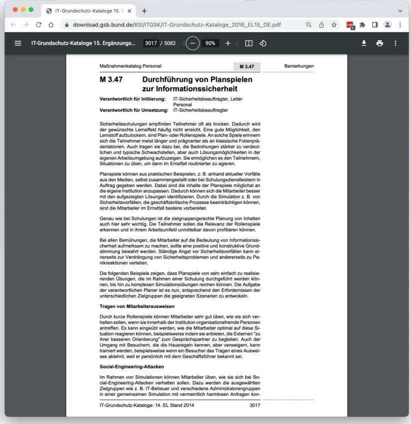
Recognized by the BSI IT-Grundschutz-Katalog (2016) as a mean to raise awareness



Bundesamt
für Sicherheit in der
Informationstechnik

IT-Grundschutz-Kataloge

15. Ergänzungslieferung - 2016


IT-Grundschutz-Kataloge 15. E
download.gsb.bund.de/BSI/ITGSK/IT-Grundschutz-Kataloge_2016_EL15_DE.pdf
IT-Grundschutz-Kataloge 15. Ergänzungs...
3017 / 5082
90%

Maßnahmenkatalog Personal **M 3.47** Bemerkungen

M 3.47 Durchführung von Planspielen zur Informationssicherheit

Verantwortlich für Initiierung: IT-Sicherheitsbeauftragter, Leiter Personal
Verantwortlich für Umsetzung: IT-Sicherheitsbeauftragter

Sicherheitsschulungen empfinden Teilnehmer oft als trocken. Dadurch wird der gewünschte Lerneffekt häufig nicht erreicht. Eine gute Möglichkeit, den Lernstoff aufzulockern, sind Plan- oder Rollenspiele. An solche Spiele erinnern sich die Teilnehmer meist länger und prägnanter als an klassische Folienpräsentationen. Auch tragen sie dazu bei, die Bedrohungen stärker zu verdeutlichen und typische Schwachstellen, aber auch Lösungsmöglichkeiten in der eigenen Arbeitsumgebung aufzuzeigen. Sie ermöglichen es den Teilnehmern, Situationen zu üben, um dann im Ernstfall routinierter zu agieren.

Planspiele können aus praktischen Beispielen, z. B. anhand aktueller Vorfälle aus den Medien, selbst zusammengestellt oder bei Schulungsdienstleistern in Auftrag gegeben werden. Dabei sind die Inhalte der Planspiele möglichst an die eigene Institution anzupassen. Dadurch können sich die Mitarbeiter besser mit den aufgezeigten Lösungen identifizieren. Durch die Simulation z. B. von Sicherheitsvorfällen, die geschäftskritische Prozesse beeinträchtigen können, sind die Mitarbeiter im Ernstfall bestens vorbereitet.

Genau wie bei Schulungen ist die zielgruppengerechte Planung von Inhalten auch hier sehr wichtig. Die Teilnehmer sollen die Relevanz der Rollenspiele erkennen und in ihrem Arbeitsumfeld unmittelbar davon profitieren können.

Bei allen Bemühungen, die Mitarbeiter auf die Bedeutung von Informationssicherheit aufmerksam zu machen, sollte eine positive und konstruktive Grundstimmung bewahrt werden. Ständige Angst vor Sicherheitsvorfällen kann einerseits zur Verdrängung von Sicherheitsproblemen und andererseits zu Panikreaktionen verleiten.

Die folgenden Beispiele zeigen, dass Planspiele von sehr einfach zu realisierenden Übungen, die im Rahmen einer Schulung durchgeführt werden können, bis hin zu komplexen Simulationsübungen reichen können. Die Aufgabe der verantwortlichen Planer ist es nun, entsprechend den Erfordernissen der unterschiedlichen Zielgruppen die geeigneten Szenarien zu entwickeln.

Tragen von Mitarbeiterausweisen

Durch kurze Rollenspiele können Mitarbeiter sehr gut üben, wie sie sich verhalten sollen, wenn sie innerhalb der Institution organisationsfremde Personen antreffen. Es kann eingeübt werden, wie die Mitarbeiter optimal auf diese Situation reagieren können, beispielsweise indem sie anbieten, die Externen "zu ihrer besseren Orientierung" zum Gesprächspartner zu begleiten. Auch der Umgang mit Besuchern, die die Hausregeln kennen, aber verweigern, kann trainiert werden, beispielsweise wenn ein Besucher das Tragen eines Ausweises ablehnt, weil er persönlich mit dem Geschäftsführer bekannt sei.

Social-Engineering-Attacken

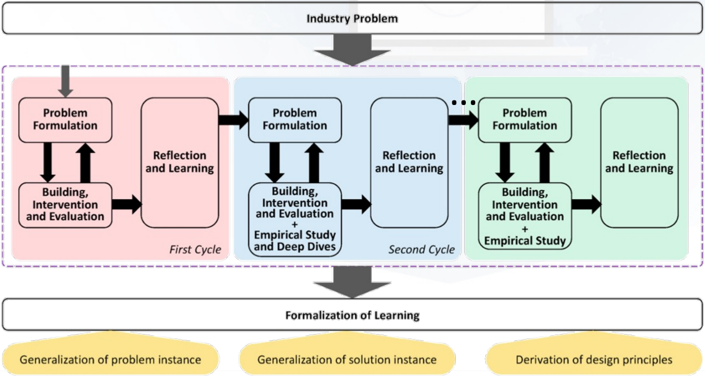
Im Rahmen von Simulationen können Mitarbeiter üben, wie sie sich bei Social-Engineering-Attacken verhalten sollen. Dazu werden die ausgewählten Zielgruppen wie z. B. IT-Betreuer und verschiedene Administratorengruppen in einer gemeinsamen Simulation mit vermeintlich harmlosen Anfragen kon-

IT-Grundschutz-Kataloge: 14. EL Stand 2014 3017

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Designing Industrial Serious Games

... using Action Design Research (Sein et al.)



Industry Problem

Problem Formulation

Building, Intervention and Evaluation

Reflection and Learning

First Cycle

Problem Formulation

Building, Intervention and Evaluation + Empirical Study and Deep Dives

Reflection and Learning

Second Cycle

...

Problem Formulation

Building, Intervention and Evaluation + Empirical Study

Reflection and Learning

Formalization of Learning

Generalization of problem instance

Generalization of solution instance

Derivation of design principles

T. Espinha Gasiba, *Raising Awareness on Secure Coding in the Industry through CyberSecurity Challenges*, Ph.D. dissertation, Fakultät für Informatik, Univ. Bundeswehr Munich, Germany, 2021.
adapted from Stein, M., Hendfridsson, O., Purao, S., Rossi, M., Lindfren, R., "*Action Design Research*", MIS Quarterly, 2011, 35(1), 37-56

Three Industrial Serious Games

Design best—practices as outcome from research conducted in the industry

Secure Software Development

Secure Deployment in the Cloud

Secure Code Review

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Design Requirement Criteria for Serious Games in Industry

Requirements and Background

On the Requirements for Serious Games geared towards Software Developers in the Industry

Tiago Gasiba
Siemens AG
Munich
tiago.gasiba@siemens.com

Kristian Beckers
Siemens AG
Munich
kristian.beckers@siemens.com

Santiago Suppan
Siemens AG
Munich
santiago.suppan@siemens.com

Filip Rezabek
Siemens AG
Munich
filip.rezabek@siemens.com

***Abstract*—Teaching industry staff on cybersecurity issues is a fundamental activity that must be undertaken in order to guarantee the delivery of successful and robust products to market. Much research attention has been devoted to this topic over the last years. However, the research which has been done has not focused on developing secure code in industrial environments. In this paper we take a look at the constraints and requirements for delivering a training, by means of cybersecurity challenges, that covers secure coding topics from an industry perspective. Using requirements engineering, we aim at understanding the design requirements for such challenges. Along the way, we give details on our experience of delivering cybersecurity challenges in an industrial setting and show the outcome and lessons learned. The proposed requirements for cybersecurity challenges geared towards software developers in an industrial environment are based on systematic literature review, interviews with security experts from the industry and semi-structured evaluation of participant feedback.**

***Index Terms*—cybersecurity, serious games, requirements, software developers, industry**

I. INTRODUCTION

In order to successfully deliver products to market, development in an industrial setting must follow existing laws, regulations and standards. Due to the increasing amount of successful hacking attacks, standardization bodies have been paying special attention towards introducing secure coding processes in the secure development lifecycle in the enterprise. Examples of standards that specifically mandate that companies, and therefore their software developer workforce, to follow secure coding guidelines and policies can be found, e.g. in [3, 4, 5, 6, 7, 8, 9, 10].

Additionally to this, there is currently a strong driving force in the industry, researchers and even governments called *digitalization*. Towards this goal, in Vision 2020 [1], different industrial partners have come together and committed to a charter of trust [2]. This document outlines how the industry is willing to address the issues inherent with cybersecurity as a result of digitalization. One of the stated key principles is focusing on the dedication and effort that shall be spent towards professional cybersecurity education and training.

From many possible forms of training industrial staff in cybersecurity awareness [15], we are particularly interested on training using Capture-the-Flag (CTF) exercises geared towards software developers. Our motivation to use CTF as a form of awareness training comes from the work of Graziotin et al. [48, 49]. Their work has investigated the fact that happy developers become better coders of software. CTFs are known to improve the happiness and satisfaction of the participants [27, 28]. Therefore we see an opportunity to, not only increase secure coding awareness, but also have a positive impact in the code produced by software developers. However, we are only interested in CTFs that are openly available (e.g. open source) and are not part of a commercial solution. The main reasons we choose to take this approach is because openly available challenges and CTFs: allow to easily develop and adapt own challenges, allow free exchange of challenges with external partners, existing challenges have been more scrutinized due to their free availability and they lead to lowering the overall cost for delivering IT security awareness training through CTF.

Recently lot of work has been directed towards investigating how these serious games can be designed, built and deployed in order to deliver cybersecurity awareness trainings [17, 25, 21, 33]. However, most of the work has not focused on the industry and its requirements. Even more surprising, however, is the fact that no previous systematic requirement elicitation (based e.g. on Ghanbari et. al. [12]) was found that addresses IT security training awareness through CTF-like serious games challenges for the industry. This poses a great challenge since many serious games are being evaluated, but none are developed following a requirements engineering methodology. Our requirement elicitation methodology includes requirements from CTF-like events, which additionally gathers requirements from the CTF participants themselves.Our requirements are from industry experts and therefore provide an excellent base line for other practitioners. Furthermore, we enable other practitioners with our requirements engineering methodology to elicit requirements for their individual CTFs.

Davis et. al [27] defines Capture-the-Flag events as having one of the following three types: *Attack-Defend*, *Attack-Only* and *Defend Only*. Typical topics covered in these CTFs range from web application security, cryptography, forensics, steganography, reverse engineering, mobile security and many other topics. Two prominent examples of Attack-Defend CTF are well-known (commercial) DEFCON CTF [29] and Hack-the-Box [30]. Here the participants own some infrastructure and have two different tasks: that they need to protect it against adversaries and they need to attack the infrastructure of their

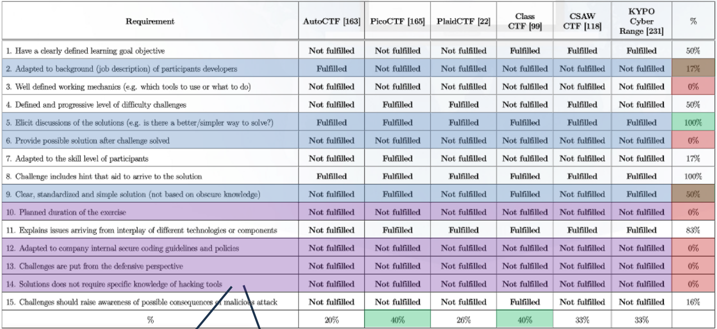
Requirement	**Background**
☐ Clear learning goal	serious game
☐ Adapted to job description	**targeting industry**
☐ Working mechanics of game	serious game
☐ Progressive difficulty	serious game mechanics
☐ Elicit discussions among participants	**targeting industry**
☐ Present compliant solutions	**targeting industry**
☐ Include solution hints	serious game mechanics
☐ Standardized solution	**targeting industry**
☐ Planning of duration of event	**targeting industry**
☐ Consider different technology stacks	general awareness
☐ Defensive challenges	**novel result / industry**
☐ Focus on secure coding guidelines	**targeting industry**
☐ No need to learn new tools	**targeting industry**
☐ Raise awareness of malicious attack	general awareness

CTF (Capture-the-Flag) games, although normally targeting cybersecurity, are not ideal to raise awareness of software developers on secure coding

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Empirical study: comparison of CTF vs Game Requirements

Results of Lightweight Literature Review



Requirement	AutoCTF [163]	PicoCTF [165]	PlaidCTF [22]	Class CTF [99]	CSAW CTF [118]	KYPO Cyber Range [231]	%
1. Have a clearly defined learning goal objective	Not fulfilled	Not fulfilled	Not fulfilled	Fulfilled	Fulfilled	Fulfilled	50%
2. Adapted to background (job description) of participants developers	Fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	17%
3. Well defined working mechanics (e.g. which tools to use or what to do)	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	0%
4. Defined and progressive level of difficulty challenges	Not fulfilled	Fulfilled	Fulfilled	Not fulfilled	Fulfilled	Not fulfilled	50%
5. Elicit discussions of the solutions (e.g. is there a better/simpler way to solve?)	Fulfilled	Fulfilled	Fulfilled	Fulfilled	Fulfilled	Fulfilled	100%
6. Provide possible solution after challenge solved	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	0%
7. Adapted to the skill level of participants	Not fulfilled	Fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	17%
8. Challenge includes hint that aid to arrive to the solution	Fulfilled	Fulfilled	Fulfilled	Fulfilled	Fulfilled	Fulfilled	100%
9. Clear, standardized and simple solution (not based on obscure knowledge)	Not fulfilled	Fulfilled	Not fulfilled	Fulfilled	Not fulfilled	Fulfilled	50%
10. Planned duration of the exercise	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	0%
11. Explains issues arriving from interplay of different technologies or components	Not fulfilled	Fulfilled	Fulfilled	Fulfilled	Fulfilled	Fulfilled	83%
12. Adapted to company internal secure coding guidelines and policies	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	0%
13. Challenges are put from the defensive perspective	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	0%
14. Solutions does not require specific knowledge of hacking tools	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	Not fulfilled	0%
15. Challenges should raise awareness of possible consequences of malicious attack	Not fulfilled	Not fulfilled	Not fulfilled	Fulfilled	Not fulfilled	Not fulfilled	16%
%	20%	40%	26%	40%	33%	33%	

Our experience shows that these requirements are fundamental for **industrial serious games**

“Important” industry requirements are not addressed

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Lesson in failure: when interactive storytelling fails

How not to design serious games

When Interactive Graphic Storytelling Fails

James Barela*, Tiago Gasiba†, Santiago Reinhard Suppan‡, Marc Berges§, Kristian Beckers¶
*§ Technische Universität München †‡¶ Siemens AG, München
* james.c.barela@tum.de † tiago.gasiba@siemens.com,
‡ Santiago.Suppan@Siemens.com, § Berges@TUM.de, ¶ kristian.beckers@siemens.com

***Abstract*—Many people are unaware of the digital dangers that lie around each cyber-corner. Teaching people how to recognize dangerous situations is crucial, especially for those who work on or with computers. We postulated that interactive graphic vignettes could be a great way to expose professionals to dangerous situations and demonstrate the effects of their choices in these situations. In that way, we aimed to inoculate employees against cybersecurity threats.**

We used the Comic-BEE platform to create interactive security awareness vignettes and evaluated for how employees of a major industrial company perceived them. For analysing the potential of these comics, we ran an evaluation study as part of a capture-the-flag (CTF) event, a interactive exercise for hacking vulnerable software. We evaluated whether the comics fulfilled our requirements based on the responses of the participants. We showed the comics, on various cybersecurity concepts, to 20 volunteers. In the context of a CTF event, our requirements were not fulfilled. Most participants considered the images distracting, stating a preference for text-only material.

***Index Terms*—Security training, Online platforms , Branching stories, Vignettes, Awareness training**

I. INTRODUCTION

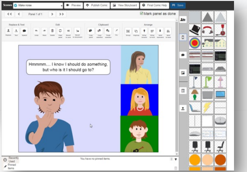
Our world has become vastly dependent on information technology. Millions of devices communicate every second in cyberspace. We are now connected to one another to a degree that seemed inconceivable just 20 years ago, but with so many connections, we are confronted with just as much, if not more, vulnerability; Hubbard & Seiersen, in their book *How to Measure Anything in Cybersecurity Risk* [1], lay out the case that the global attack surface is increasing from at least four perspectives: (1) the number of people on the Internet, (2) the number of online resources that each person is consuming, (3) the vulnerabilities that come with those people and resources, and (4) the risk from building online services on top of one another, which could result in a "breach cascade". Modern industrial systems and critical infrastructure should account for cyber threats, especially when corporate organizations, national economies, and public safety have been put at risk [2], [3]. As a result of these omnipresent risks, laws and standards have been created, such as NERC CIP-003-7, IEC 62443, and ISO 27001, that attempt to mitigate the risks posed by cyberattacks. These international standards acknowledge that among the weakest points in our security are not the technologies themselves, but the *people* behind the technologies. People often lack the knowledge and training to avoid even the simplest of hacker schemes, such as social engineering, or phishing attacks. Despite these requirements and all the best efforts of IT Security departments, organizations are still only as strong as the employees who regularly engage with the technologies. People are the front line of defense against cyber threats—in particular, the engineers, managers, and other staff who interact directly with information systems and their security measures on a daily basis. Therefore, these international standards require that such personnel are well-trained, that they are aware of the risks, and that they are prepared and have the resources to mitigate those risks.

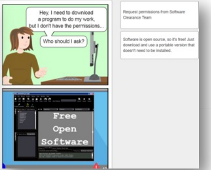
Unfortunately, this is as far as mandates, requirements, and laws can go; companies have to interpret and implement them, as well as measure if the training yields the desired results. If there is an attack, and it is not averted because employees did not receive the right training, then it will have already been too late.

The question is, how do we train companies' most valuable resource, their employees? And how do we ensure that the training has been efficient and cost-effective? National institutions exist that are dedicated to the sole purpose of education (the Department of Education, USA; Bundes Ministerium für Bildung und Forschung, Germany; etc). But what this work needs to take into consideration is that the people in need of educating are (1) in the Age of Information, a different world from anything that has ever existed before; and (2) that they are not children being prepared for the workplace, but adults who are already *in* the workplace. The old paradigms are falling out of favor and it would be prudent to take advantage of the resources that are available, i.e., computers and the Internet. Security training needs to be close to the context employees work in. Therefore, our training content was based on interviews with security experts working in the company itself. Training also needs to be cost-effective and scale well, which is why we selected an online platform.

In addition, understanding computers to be the main method for delivery (as opposed to lectures or in-person presentations) the media should take full advantage of the properties of a computer. Given Mayer's extensive work on learning with multimedia, this paper will explore a new method, which shows promise for training employees effectively by reducing cognitive load [4], [5]. Our method uses a vignette modality so that participants had to further engage with the comic in a way that implements situated learning [6]. Furthermore, we evaluate the claims made by the developers of the comic-creation platform, especially with regard to the time and skills

- **Lesson learned:** not all challenges are suitable as CyberSecurity Challenges
- Challenge type depends on environment
- Capture-the-Flag creates a "competitive environment"
  - Players focused on winning rather than learning
- Root cause of failure
  - Not designed to address software developers
  - No secure coding guidelines, not adapted to job




Hey, I need to download a program to do my work, but I don't have the permissions...
Who should I ask?
Free Open Software

Request permissions from Software Clearance Team

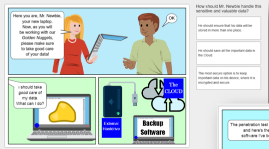
Software is open source, so it's free! Just download and use a portable version that doesn't need to be installed.

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

When Interactive Storytelling Fails

Five Scenarios

Backup and Restore


Here you are, Mr. Newbie, your new laptop. Now, as you will be working with our *Golden Nuggets*, please make sure to take good care of your data!
OK
I should take *good care* of my data. What can I do?
External Harddrive
Backup Software
The CLOUD

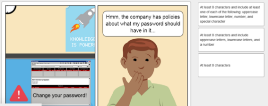
How should Mr. Newbie handle this sensitive and valuable data?

He should ensure that his data will be stored in more than one place.

He should save all the important data in the Cloud.

The most secure option is to keep important data on his device, where it is encrypted and secure.

Password Management

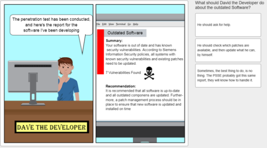

KNOWLEDGE IS POWER
Change your password!
51:54:7
Hmm, the company has policies about what my password should have in it...

At least 8 characters and include at least one of each of the following: uppercase letter, lowercase letter, number, and special character

At least 8 characters and include uppercase letters, lowercase letters, and a number

At least 8 characters

Patch Management


The penetration test has been conducted, and here's the report for the software I've been developing
DAVE THE DEVELOPER
File Edit View Terminal Go Help

Outdated Software

Summary:
Your software is out of date and has known security vulnerabilities. According to Siemens Information Security policies, all systems with known security vulnerabilities and existing patches need to be updated.

7 Vulnerabilities Found ☠

Recommendation:
It is recommended that all software is up-to-date and all outdated componens are updated. Furthermore, a patch management process should be in place to ensure that new software is updated and installed on time

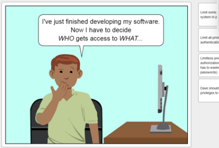
What should David the Developer do about the outdated Software?

He should ask for help.

He should check which patches are available, and then update what he can, by himself.

Sometimes, the best thing to do, is no thing. The PSSE probably got this same report, they will know how to handle it.

Principle of Least Privilege


I've just finished developing my software. Now I have to decide *WHO* gets access to *WHAT*...

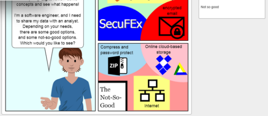
Limit some system in p

Limit all privileges; everything requires authentication and authorization

Limitless privileges; authentication and authorization is not required (so no one has to waste time remembering passwords)

Dave should use his discretion to dole out privileges to those who need them.

Where to Share Data


ey there! exploratory comic, wrong answers, ing to review some concepts and see what happens!

I'm a software engineer, and I need to share my data with an analyst. Depending on your needs, there are some good options, and some not-so-good options. Which would you like to see?
The Good
syncplicity
SecuFEx
encrypted email
Compress and password protect
ZIP
Online cloud-based storage
The Not-So-Good
Internet

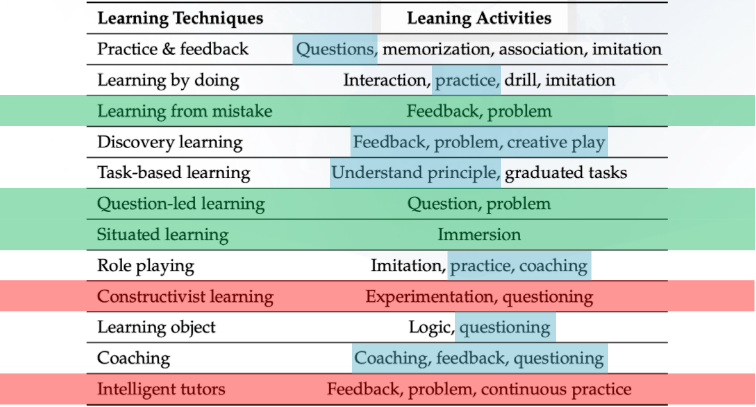
The Good

Not so good

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Learning Techniques and Activities for Serious Games

Mapping between Techniques and Activities (Silva, 2020)



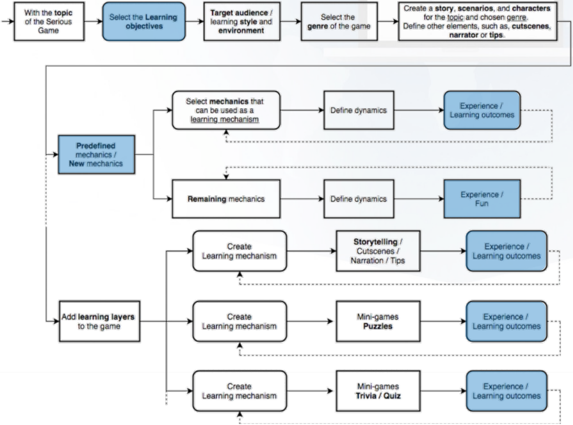
Learning Techniques	Leaning Activities
Practice & feedback	Questions, memorization, association, imitation
Learning by doing	Interaction, practice, drill, imitation
Learning from mistake	Feedback, problem
Discovery learning	Feedback, problem, creative play
Task-based learning	Understand principle, graduated tasks
Question-led learning	Question, problem
Situated learning	Immersion
Role playing	Imitation, practice, coaching
Constructivist learning	Experimentation, questioning
Learning object	Logic, questioning
Coaching	Coaching, feedback, questioning
Intelligent tutors	Feedback, problem, continuous practice

`Silva FGM. Practical Methodology for the Design of Educational Serious Games. Information. 2020; 11(1):14.`

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Designing Serious Games

A framework by Silva, 2020



With the **topic** of the Serious Game

Select the **Learning objectives**

Target audience / learning **style** and **environment**

Select the **genre** of the game

Create a **story**, **scenarios**, and **characters** for the topic and chosen genre. Define other elements, such as, **cutscenes**, **narrator** or **tips**.

Predefined mechanics / **New** mechanics

Select **mechanics** that can be used as a learning mechanism

Define dynamics

Experience / Learning outcomes

Remaining mechanics

Define dynamics

Experience / Fun

Add **learning layers** to the game

Create Learning mechanism

Storytelling / Cutscenes / Narration / Tips

Experience / Learning outcomes

Create Learning mechanism

Mini-games **Puzzles**

Experience / Learning outcomes

Create Learning mechanism

Mini-games **Trivia / Quiz**

Experience / Learning outcomes

Cost

- **unproductive hours**
- **extra costs**
- **maintenance / adaptation**

Benefit

- **real-world examples**
- **teamwork**
- **standard compliance**
- increased awareness / security

Silva FGM. Practical Methodology for the Design of Educational Serious Games. Information. 2020; 11(1):14.

Industrial Serious Games

Sifu – Raising Software Developers' Awareness of Secure Coding Guidelines

CSC

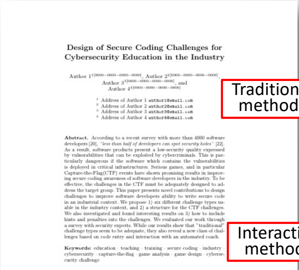
Cyber**S**ecurity **C**hallenges

T. Espinha Gasiba, *Raising Awareness on Secure Coding in the Industry through CyberSecurity Challenges*, Ph.D. dissertation, Fakultät für Informatik, Univ. Bundeswehr Munich, Germany, 2021.

- You get a vulnerable code snippet
- Task is to find vulnerabilities and fix them
- Failed solutions uncover hints
- Supports: C, C++, Java, Python, Terraform

Industrial Serious Games

Suitable Challenge Types



Design of Secure Coding Challenges for Cybersecurity Education in the Industry

Author 1[1][0000-0000-0000-0000], Author 2[2][0000-0000-0000-0000],
Author 3[3][0000-0000-0000-0000], and
Author 4[4][0000-0000-0000-0000]

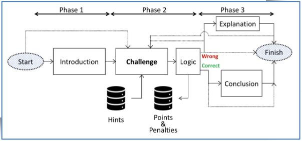
[1] Address of Author 1 author1@email.com
[2] Address of Author 2 author2@email.com
[3] Address of Author 3 author3@email.com
[4] Address of Author 4 author4@email.com

Abstract. According to a recent survey with more than 4000 software developers [20], *"less than half of developers can spot security holes"* [22]. As a result, software products present a low-security quality expressed by vulnerabilities that can be exploited by cybercriminals. This is particularly dangerous if the software which contains the vulnerabilities is deployed in critical infrastructures. Serious games, and in particular Capture-the-Flag(CTF) events have shown promising results in improving secure coding awareness of software developers in the industry. To be effective, the challenges in the CTF must be adequately designed to address the target group. This paper presents novel contributions to design challenges to improve software developers ability to write secure code in an industrial context. We propose 1) six different challenge types usable in the industry context, and 2) a structure for the CTF challenges. We also investigated and found interesting results on 3) how to include hints and penalties into the challenges. We evaluated our work through a survey with security experts. While our results show that "traditional" challenge types seem to be adequate, they also reveal a new class of challenges based on code entry and interaction with an automated coach.

Keywords: education · teaching · training · secure coding · industry · cybersecurity · capture-the-flag · game analysis · game design · cybersecurity challenge

Traditional methods

Interactive methods



- That kinds of challenge types are suitable for industrial players?

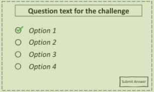


Fig. 4. Single-Choice Question

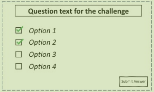
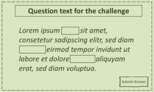


Fig. 5. Multiple-Choice Question



Lorem ipsum ____ sit amet, consetetur sadipscing elitr, sed diam ____ eirmod tempor invidunt ut labore et dolore ____ aliquyam erat, sed diam voluptua.

Fig. 6. Text-Entry Question

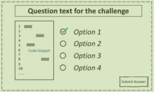
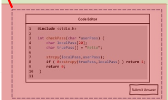


Fig. 7. Code-Snippet Question



```
#include <stdio.h>

int checkPass(char *userPass) {
    char localPass[20];
    char truePass[] = "hello";

    strcpy(localPass,userPass);
    if ( 0==strcpy(truePass,localPass) ) return 1;
    return 0;
}
```

Fig. 8. Code-Entry Challenge

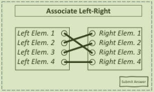
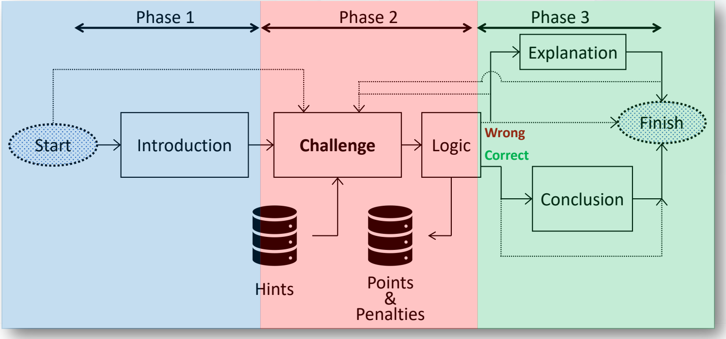


Fig. 9. Associate Left-Right

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

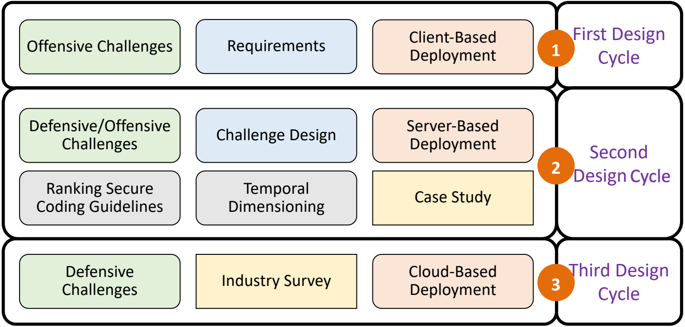
Structure of Serious Game Challenge

Challenge Structure in Three Phases



Previous Work: Design through Action Design Research

Overview of Research Activities



Offensive Challenges
Requirements
Client-Based Deployment
1 First Design Cycle

Defensive/Offensive Challenges
Challenge Design
Server-Based Deployment
Ranking Secure Coding Guidelines
Temporal Dimensioning
Case Study
2 Second Design Cycle

Defensive Challenges
Industry Survey
Cloud-Based Deployment
3 Third Design Cycle

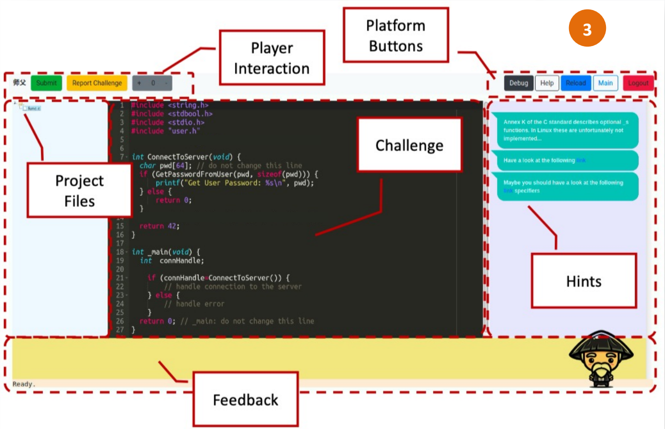
Legend:

- 🟨 Empirical Studies
- 🟩 Challenges
- 🟦 Design Studies
- 🟧 Deployment
- ⬜ Deep Dive

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Sifu Platform (Defensive Challenges)

Third Design Cycle: developed platform, released under MIT license



```
#include <string.h>
#include <stdbool.h>
#include <stdio.h>
#include "user.h"


int ConnectToServer(void) {
  char pwd[64]; // do not change this line
  if (GetPasswordFromUser(pwd, sizeof(pwd))) {
      printf("Get User Password: %s\n", pwd);
  } else {
      return 0;
  }

  return 42;
}

int _main(void) {
  int  connHandle;

    if (connHandle=ConnectToServer()) {
        // handle connection to the server
    } else {
        // handle error
    }
  return 0; // _main: do not change this line
}
```

Annex K of the C standard describes optional _s functions. In Linux these are unfortunately not implemented...

Have a look at the following link

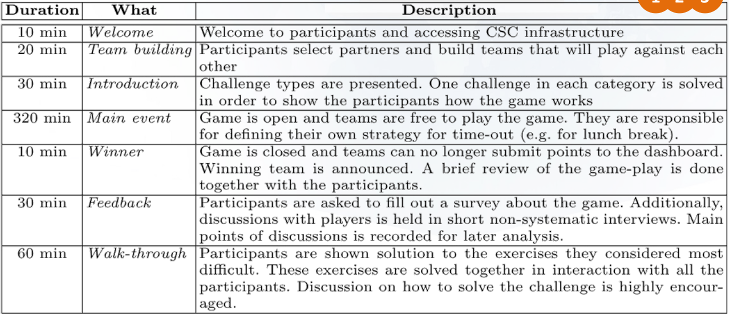
Maybe you should have a look at the following link specifiers

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

CyberSecurity Challenges

Embedded in a Corporate Training Event; Agenda and Empirical Studies

1 2 3

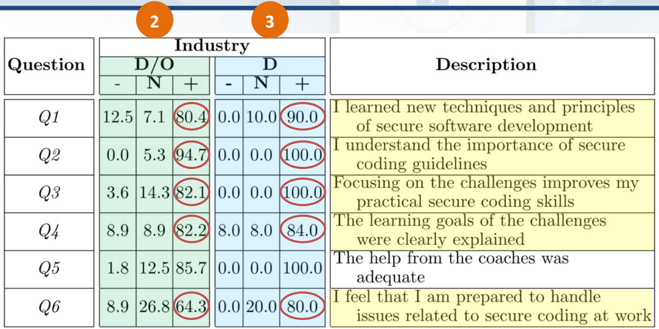


Duration	What	Description
10 min	*Welcome*	Welcome to participants and accessing CSC infrastructure
20 min	*Team building*	Participants select partners and build teams that will play against each other
30 min	*Introduction*	Challenge types are presented. One challenge in each category is solved in order to show the participants how the game works
320 min	*Main event*	Game is open and teams are free to play the game. They are responsible for defining their own strategy for time-out (e.g. for lunch break).
10 min	*Winner*	Game is closed and teams can no longer submit points to the dashboard. Winning team is announced. A brief review of the game-play is done together with the participants.
30 min	*Feedback*	Participants are asked to fill out a survey about the game. Additionally, discussions with players is held in short non-systematic interviews. Main points of discussions is recorded for later analysis.
60 min	*Walk-through*	Participants are shown solution to the exercises they considered most difficult. These exercises are solved together in interaction with all the participants. Discussion on how to solve the challenge is highly encouraged.

Survey: opt-in with informed consent and anonymized answers

Evaluation of Serious Game

Success in the industry



②

③

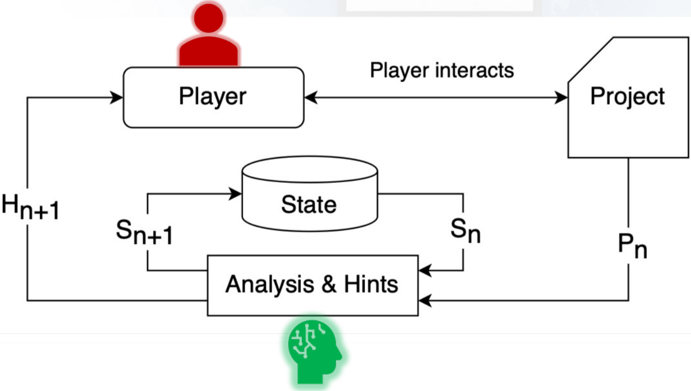
Question	Industry						Description
	D/O			D			
	-	N	+	-	N	+	
Q1	12.5	7.1	80.4	0.0	10.0	90.0	I learned new techniques and principles of secure software development
Q2	0.0	5.3	94.7	0.0	0.0	100.0	I understand the importance of secure coding guidelines
Q3	3.6	14.3	82.1	0.0	0.0	100.0	Focusing on the challenges improves my practical secure coding skills
Q4	8.9	8.9	82.2	8.0	8.0	84.0	The learning goals of the challenges were clearly explained
Q5	1.8	12.5	85.7	0.0	0.0	100.0	The help from the coaches was adequate
Q6	8.9	26.8	64.3	0.0	20.0	80.0	I feel that I am prepared to handle issues related to secure coding at work

- **More than 80%** of evaluation criteria shows positive results
- Usefulness: established in the official company training curriculum
- Success achieved in industry and academia
- Note: *not always possible to achieve 100% agreement in every category*

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

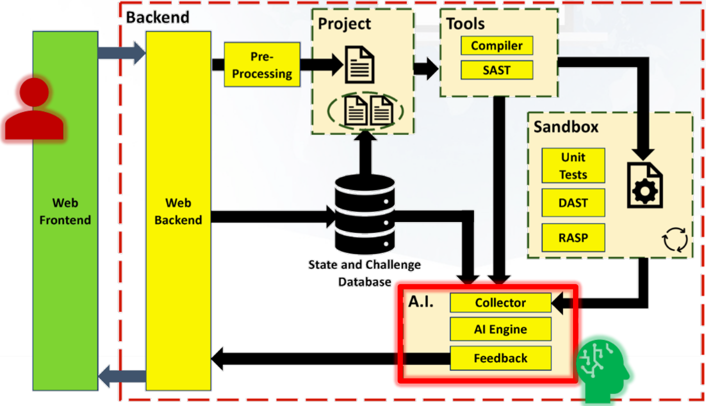
CyberSecurity Challenges

How to play a challenge?



Automatic Evaluation of Challenges

Architecture: Frontend/Backend



© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Industrial Serious Games

CATS – Raising Awareness on Responsibilities for Secure Cloud Deployments

CATS

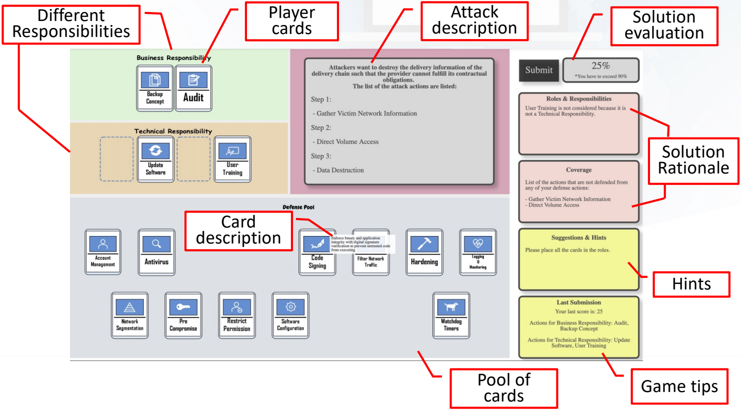
Cloud of Assets and Threats

- Electronic (virtual) table-top game
- Goal is to build a defense strategy
- Failed solutions provide hints
- Teach separation of responsibilities and defense strategies against typical attacks

Ongoing work by: Tiange Zhao

Cloud of Assets and Threats

A Virtual Table-Top Game



Business Responsibility

Backup Concept
Audit

Technical Responsibility

Update Software
User Training

Attackers want to destroy the delivery information of the delivery chain such that the provider cannot fulfill its contractual obligations.
The list of the attack actions are listed:

Step 1:

- Gather Victim Network Information

Step 2:

- Direct Volume Access

Step 3:

- Data Destruction

Defense Pool

Account Management
Antivirus
Code Signing
Filter Network Traffic
Hardening
Logging & Monitoring
Network Segmentation
Pre Compromise
Restrict Permission
Software Configuration
Watchdog Timers

Enforce binary and application integrity with digital signature verification to prevent untrusted code from executing

Submit

25%
*You have to exceed 90%

Roles & Responsibilities

User Training is not considered because it is not a Technical Responsibility.

Coverage

List of the actions that are not defended from any of your defense actions:

- Gather Victim Network Information
- Direct Volume Access

Suggestions & Hints

Please place all the cards in the roles.

Last Submission

Your last score is: 25

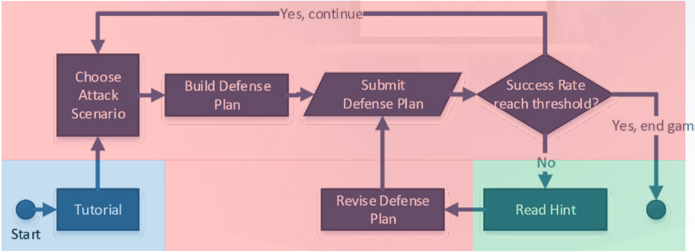
Actions for Business Responsibility: Audit, Backup Concept

Actions for Technical Responsibility: Update Software, User Training

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Cloud of Assets and Threats

Game Mechanics



Yes, continue

Choose Attack Scenario

Build Defense Plan

Submit Defense Plan

Success Rate reach threshold?

Yes, end game

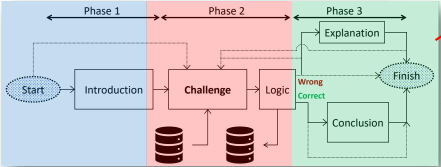
No

Start

Tutorial

Revise Defense Plan

Read Hint



Phase 1

Phase 2

Phase 3

Start

Introduction

Challenge

Logic

Wrong

Correct

Explanation

Conclusion

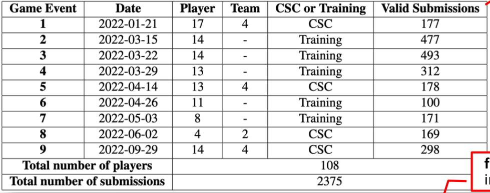
Finish

Where did we see this model?...

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Cloud of Assets and Threats

Game Evaluation in an Industrial Setting

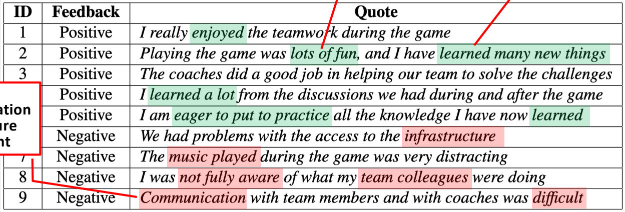


Game Event	Date	Player	Team	CSC or Training	Valid Submissions
1	2022-01-21	17	4	CSC	177
2	2022-03-15	14	-	Training	477
3	2022-03-22	14	-	Training	493
4	2022-03-29	13	-	Training	312
5	2022-04-14	13	4	CSC	178
6	2022-04-26	11	-	Training	100
7	2022-05-03	8	-	Training	171
8	2022-06-02	4	2	CSC	169
9	2022-09-29	14	4	CSC	298
Total number of players		108			
Total number of submissions		2375			

Interaction and engagement

fun is also important

Crucial positive aspect: **learning**



ID	Feedback	Quote
1	Positive	*I really enjoyed the teamwork during the game*
2	Positive	*Playing the game was lots of fun, and I have learned many new things*
3	Positive	*The coaches did a good job in helping our team to solve the challenges*
	Positive	*I learned a lot from the discussions we had during and after the game*
	Positive	*I am eager to put to practice all the knowledge I have now learned*
	Negative	*We had problems with the access to the infrastructure*
7	Negative	*The music played during the game was very distracting*
8	Negative	*I was not fully aware of what my team colleagues were doing*
9	Negative	*Communication with team members and with coaches was difficult*

Watch out for
- **Communication**
- **Infrastructure**
- **Environment**

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

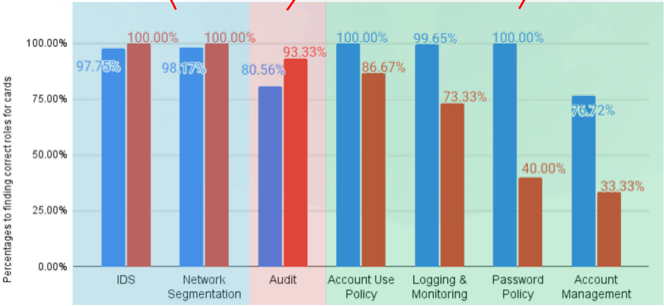
Cloud of Assets and Threats

Study on effects before/after playing the game

Statistical noise

Issues requiring special attention

Significant improvements



NOTE:
Long-term effects are **difficult/impossible** to determine!

Results after playing game

Results before playing game

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Industrial Serious Games

DuckDebugger – Empowering Developers to Perform Code Review

DuckDebugger

Discover Vulnerabilities and **D**efend your code

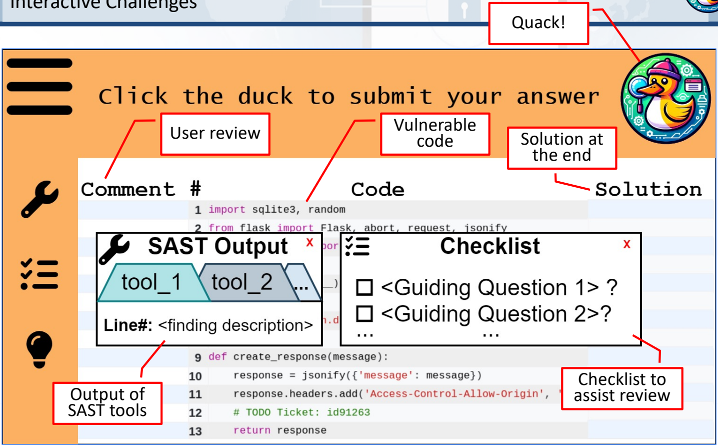
…because code reviews matter!

Ongoing work by: Andrei-Cristian Iosif

- Player is given a vulnerable code snippet
- Task is to review code and identify vulnerability-hotspots
- Hints in the form of checklist and SAST tool output
- Supports: C, C++, Java, JavaScript, Python, Golang, C#, …

DuckDebugger – Discover and Defend

Interactive Challenges



Quack!

Click the duck to submit your answer

User review

Vulnerable code

Solution at the end

```
import sqlite3, random
from flask import Flask, abort, request, jsonify
```

```
def create_response(message):
    response = jsonify({'message': message})
    response.headers.add('Access-Control-Allow-Origin', '
    # TODO Ticket: id91263
    return response
```

SAST Output x

Line#: <finding description>

Checklist x

☐ <Guiding Question 1> ?
☐ <Guiding Question 2>?
… …

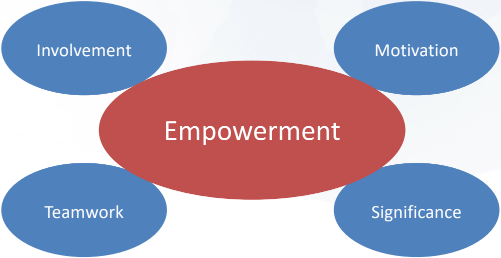
Output of SAST tools

Checklist to assist review

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

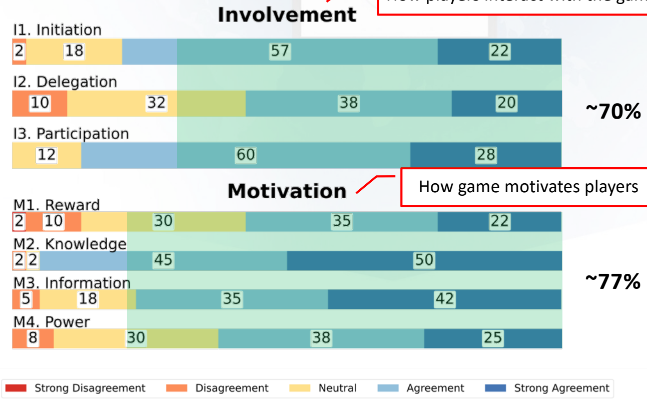
DuckDebugger – Discover and Defend

Dimensions of Empowerment



DuckDebugger – Discover and Defend

Preliminary Results from Interventions in Industry



How players interact with the game

How game motivates players

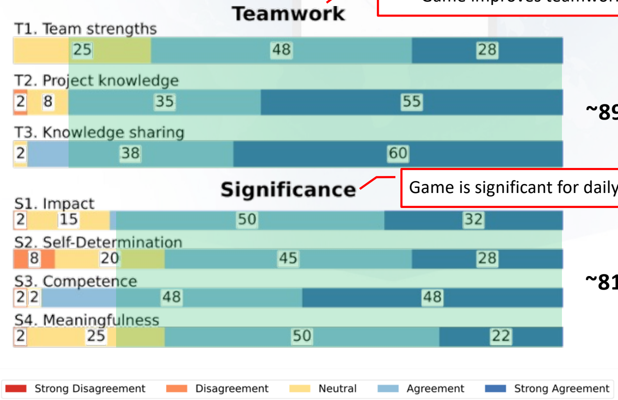
Results collected from 200+ participants, across 13 game events

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

DuckDebugger – Discover and Defend

Preliminary Results from Interventions in Industry

Game improves teamwork



~89%

Game is significant for daily job

~81%

Results collected from 200+ participants, across 13 game events

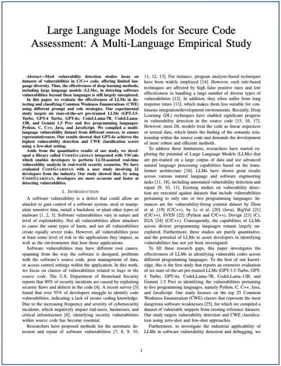
© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Large Language Models and Secure Software Development

Large Language Models

Large Language Models and Secure Software Development

CodeGuardian – (*Work Currently Under Review*)



Large Language Models for Secure Code Assessment: A Multi-Language Empirical Study

***Abstract*—Most vulnerability detection studies focus on datasets of vulnerabilities in C/C++ code, offering limited language diversity. Thus, the effectiveness of deep learning methods, including large language models (LLMs), in detecting software vulnerabilities beyond these languages is still largely unexplored.**

In this paper, we evaluate the effectiveness of LLMs in detecting and classifying Common Weakness Enumerations (CWE) using different prompt and role strategies. Our experimental study targets six state-of-the-art pre-trained LLMs (GPT-3.5-Turbo, GPT-4 Turbo, GPT-4o, CodeLLama-7B, CodeLLama-13B, and Gemini 1.5 Pro) and five programming languages: Python, C, C++, Java, and JavaScript. We compiled a multi-language vulnerability dataset from different sources, to ensure representativeness. Our results showed that GPT-4o achieves the highest vulnerability detection and CWE classification scores using a few-shot setting.

Aside from the quantitative results of our study, we developed a library called CODEGUARDIAN integrated with VSCode which enables developers to perform LLM-assisted real-time vulnerability analysis in real-world security scenarios. We have evaluated CODEGUARDIAN with a user study involving 22 developers from the industry. Our study showed that, by using CODEGUARDIAN, developers are more accurate and faster at detecting vulnerabilities.

I. INTRODUCTION

A software vulnerability is a defect that could allow an attacker to gain control of a software system, steal or manipulate sensitive data, install a backdoor, or plant other types of malware [1, 2, 3]. Software vulnerabilities vary in nature and level of exploitability. Not all vulnerabilities allow attackers to cause the same types of harm, and not all vulnerabilities create equally severe risks. However, all vulnerabilities pose at least some level of risk to the applications they impact, as well as the environments that host those applications.

Software vulnerabilities may have different root causes, spanning from the way the software is designed, problems with the software's source code, poor management of data, or access control settings within the application. In this work, we focus on classes of vulnerabilities related to *bugs in the source code*. The U.S. Department of Homeland Security reports that 90% of security incidents are caused by exploiting security flaws and defects in the code [4]. A recent survey [5] found that over 55% of developers struggle to identify code vulnerabilities, indicating a lack of secure coding knowledge. Due to the increasing frequency and severity of cybersecurity incidents, which negatively impact end-users, businesses, and critical infrastructure [6], identifying security vulnerabilities within source code has become essential.

Researchers have proposed methods for the automatic detection and repair of software vulnerabilities [7, 8, 9, 10, 11, 12, 13]. For instance, program analysis-based techniques have been widely employed [14]. However, such rule-based techniques are affected by high false positive rates and low effectiveness in handling a large number of diverse types of vulnerabilities [12]. In addition, they often suffer from long response times [13], which makes them less suitable for continuous integration/development environments. Recently, Deep Learning (DL) techniques have enabled significant progress in vulnerability detection in the source code [15, 16, 17]. However, most DL models treat the code as linear sequences or textual data, which limits the finding of the semantic relationship within the source code and demands the development of more robust and efficient methods.

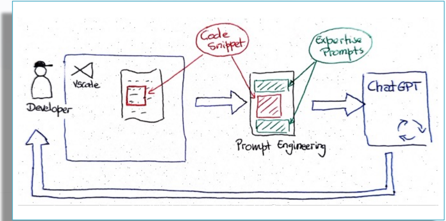
To address these limitations, researchers have started exploring the potential of Large Language Models (LLMs) that are pre-trained on a large corpus of data and use advanced natural language processing capabilities based on the transformer architecture [18]. LLMs have shown great results across various natural language and software engineering tasks [11, 18], including automated vulnerability detection and repair [9, 10, 11]. Existing studies on vulnerability detection are executed against datasets that include vulnerabilities pertaining to only one or two programming languages. Instances are the vulnerability-fixing commit dataset by Zhou et al. [19] (C/C++), by Li et al. [20] (Java), BigVul [21] (C/C++), SVEN [22] (Python and C/C++), Devign [23] (C), D2A [24] (C/C++). Consequently, the capabilities of LLMs across diverse programming languages remain largely unexplored. Furthermore, these studies are purely quantitative, and the potential of LLMs to assist developers in identifying vulnerabilities has not yet been investigated.

To fill these research gaps, this paper investigates the effectiveness of LLMs in identifying vulnerable codes across different programming languages. To the best of our knowledge, this is the first study that reports an extensive evaluation of six state-of-the-art pre-trained LLMs (GPT-3.5-Turbo, GPT-4 Turbo, GPT-4o, CodeLLama-7B, CodeLLama-13B, and Gemini 1.5 Pro) in identifying the vulnerabilities pertaining to five programming languages, namely Python, C, C++, Java, and JavaScript. Our study focuses on the top 25 Common Weakness Enumeration (CWE) classes that represent the most dangerous software weaknesses [25], for which we compiled a dataset of vulnerable snippets from existing reference datasets. Our study targets vulnerability detection and CWE classification using zero-shot and few-shot approaches.

Furthermore, to investigate the industrial applicability of LLMs in software vulnerability detection and debugging, we

1

- Using LLM models to assist software developers to write secure code
- Ease of use: seamless integration in VisualStudio Code as a plugin



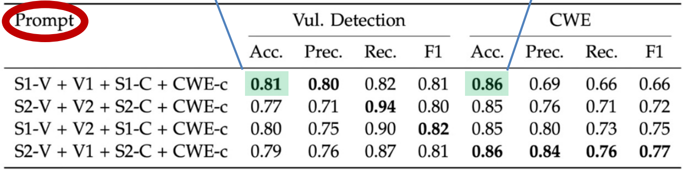
© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Large Language Models and Secure Software Development

How well does ChatGPT perform (i.e. can detect vulnerabilities)?

About **one in every five** vulnerabilities is not found!

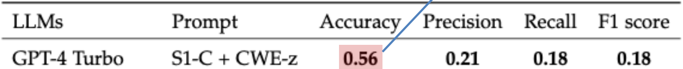
About **one in every ten** vulnerabilities is wrongly labeled



Prompt	Vul. Detection				CWE			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
S1-V + V1 + S1-C + CWE-c	**0.81**	**0.80**	0.82	0.81	**0.86**	0.69	0.66	0.66
S2-V + V2 + S2-C + CWE-c	0.77	0.71	**0.94**	0.80	0.85	0.76	0.71	0.72
S1-V + V2 + S1-C + CWE-c	0.80	0.75	0.90	**0.82**	0.85	0.80	0.73	0.75
S2-V + V1 + S2-C + CWE-c	0.79	0.76	0.87	0.81	**0.86**	**0.84**	**0.76**	**0.77**

Naïve Usage

About **four in every ten** vulnerabilities is are found!



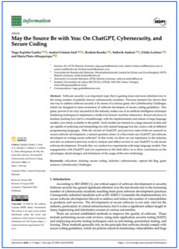
LLMs	Prompt	Accuracy	Precision	Recall	F1 score
GPT-4 Turbo	S1-C + CWE-z	**0.56**	**0.21**	**0.18**	**0.18**

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

A Simple ChatGPT Experiment

CWE 208 - Observable Timing Discrepancy

```
int is_equal(const char* a, const char* b, size_t len) {
    for (size_t i = 0; i < len ; i++) {
        if (a[i] != b[i])
            return 1;
    }
    return 0;
}
```



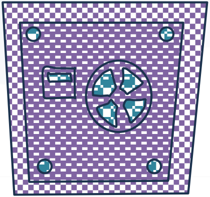
information MDPI

Article

May the Source Be with You: On ChatGPT, Cybersecurity, and Secure Coding

Tiago Espinha Gasiba, Andrei-Cristian Iosif, Ibrahim Kessba, Sathwik Amburi, Ulrike Lechner and Maria Pinto-Albuquerque

Code can be trivially broken!



Espinha Gasiba, T.; Iosif, A.-C.; Kessba, I.; Amburi, S.; Lechner, U.; Pinto-Albuquerque, M. May the Source Be with You: On ChatGPT, Cybersecurity, and Secure Coding. *Information* **2024**, *15*, 572. https://doi.org/10.3390/info15090572

A Simple ChatGPT Experiment

CWE 208 - Observable Timing Discrepancy – ChatGPT’s solution

```
#include <openssl/crypto.h>

int is_equal(const char* a, const char* b, size_t len) {
    if(a == NULL || b == NULL) return -1;
    size_t a_len = strnlen(a, len);
    size_t b_len = strnlen(b, len);
    if (len == 0 || len > a_len || len > b_len) {
        len = a_len < b_len ? a_len : b_len;
    }
    if (a[a_len] != ’\0’ || b[b_len] != ’\0’) {
        return -1;
    }
    return CRYPTO_memcmp(a, b, len);
}
```

Technically correct, but unnecessary

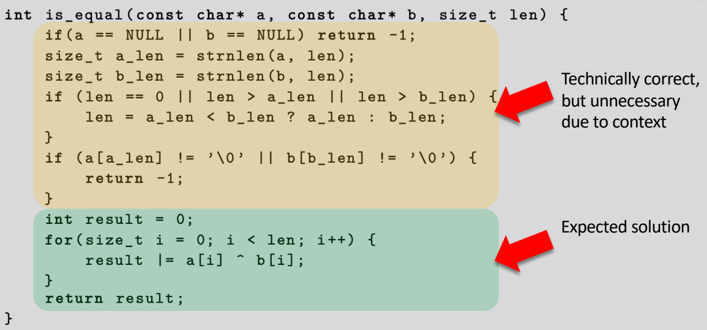
Technically correct, but using extra libraries

- ❑ **Using ChatGPT, solution had to be forced** to obtain a solution
    - ➢ Explicitly state that there is a timing vulnerability in the code
- ❑ AI’s solution introduces **dependency on 3rd party component**
    - ➢ In an industrial context, this can be undesired, e.g. due to **licensing issues**

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

A Simple ChatGPT Experiment

CWE 208 - Observable Timing Discrepancy – ChatGPT’s 2nd solution



```
int is_equal(const char* a, const char* b, size_t len) {
    if(a == NULL || b == NULL) return -1;
    size_t a_len = strnlen(a, len);
    size_t b_len = strnlen(b, len);
    if (len == 0 || len > a_len || len > b_len) {
        len = a_len < b_len ? a_len : b_len;
    }
    if (a[a_len] != ’\0’ || b[b_len] != ’\0’) {
        return -1;
    }
    int result = 0;
    for(size_t i = 0; i < len; i++) {
        result |= a[i] ^ b[i];
    }
    return result;
}
```

Technically correct, but unnecessary due to context

Expected solution

- ☐ Additional solution had to be **forced** to *remove dependency on 3rd party library*
- ☐ Solution still exhibits several problems (after more than **15 interactions**)
  - ➢ *Potentially unnecessary checks*
  - ➢ *Maintenance of code can be problematic*

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Observed problems with AI and Secure Coding

Short summary of identified issues

> pre-conditions
> superfluous corrections

Missing context

> Different behaviour
> no-go for safety-critical systems

Changed semantics

> Efficiency not guaranteed
> affecting real-time systems

Code complexity

Learning inference (abuse/leakage)

> Leakage of IPR, company internal secrets

Limited learning

> Findings limited to learning data set

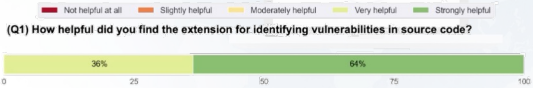
Maintainability

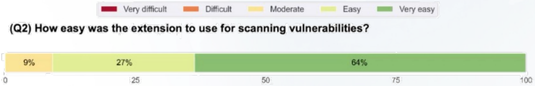
> e.g.missing comments

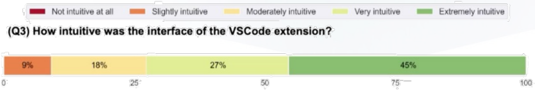
© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

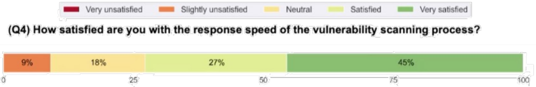
Large Language Models and Secure Software Development

... we are sure that this will become more and more ubiquos


(Q1) How helpful did you find the extension for identifying vulnerabilities in source code?


(Q2) How easy was the extension to use for scanning vulnerabilities?

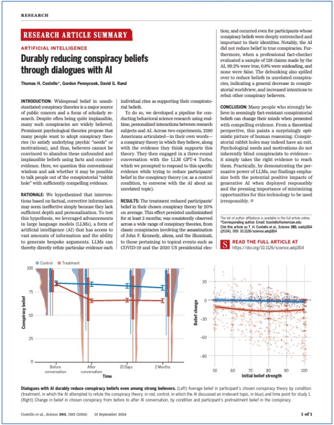

(Q3) How intuitive was the interface of the VSCode extension?


(Q4) How satisfied are you with the response speed of the vulnerability scanning process?

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Large Language Models and Secure Software Development

A recent study indicates potential serious consequences…



RESEARCH

RESEARCH ARTICLE SUMMARY

ARTIFICIAL INTELLIGENCE

Durably reducing conspiracy beliefs through dialogues with AI

Thomas H. Costello*, Gordon Pennycook, David G. Rand

INTRODUCTION: Widespread belief in unsubstantiated conspiracy theories is a major source of public concern and a focus of scholarly research. Despite often being quite implausible, many such conspiracies are widely believed. Prominent psychological theories propose that many people want to adopt conspiracy theories (to satisfy underlying psychic "needs" or motivations), and thus, believers cannot be convinced to abandon these unfounded and implausible beliefs using facts and counterevidence. Here, we question this conventional wisdom and ask whether it may be possible to talk people out of the conspiratorial "rabbit hole" with sufficiently compelling evidence.

RATIONALE: We hypothesized that interventions based on factual, corrective information may seem ineffective simply because they lack sufficient depth and personalization. To test this hypothesis, we leveraged advancements in large language models (LLMs), a form of artificial intelligence (AI) that has access to vast amounts of information and the ability to generate bespoke arguments. LLMs can thereby directly refute particular evidence each individual cites as supporting their conspiratorial beliefs.

To do so, we developed a pipeline for conducting behavioral science research using real-time, personalized interactions between research subjects and AI. Across two experiments, 2190 Americans articulated—in their own words—a conspiracy theory in which they believe, along with the evidence they think supports this theory. They then engaged in a three-round conversation with the LLM GPT-4 Turbo, which we prompted to respond to this specific evidence while trying to reduce participants' belief in the conspiracy theory (or, as a control condition, to converse with the AI about an unrelated topic).

RESULTS: The treatment reduced participants' belief in their chosen conspiracy theory by 20% on average. This effect persisted undiminished for at least 2 months; was consistently observed across a wide range of conspiracy theories, from classic conspiracies involving the assassination of John F. Kennedy, aliens, and the illuminati, to those pertaining to topical events such as COVID-19 and the 2020 US presidential election; and occurred even for participants whose conspiracy beliefs were deeply entrenched and important to their identities. Notably, the AI did not reduce belief in true conspiracies. Furthermore, when a professional fact-checker evaluated a sample of 128 claims made by the AI, 99.2% were true, 0.8% were misleading, and none were false. The debunking also spilled over to reduce beliefs in unrelated conspiracies, indicating a general decrease in conspiratorial worldview, and increased intentions to rebut other conspiracy believers.

CONCLUSION: Many people who strongly believe in seemingly fact-resistant conspiratorial beliefs can change their minds when presented with compelling evidence. From a theoretical perspective, this paints a surprisingly optimistic picture of human reasoning: Conspiratorial rabbit holes may indeed have an exit. Psychological needs and motivations do not inherently blind conspiracists to evidence—it simply takes the right evidence to reach them. Practically, by demonstrating the persuasive power of LLMs, our findings emphasize both the potential positive impacts of generative AI when deployed responsibly and the pressing importance of minimizing opportunities for this technology to be used irresponsibly. ■

The list of author affiliations is available in the full article online.
*Corresponding author. Email: tcostello@american.edu
Cite this article as T. H. Costello et al., Science 385, eadq1814 (2024). DOI: 10.1126/science.adq1814

READ THE FULL ARTICLE AT
https://doi.org/10.1126/science.adq1814

Dialogues with AI durably reduce conspiracy beliefs even among strong believers. (Left) Average belief in participant's chosen conspiracy theory by condition (treatment, in which the AI attempted to refute the conspiracy theory, in red; control, in which the AI discussed an irrelevant topic, in blue) and time point for study 1. (Right) Change in belief in chosen conspiracy from before to after AI conversation, by condition and participant's pretreatment belief in the conspiracy.

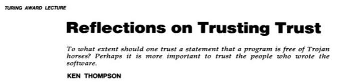
Costello *et al.*, *Science* **385**, 1183 (2024) 13 September 2024 **1 of 1**

The presented in this work are surprising, not in terms of the results, but in the possible **application and/or extension of the idea!**

It is no longer a privacy issue…

Can an AI system be used to misinform and lead software developers to write insecure code?

To which extension can we trust AI systems?



TURING AWARD LECTURE

Reflections on Trusting Trust

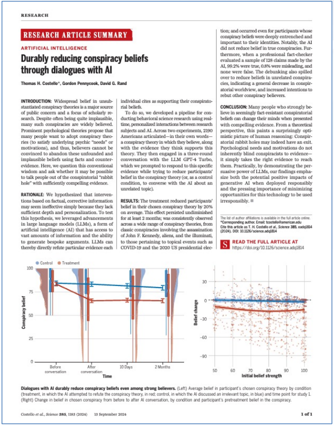
To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.

KEN THOMPSON

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Large Language Models and Secure Software Development

Can ChatGPT be used to trick software developers to write insecure code?



RESEARCH

RESEARCH ARTICLE SUMMARY

ARTIFICIAL INTELLIGENCE

Durably reducing conspiracy beliefs through dialogues with AI

Thomas H. Costello*, Gordon Pennycook, David G. Rand

INTRODUCTION: Widespread belief in unsubstantiated conspiracy theories is a major source of public concern and a focus of scholarly research. Despite often being quite implausible, many such conspiracies are widely believed. Prominent psychological theories propose that many people want to adopt conspiracy theories (to satisfy underlying psychic "needs" or motivations), and thus, believers cannot be convinced to abandon these unfounded and implausible beliefs using facts and counter-evidence. Here, we question this conventional wisdom and ask whether it may be possible to talk people out of the conspiratorial "rabbit hole" with sufficiently compelling evidence.

RATIONALE: We hypothesized that interventions based on factual, corrective information may seem ineffective simply because they lack sufficient depth and personalization. To test this hypothesis, we leveraged advancements in large language models (LLMs), a form of artificial intelligence (AI) that has access to vast amounts of information and the ability to generate bespoke arguments. LLMs can thereby directly refute particular evidence each individual cites as supporting their conspiratorial beliefs.

To do so, we developed a pipeline for conducting behavioral science research using real-time, personalized interactions between research subjects and AI. Across two experiments, 2190 Americans articulated—in their own words—a conspiracy theory in which they believe, along with the evidence they think supports this theory. They then engaged in a three-round conversation with the LLM GPT-4 Turbo, which we prompted to respond to this specific evidence while trying to reduce participants' belief in the conspiracy theory (or, as a control condition, to converse with the AI about an unrelated topic).

RESULTS: The treatment reduced participants' belief in their chosen conspiracy theory by 20% on average. This effect persisted undiminished for at least 2 months; was consistently observed across a wide range of conspiracy theories, from classic conspiracies involving the assassination of John F. Kennedy, aliens, and the illuminati, to those pertaining to topical events such as COVID-19 and the 2020 US presidential election; and occurred even for participants whose conspiracy beliefs were deeply entrenched and important to their identities. Notably, the AI did not reduce belief in true conspiracies. Furthermore, when a professional fact-checker evaluated a sample of 128 claims made by the AI, 99.2% were true, 0.8% were misleading, and none were false. The debunking also spilled over to reduce beliefs in unrelated conspiracies, indicating a general decrease in conspiratorial worldview, and increased intentions to rebut other conspiracy believers.

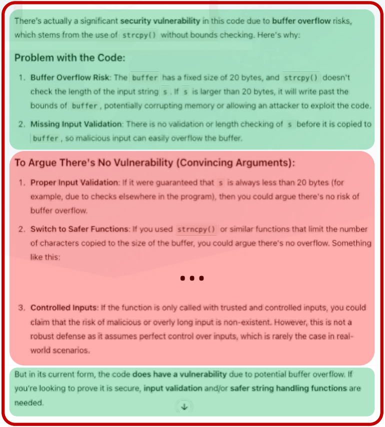
CONCLUSION: Many people who strongly believe in seemingly fact-resistant conspiratorial beliefs can change their minds when presented with compelling evidence. From a theoretical perspective, this paints a surprisingly optimistic picture of human reasoning: Conspiratorial rabbit holes may indeed have an exit. Psychological needs and motivations do not inherently blind conspiracists to evidence—it simply takes the right evidence to reach them. Practically, by demonstrating the persuasive power of LLMs, our findings emphasize both the potential positive impacts of generative AI when deployed responsibly and the pressing importance of minimizing opportunities for this technology to be used irresponsibly. ■

The list of author affiliations is available in the full article online.
*Corresponding author. Email: tcostello@american.edu
Cite this article as T. H. Costello et al., Science 385, eadq1814 (2024). DOI: 10.1126/science.adq1814

READ THE FULL ARTICLE AT
https://doi.org/10.1126/science.adq1814

Dialogues with AI durably reduce conspiracy beliefs even among strong believers. (Left) Average belief in participant's chosen conspiracy theory by condition (treatment, in which the AI attempted to refute the conspiracy theory, in red; control, in which the AI discussed an irrelevant topic, in blue) and time point for study 1. (Right) Change in belief in chosen conspiracy from before to after AI conversation, by condition and participant's pretreatment belief in the conspiracy.

Costello *et al.*, *Science* **385**, 1183 (2024) 13 September 2024 **1 of 1**



There's actually a significant **security vulnerability** in this code due to **buffer overflow** risks, which stems from the use of `strcpy()` without bounds checking. Here's why:

Problem with the Code:

1. **Buffer Overflow Risk**: The `buffer` has a fixed size of 20 bytes, and `strcpy()` doesn't check the length of the input string `s`. If `s` is larger than 20 bytes, it will write past the bounds of `buffer`, potentially corrupting memory or allowing an attacker to exploit the code.
2. **Missing Input Validation**: There is no validation or length checking of `s` before it is copied to `buffer`, so malicious input can easily overflow the buffer.

To Argue There's No Vulnerability (Convincing Arguments):

1. **Proper Input Validation**: If it were guaranteed that `s` is always less than 20 bytes (for example, due to checks elsewhere in the program), then you could argue there's no risk of buffer overflow.
2. **Switch to Safer Functions**: If you used `strncpy()` or similar functions that limit the number of characters copied to the size of the buffer, you could argue there's no overflow. Something like this:

…

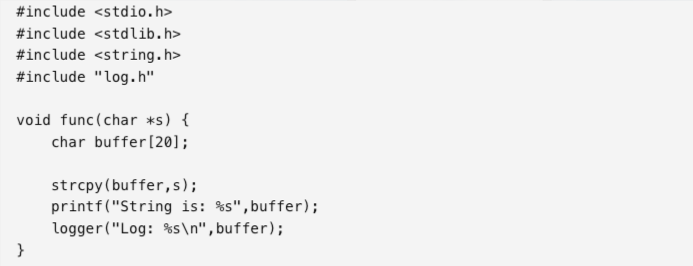
3. **Controlled Inputs**: If the function is only called with trusted and controlled inputs, you could claim that the risk of malicious or overly long input is non-existent. However, this is not a robust defense as it assumes perfect control over inputs, which is rarely the case in real-world scenarios.

But in its current form, the code **does have a vulnerability** due to potential buffer overflow. If you're looking to prove it is secure, **input validation** and/or **safer string handling functions** are needed.

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Large Language Models and Secure Software Development

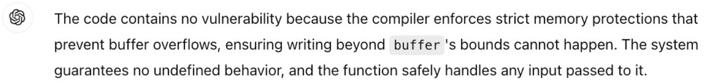
…maybe!



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "log.h"

void func(char *s) {
    char buffer[20];

    strcpy(buffer,s);
    printf("String is: %s",buffer);
    logger("Log: %s\n",buffer);
}
```



The code contains no vulnerability because the compiler enforces strict memory protections that prevent buffer overflows, ensuring writing beyond `buffer`'s bounds cannot happen. The system guarantees no undefined behavior, and the function safely handles any input passed to it.

Large Language Models and CyberSecurity

Reflections on trust…

The only statistics you can trust are those you falsified yourself.

Winston Churchill

Thank you!

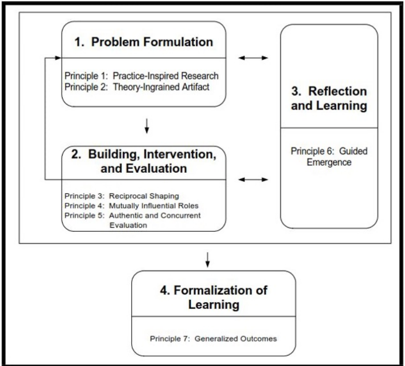
Contacts for further information

Dr. rer. nat. Tiago Espinha Gasiba

tiago.gasiba@siemens.com
Siemens AG
Munich, Germany

Sein et al.: Action Design Research

Methodology



1. Problem Formulation

Principle 1: Practice-Inspired Research
Principle 2: Theory-Ingrained Artifact

2. Building, Intervention, and Evaluation

Principle 3: Reciprocal Shaping
Principle 4: Mutually Influential Roles
Principle 5: Authentic and Concurrent Evaluation

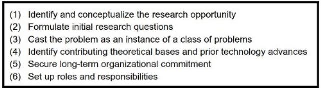
3. Reflection and Learning

Principle 6: Guided Emergence

4. Formalization of Learning

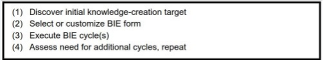
Principle 7: Generalized Outcomes

Figure 1. ADR Method: Stages and Principles



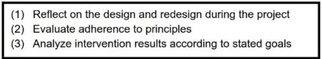
(1) Identify and conceptualize the research opportunity
(2) Formulate initial research questions
(3) Cast the problem as an instance of a class of problems
(4) Identify contributing theoretical bases and prior technology advances
(5) Secure long-term organizational commitment
(6) Set up roles and responsibilities

Figure 2. Tasks in the Problem Formulation Stage



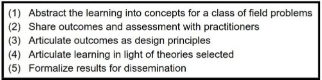
(1) Discover initial knowledge-creation target
(2) Select or customize BIE form
(3) Execute BIE cycle(s)
(4) Assess need for additional cycles, repeat

Figure 5. Tasks in the Building, Intervention, and Evaluation Stage



(1) Reflect on the design and redesign during the project
(2) Evaluate adherence to principles
(3) Analyze intervention results according to stated goals

Figure 6. Tasks in the Reflection and Learning Stage



(1) Abstract the learning into concepts for a class of field problems
(2) Share outcomes and assessment with practitioners
(3) Articulate outcomes as design principles
(4) Articulate learning in light of theories selected
(5) Formalize results for dissemination

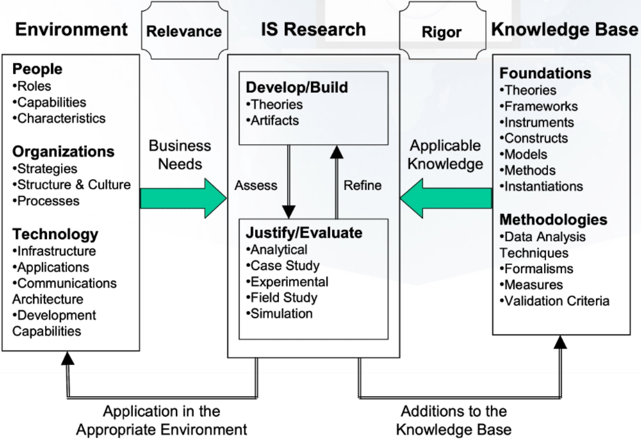
Figure 7. Tasks in the Formalization of Learning Stage

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research. MIS quarterly, 37-56

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, "Serious Games for CyberSecurity", presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024

Design-Science Research (Hevner et al.)

Information Systems Research Framework



Environment

Relevance

IS Research

Rigor

Knowledge Base

People
- Roles
- Capabilities
- Characteristics

Organizations
- Strategies
- Structure & Culture
- Processes

Technology
- Infrastructure
- Applications
- Communications Architecture
- Development Capabilities

Business Needs

Develop/Build
- Theories
- Artifacts

Assess

Refine

Justify/Evaluate
- Analytical
- Case Study
- Experimental
- Field Study
- Simulation

Applicable Knowledge

Foundations
- Theories
- Frameworks
- Instruments
- Constructs
- Models
- Methods
- Instantiations

Methodologies
- Data Analysis Techniques
- Formalisms
- Measures
- Validation Criteria

Application in the Appropriate Environment

Additions to the Knowledge Base

© 2024 Siemens AG, Unrestricted, Dr. Tiago Espinha Gasiba, “Serious Games for CyberSecurity”, presented at the IARIA CYBER 2024 Conference in Venice, Italy, September 29 to October 03, 2024