CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Video Game Development from the Perspective of Security Research

GermanDevDays | Juni 2023

TeamUSEC
Human-Centered Security

Introduction

Empirical & Behavioral Security Group (Hannover)

Philip Klostermeyer

Project Lead
PhD Researcher

Sabrina Amft

PhD Researcher

Alexander Krause

PhD Researcher

Prof. Dr. Sascha Fahl

Tenured Faculty @ CISPA
Full Professor @ LUH

TeamUSEC
Human-Centered Security

https://teamusec.de

CISPA Helmholtz Center for Information Security

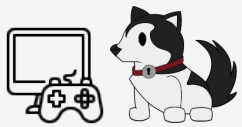
- Founded 2011 in Saarbrücken, joined Helmholtz 2019
- Basic and applied research in the field of cybersecurity and privacy
- gGmbH of the federal German state (90%) and the Saarland (10%)

- 6 research areas
- 34 faculties
- 300 employees

Security and Usability

General: Why is security so hard?

- **Functionality**
  - If user does *<some expected input>*, then system does *<some expected action>*

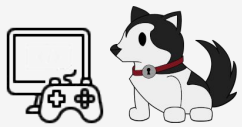


General: Why is security so hard?

- Functionality
  - If user does *<some expected input>*, then system does *<some expected action>*
- **Security**
  - If a user or outsider does *<some unexpected thing>*, then the system does not do *<any real bad thing>*

General: Why is security so hard?

- Functionality
  - If user does *<some expected input>*, then system does *<some expected action>*
- Security
  - If a user or outsider does *<some unexpected thing>*, then the system does not do *<any real bad thing>*
- **Why is security difficult?**
  - What are all the possible unexpected things?
  - How do we know that all of them are protected?
  - At what level of system abstraction?
    - software, hardware, crypto, user, …

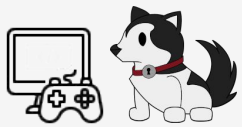


General: Why is security so hard?

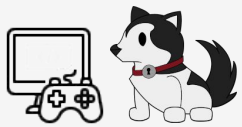
- Functionality
  - If user does *<some expected input>*, then system does *<some expected action>*
- Security
  - If a user or outsider does *<some unexpected thing>*, then the system does not do *<any real bad thing>*
- Why is security difficult?
  - What are all the possible unexpected things?
  - How do we know that all of them are protected?
  - At what level of system abstraction?
    - software, hardware, crypto, user, …

Security needs to protect against *everything*

But: Attackers only need to find one vulnerability

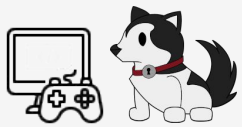


Security is a secondary task



> 95% of all security incidents recognize “human error” as a contributing factor*

- Admins: system misconfigurations, poor patch management
- Developers: security APIs misuse, outdated libraries
- End-users: easy-to-guess passwords, noncompliance to security warnings



*IBM Security Services Cyber Security Intelligence Index

Developers life is unnecessarily complicated

- Unsafe API Defaults

```
SecretKeySpec localSecretKeySpec = new SecretKeySpec(arrayOfByte, "AES");
Cipher localCipher = Cipher.getInstance("AES");
```

- Bad Documentation

```
Cipher c = Cipher.getInstance("DES/CBC/PKCS5Padding");
```

Using modes such as CFB and OFB, block ciphers can encrypt data in units smaller than
you may optionally specify the number of bits to be processed at a time by appending t

- Bad Tool Support

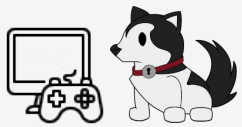
```
@Override
public void checkServerTrusted(X509Certificate[] arg0, String arg1)
        throws CertificateException {
    // TODO Auto-generated method stub
}
```

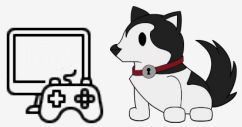
- Even the Bad Guys Fail

Security 31

Insane blackhats behind world's most expensive ransomware 'forget' to backup crypto keys

Only Linux victims can decrypt warped $247,000 BlackEnergy module - and then only maybe





Security in Video Games

*"[...] Whether it's **DDoS-for-hire botnets**, **credential abuse**, or **application attacks**, the gaming industry is popular in the worst of ways: **as the target**."*

State of the Internet Security | Web Attacks and Gaming Abuse (Volume 5, Issue 3) | Akamai, 2019

Who of you has implemented …

- Multiplayer / Client-Server Communication
  - Anti-Cheating
- Anti-Piracy / DRM
- Self-developed / Integrated APIs for
  - Login / Authentication
  - Payment
- Cryptography

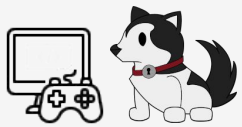
Who of you has implemented …

- Multiplayer / Client-Server Communication
  - Anti-Cheating
- Anti-Piracy / DRM
- Self-developed / Integrated APIs for
  - Login / Authentication
  - Payment
- Cryptography

- **Any Incidents?**

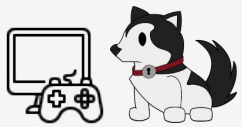
Security as a Feature: Constraints

- Time Issues
- Money Issues
- Knowledge Gap
- At odds with gameplay



Related Work

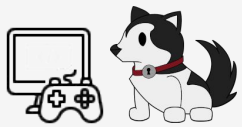
- *Cowboys, ankle sprains, and keepers of quality* (2014), E. Murphy-Hill
  - How is video game development different from software development?
  - Mixed-method study: Interviews + Surveys



Source: https://doi.org/10.1145/2568225.2568226

Related Work

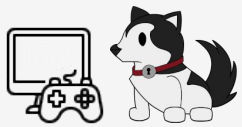
- *Cowboys, ankle sprains, and keepers of quality* (2014), E. Murphy-Hill
  - How is video game development different from software development?
  - Mixed-method study: Interviews + Surveys
    - **Vague requirements** + **complex architecture** leads to bug-prone software



Source: https://doi.org/10.1145/2568225.2568226

Related Work

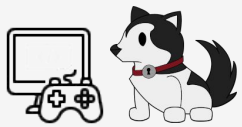
- *Cowboys, ankle sprains, and keepers of quality* (2014), E. Murphy-Hill
  - How is video game development different from software development?
  - Mixed-method study: Interviews + Surveys
    - **Vague requirements** + **complex architecture** leads to bug-prone software
    - Rapidly **changing code base hinders testing** and automation



Source: https://doi.org/10.1145/2568225.2568226

Related Work

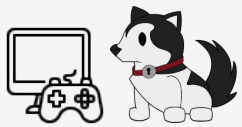
- *Cowboys, ankle sprains, and keepers of quality* (2014), E. Murphy-Hill
  - How is video game development different from software development?
  - Mixed-method study: Interviews + Surveys
    - **Vague requirements** + **complex architecture** leads to bug-prone software
    - Rapidly **changing code base hinders testing** and automation
    - **Complex tool pipelines**



Source: https://doi.org/10.1145/2568225.2568226

Related Work

- *Cowboys, ankle sprains, and keepers of quality* (2014), E. Murphy-Hill
  - How is video game development different from software development?
  - Mixed-method study: Interviews + Surveys
    - **Vague requirements** + **complex architecture** leads to bug-prone software
    - Rapidly **changing code base hinders testing** and automation
    - **Complex tool pipelines**
    - **Tight deadlines** lead to rushed development



Source: https://doi.org/10.1145/2568225.2568226

Incidents

The Diablo 4 beta has been bricking graphics cards

A repeat of the New

Malicious Dota 2 game modes infected players with malware

Apex Legends cheaters are removing players from lobbies at will

Fans Freak Out As *Zelda: Tears Of The Kingdom* Leaks Two Weeks Early

Some people are already playing the *Breath of the Wild* sequel

Hackers abuse Genshin Impact anti-cheat system to disable antivirus

Ubisoft Hacked And Private User Data Posted

CVE-2021-30481: Source engine remote code execution via game invites

Dark Souls servers taken down due to an exploit 'that could let someone take over your PC'

Zero-day in EA's Origin exposes gamers to yet more RCE pwnage

A Directory Traversal Attack on Punkbuster Server can be Leveraged to Gain Remote Code Execution

Grand Theft Auto 6 leak: who hacked Rockstar and what was stolen?

Battle.net has recovered from DDoS attack, Blizzard says

CD Projekt Red says it was hacked but won't pay the ransom

Ubisoft confirms Just Dance data breach amid developer exodus

XSS slip-up exposed Fortnite gamers to account hijack

Attack Vectors and Consequences

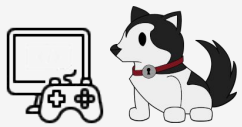
- **User issues** (e.g., *Clients, Game, Engine, Services*)
  - Engagement/Trust
  - Account issues
  - Monetary damages

Attack Vectors and Consequences

- User issues (Clients, Game, Engine, Services)
  - Engagement/Trust
  - Account issues
  - Monetary damages



- **Studio, Publisher, Games(-as-a-Service)** (e.g., *Infrastructure, Other Channels*)
  - Availability
  - Public Relations
  - Revenue

Attack Vectors and Consequences

- User issues (Clients, Game, Engine, Services)
  - Engagement/Trust
  - Account issues
  - Monetary damages





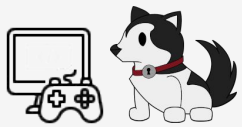


- Studio, Publisher, Games(-as-a-Service) (Infrastructure, Other Channels)
  - Availability
  - Public Relations
  - Revenue




- **Data Protection / Law**

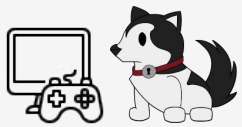




Security in Development

World of CVEs

- **C**ommon **V**ulnerabilities and **E**xposures
  - Public disclosure of security vulnerabilities



Source: https://cve.mitre.org/

Switch to https://
Home
Browse :
Vendors
Products
Vulnerabilities By Date
Vulnerabilities By Type
Reports :
CVSS Score Report
CVSS Score Distribution
Search :
Vendor Search
Product Search
Version Search
Vulnerability Search
By Microsoft References
Top 50 :
Vendors
Vendor Cvss Scores
Products
Product Cvss Scores
Versions
Other :
Microsoft Bulletins
Bugtraq Entries
CWE Definitions
About & Contact
Feedback
CVE Help
FAQ
Articles
External Links :
NVD Website
CWE Web Site
View CVE :
Go
(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)
View BID :
Go
(e.g.: 12345)
Search By Microsoft Reference ID:
Go

Vulnerability Details : CVE-2009-4768

Unspecified vulnerability in the JASS script interpreter in Warcraft III: The Frozen Throne 1.24b and earlier allows user-assisted remote attackers to execute arbitrary code via a crafted custom map. NOTE: some of these details are obtained from third party information.

Publish Date : 2010-04-20 Last Update Date : 2017-08-17

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

– CVSS Scores & Vulnerability Types

CVSS Score	9.3
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satistified to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Execute Code
CWE ID	94

– Products Affected By CVE-2009-4768

#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	Application	Blizzard	Warcraft 3 The Frozen Throne	*	*	*	*	Version Details Vulnerabilities
2	Application	Blizzard	Warcraft 3 The Frozen Throne	1.2.4a	*	*	*	Version Details Vulnerabilities
3	Application	Blizzard	Warcraft 3 The Frozen Throne	1.2.4	*	*	*	Version Details Vulnerabilities

– Number Of Affected Versions By Product

Vendor	Product	Vulnerable Versions
Blizzard	Warcraft 3 The Frozen Throne	3

– References For CVE-2009-4768

https://exchange.xforce.ibmcloud.com/vulnerabilities/54324
XF warcraft3-frozenthrone-jass-code-execution(54324)

http://forums.battle.net/thread.html?topicId=16888549346 CONFIRM

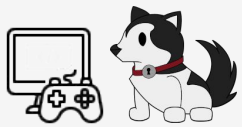
http://www.securityfocus.com/bid/37052
BID 37052 Warcraft III: The Frozen Throne JASS Interpreter Multiple Remote Code Execution Vulnerabilities *Release Date:*2009-11-18

http://secunia.com/advisories/37390

World of CVEs

- Common Vulnerabilities and Exposures
  - Public disclosure of security vulnerabilities

- **Standardised description, scoring, and references**
  - Easy to see if, e.g., your lib dependency has a flaw
  - Lessons learned what went wrong



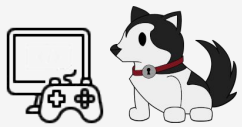
Source: https://cve.mitre.org/

World of CVEs

- Common Vulnerabilities and Exposures
  - Public disclosure of security vulnerabilities

- Standardised description, scoring, and references
  - Easy to see if, e.g., your lib dependency has a flaw
  - Lessons learned what went wrong

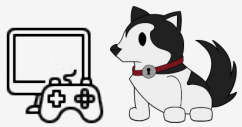
- **You can report that too!**



Source: https://cve.mitre.org/

World of CVEs

- CVE-2022-24125, CVE-2022-24126: FromSoftware matchmaking servers allow arbitrary push requests to clients
- CVE-2020-36603: Genshin Impact anti-cheat driver does not adequately restrict unprivileged function calls
- CVE-2021-30481: Source engine remote code execution via game invites
- CVE-2020-27708: Origin Client elevation of privileges
- CVE-2020-27383: Battle.Net elevation of privileges
- CVE-2019-14737: Ubisoft Uplay 92.0.0.6280 insecure permissions
- CVE-2015-9288: Unity Web Player allows accessing to online services via a victim's credentials
- [....]

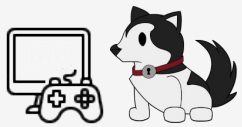


Source: https://cve.mitre.org/

World of CVEs

- **CVE-2022-24125**, **CVE-2022-24126**: FromSoftware matchmaking servers allow arbitrary push requests to clients
- **CVE-2020-36603**: Genshin Impact anti-cheat driver does not adequately restrict unprivileged function calls
- CVE-2021-30481: Source engine remote code execution via game invites
- CVE-2020-27708: Origin Client elevation of privileges
- CVE-2020-27383: Battle.Net elevation of privileges
- CVE-2019-14737: Ubisoft Uplay 92.0.0.6280 insecure permissions
- CVE-2015-9288: Unity Web Player allows accessing to online services via a victim's credentials
- [....]

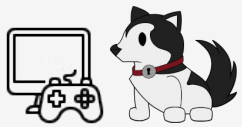
+ **CVE-2021-38003** Google's V8



Source: https://cve.mitre.org/

Example: Dark Souls (CVE-2022-24125, CVE-2022-24126)

- Third-person action role-playing (FromSoftware, 2011, 2014, 2016)

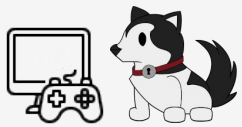


Source: https://github.com/tremwil/ds3-nrssr-rce

Example: Dark Souls (CVE-2022-24125, CVE-2022-24126)

- Third-person action role-playing (FromSoftware, 2011, 2014, 2016)

- Multiplayer: "Summon" other players for PvP/PvE



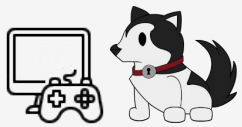
Source: https://github.com/tremwil/ds3-nrssr-rce

Example: Dark Souls (CVE-2022-24125, CVE-2022-24126)

- Third-person action role-playing (FromSoftware, 2011, 2014, 2016)

- Multiplayer: "Summon" other players for PvP/PvE

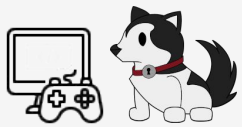
- Matchmaking via push requests and player IDs
- Send arbitrary push messages/payload
- Improper bounds checking, stack overflow to execute planted code



Source: https://github.com/tremwil/ds3-nrssr-rce

Example: Genshin Impact (CVE-2020-36603)

- Multiplatform open-world online action RPG (miHoYo, 2020)

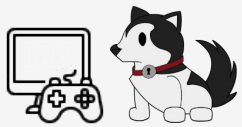


Source: https://github.com/kkent030315/evil-mhyprot-cli

Example: Genshin Impact (CVE-2020-36603)

- Multiplatform open-world online action RPG (miHoYo, 2020)

- Vulnerability in `mhyprot2.sys`, windows kernel-mode driver (Anti-Cheat with system-level privilege)



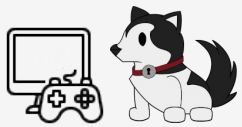
Source: https://github.com/kkent030315/evil-mhyprot-cli

Example: Genshin Impact (CVE-2020-36603)

- Multiplatform open-world online action RPG (miHoYo, 2020)

- Vulnerability in `mhyprot2.sys`, windows kernel-mode driver (Anti-Cheat with system-level privilege)

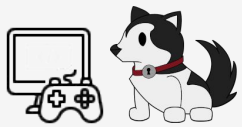
- Exposing high-privileged IOCTL (Input/Output Control) calls to user-mode
  - Read/Write arbitrary kernel/process memory
  - Terminate arbitrary processes
  - …



Source: https://github.com/kkent030315/evil-mhyprot-cli

Example: DotA 2 (CVE-2021-38003, Google's V8)

- Multiplayer online battle arena (Valve Corporation, 2013)

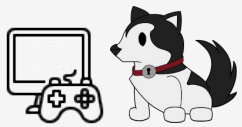


Source: https://decoded.avast.io/janvojtesek/dota-2-under-attack-how-a-v8-bug-was-exploited-in-the-game/

Example: DotA 2 (CVE-2021-38003, Google's V8)

- Multiplayer online battle arena (Valve Corporation, 2013)

- V8: Google's open-source JavaScript engine
- Embedded into DotA as `v8.dll`



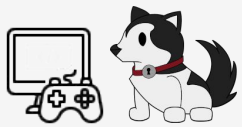
Source: https://decoded.avast.io/janvojtesek/dota-2-under-attack-how-a-v8-bug-was-exploited-in-the-game/

Example: DotA 2 (CVE-2021-38003, Google's V8)

- Multiplayer online battle arena (Valve Corporation, 2013)

- V8: Google's open-source JavaScript engine
- Embedded into DotA as `v8.dll`

- Exploit in outdated version included in game client
- Backdoor introduced through custom game mode (Steam Workshop)
- Execute arbitrary additional JavaScript code fetched via HTTP

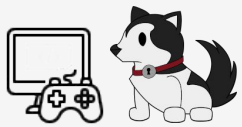


Source: https://decoded.avast.io/janvojtesek/dota-2-under-attack-how-a-v8-bug-was-exploited-in-the-game/

Security on Higher Levels

Related Work

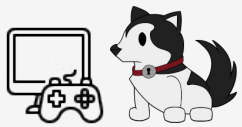
- *Security in Online Games: Current Implementations and Challenges* (2019), R. M. Parizi
  - Consider all components and interactions to **create security policies**
    - **Redundancies in policies** to avoid "weakest link" scenario



Source: https://doi.org/10.1007/978-3-030-10543-3_16

Related Work

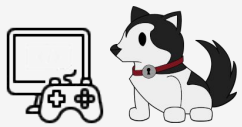
- *Security in Online Games: Current Implementations and Challenges* (2019), R. M. Parizi
  - Consider all components and interactions to **create security policies**
    - **Redundancies in policies** to avoid "weakest link" scenario
  - Thorough **testing** and **reviewing**, preferably automated, to ensure quality of product



Source: https://doi.org/10.1007/978-3-030-10543-3_16

Related Work

- *Security in Online Games: Current Implementations and Challenges* (2019), R. M. Parizi
  - Consider all components and interactions to **create security policies**
    - **Redundancies in policies** to avoid "weakest link" scenario
  - Thorough **testing** and **reviewing**, preferably automated, to ensure quality of product
  - **Developers must adapt** to changes in technology to strengthen security measures and protect, e.g., user data



Source: https://doi.org/10.1007/978-3-030-10543-3_16

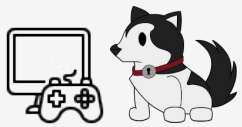
Related Work: Security in Development

- *Security in the Software Development Lifecycle* (2018), H. Assal
  - Interview study exploring security practices in development stages

Source: https://www.usenix.org/conference/soups2018/presentation/assal

Related Work: Security in Development

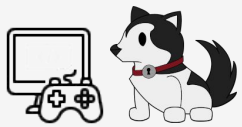
- *Security in the Software Development Lifecycle* (2018), H. Assal
  - Interview study exploring security practices in development stages
  - Factors affecting security practices:
    - Security knowledge, division of labor
    - **Company culture**, **availability of resources**
    - External pressure, security incidents



Source: https://www.usenix.org/conference/soups2018/presentation/assal

Related Work: Security Advocates

- *An Analysis of the Role of Situated Learning in Starting a Security Culture in a Software Company* (2021), A. Tuladhar
  - Ethnographic study embedding PhD student in company for 8 months

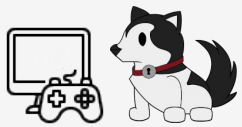


Source: https://www.usenix.org/conference/soups2021/presentation/tuladhar

Related Work: Security Advocates

- *An Analysis of the Role of Situated Learning in Starting a Security Culture in a Software Company* (2021), A. Tuladhar

  - Ethnographic study embedding PhD student in company for 8 months
  - Informing and **influencing teams regarding secure development** practices (e.g. knowledge transfer during discussions, code reviews)

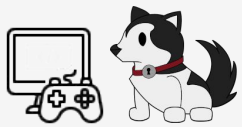


Source: https://www.usenix.org/conference/soups2021/presentation/tuladhar

Related Work: Security Advocates

- *An Analysis of the Role of Situated Learning in Starting a Security Culture in a Software Company* (2021), A. Tuladhar

  - Ethnographic study embedding PhD student in company for 8 months
  - Informing and **influencing teams regarding secure development** practices (e.g. knowledge transfer during discussions, code reviews)
  - **Management actively supporting** featured practices helps

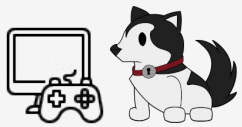


Source: https://www.usenix.org/conference/soups2021/presentation/tuladhar

Related Work: Security Advocates

- *An Analysis of the Role of Situated Learning in Starting a Security Culture in a Software Company* (2021), A. Tuladhar

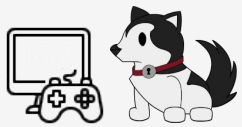
  - Ethnographic study embedding PhD student in company for 8 months
  - Informing and **influencing teams regarding secure development** practices (e.g. knowledge transfer during discussions, code reviews)
  - **Management actively supporting** featured practices helps
  - Positively **giving rise to security-aware software engineers**



Source: https://www.usenix.org/conference/soups2021/presentation/tuladhar

Applied Security Advices

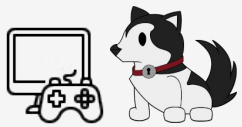
- Be **aware**
  - Even better: Have **policies**



Applied Security Advices

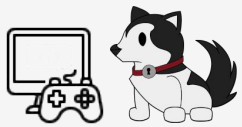
- Be **aware**
  - Even better: Have **policies**

- **Support devs** as manager



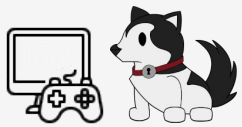
Applied Security Advices

- Be **aware**
  - Even better: Have **policies**
- **Support devs** as manager
- Do **tests** and **reviews**



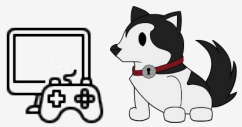
Applied Security Advices

- Be **aware**
  - Even better: Have **policies**
- **Support devs** as manager
- Do **tests** and **reviews**
- Have dedicated **people / resources**



Know your Enemy: Threat Modelling

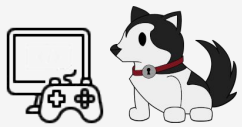
- Proactive: **Identify potential threats and vulnerabilities** in a system
  - Prioritize and allocate resources



Sources: https://owasp.org/www-community/Threat_Modeling

Know your Enemy: Threat Modelling

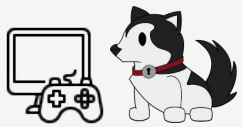
- Proactive: Identify potential threats and vulnerabilities in a system
  - Prioritize and allocate resources
- Enhanced system resilience, **reduce potential for breaches**
  - Foster security-conscious mindset in teams
  - Identify and address concerns early, reduce cost and effort for remediation



Sources: https://owasp.org/www-community/Threat_Modeling

Know your Enemy: Threat Modelling

- Proactive: Identify potential threats and vulnerabilities in a system
  - Prioritize and allocate resources
- Enhanced system resilience, reduce potential for breaches
  - Foster security-conscious mindset in teams
  - Identify and address concerns early, reduce cost and effort for remediation
- **Steps** in threat modeling:
  - Identify assets and resources to be protected
  - Enumerate potential threats and vulnerabilities
  - Assess impact and likelihood of each threat
  - Prioritize and implement *appropriate* security measures

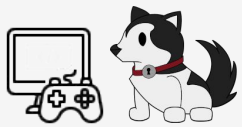


Sources: https://owasp.org/www-community/Threat_Modeling

Ongoing work and outlook

Case Study: P2P Security in Steam

- Master’s thesis in cooperation with Ruhr-Universität Bochum
- Recently released Sons of the Forest got cracked with multiplayer
- Crack using P2P system in Steam through specific developer API
- Investigating misuse potential of this loophole

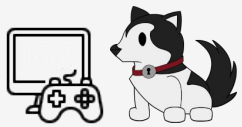


Interview Study: Security Challenges in Game Dev

- Why do security vulnerabilities sneak into video games in the first place?
  … and what can we do?
- Interviewing experts from the game development community
  - Involvement in **programming** components, **managing** teams, or **negotiating** contracts that address **security or privacy-related issues in or about a video game product**
- Develop tools, guidelines, and recommendations shared with the industry
- Enhancing the adoption of robust security practices in game development

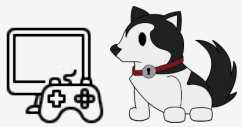
- **Let's build the baseline together!**

research.teamusec.de/2023-game-dev



Future Work

- Usage of unsafe programming practices in C#/C++
  (cf. *Interview Study on Use and Risks of Unsafe Rust* (2023), S. Höltervennhoff)[1]
- Tool development (Unity Engine/Unreal Engine/Unrelated)
  (cf. *Social influences on secure development tool adoption* (2014), S. Xiao)[2]
- Networking & fitting encryption
  (cf. *Evaluation of videogame network architecture perf. and security* (2021), B. Bryant)[3]
- Assets / Supply Chain
  - Security & Usability of proprietary internal (publisher) SDKs/Libs



Sources: [1] Publication in progress, [2] https://doi.org/10.1145/2531602.2531722, [3] https://doi.org/10.1016/j.comnet.2021.108128

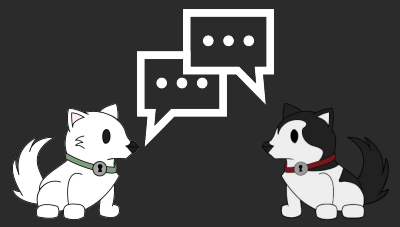
Q&A

Contact

Philip Klostermeyer
+49 681 87083 2099
philip.klostermeyer@cispa.de

linkedin.com/in/pklostermeyer

@pklosti



Participate in our Interview Study!

research.teamusec.de/2023-game-dev