Computers & Security 151 (2025) 104287

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full length article

Gamifying information security: Adversarial risk exploration for IT/OT infrastructures

Robert Luh [a,b,*], Sebastian Eresheim [a,b], Paul Tavolato [b], Thomas Petelin [b], Simon Gmeiner [a,b], Andreas Holzinger [c], Sebastian Schrittwieser [b]

[a] Department of Computer Science & Security, St. Pölten University of Applied Sciences, Campus Platz 1, St. Pölten, Austria
[b] Faculty of Computer Science, University of Vienna, Währinger Straße 29, Vienna, Austria
[c] Department of Forest- and Soil Sciences, BOKU University Vienna, Peter Jordan Straße 82, Vienna, Austria

ARTICLE INFO

Keywords:
Hacking
Security game
Model
Gamification

ABSTRACT

Today's interconnected IT and OT infrastructure faces an array of cyber threats from diverse actors with varying motivations and capabilities. The increasing complexity of exposed systems, coupled with adversaries' sophisticated technical arsenals, poses significant challenges for organizations seeking to defend against these attacks. Understanding the relationship between specific attack techniques and effective technical, organizational and human-centric mitigation measures remains elusive, as does grasping the underlying principles of information security and how they may be applied to cyber defense.

In response to these challenges, we propose a gamified metamodel that combines well-established frameworks, including MITRE ATT&CK, D3FEND, CAPEC, and the NIST SP 800-53 security standard. The programmatic implementation of the model, "PenQuest", combines elements of game theory with cybersecurity concepts to enhance risk assessment and training for IT practitioners and security engineers. In PenQuest, participants engage in a digital battle — attackers attempt to compromise an abstracted IT infrastructure, while defenders work to prevent or mitigate the threat. Bot opponents and the technical foundation for reinforcement learning enable future automated strategy inference.

This paper provides an in-depth exploration of the metamodel, the game's components and features built to translate cybersecurity principles into strategy game rules, and the technical implementation of a mature, ready-to-use education and risk exploration solution. Future work will focus on further improving the attack likelihood and detection chance algorithms for seamless risk assessment.

1. Introduction

Cyberattacks on IT and OT systems have become increasingly prevalent. Beyond their sheer volume, the economic impact of such threats continues to increase. The World Economic Forum's Global Cybersecurity Outlook 2024 (World Economic Forum, 2024) highlights major findings and emphasizes the widening cyber inequity and the profound impact of emerging technologies. In 2023, domain experts witnessed a series of major breaches of digital security, and this trend is expected to persist (Technology.org, 2024).

The economic toll of cyberattacks is increasingly high, with worldwide losses in the billions (Natalucci et al., 2024). Ransomware alone is expected to exceed $265 billion by 2031 (Cybersecurityventures.com, 2024). These figures, as well as the complexity of targeted information systems and the asymmetric nature of digital threats, underscore the urgency for organizations to understand the link between specific likely attacks and effective mitigating measures. Security education and risk assessment, as emphasized by most information security standards, remain two of the most critical components of effective defense strategies in this regard.

Serious games, which combine entertainment with educational objectives, have gained prominence in various domains. Rebah et al. (2019) emphasize the dual nature of serious games and define it as a 'computer application that combines a serious intention of educational, informative, and communicative nature with playful elements from video games.' In the context of IT security *risk assessment*, serious games offer several advantages (Rajbhandari and Snekkenes, 2018):

1. **Engagement and Learning**: Serious games engage participants actively, allowing them to learn and apply security concepts in a dynamic environment. Players experience risk scenarios

* Corresponding author at: Department of Computer Science & Security, St. Pölten University of Applied Sciences, Campus Platz 1, St. Pölten, Austria.
E-mail address: robert.luh@fhstp.ac.at (R. Luh).

https://doi.org/10.1016/j.cose.2024.104287
Received 3 October 2024; Received in revised form 9 December 2024; Accepted 16 December 2024
Available online 8 January 2025

firsthand, enhancing their understanding.
2. **Risk Scenario Exploration**: Games simulate various risk scenarios, enabling participants to explore different outcomes. This exploration helps identify potential risks and their consequences.
3. **Decision-Making Practice**: Participants make decisions within the game, similar to real-world risk assessment. These decisions impact the game's outcome, providing valuable practice for risk management.
4. **Quantitative Assessment**: Game-based models can incorporate quantitative data, such as probabilities and impact assessments. This enhances the accuracy of risk evaluation.

In terms of *education* alone, serious games have long been considered a viable approach (Abt, 1987). Research studies (Wilkinson, 2016) and security guidelines (Bundesamt für Sicherheit in der Informationstechnik (BSI), 2023) emphasize that such games not only serve as effective teaching tools, but also provide a platform to model information security principles across diverse audiences.

In addition to risk assessment and education, we argue that serious games can also serve as *planning tool* for penetration tests, for the assessment of defense measures in general, and as a means to *model cascading effects*: Certain real-world penetration tests, security controls, or awareness-raising measures could be implemented/updated to mirror the course of the game and address the identified shortcomings. The latter allows IT personnel to see how e.g. an outage of certain assets would impact the entire infrastructure.

In the context of these four areas of application, we introduce both an intricate *metamodel* for gamifiying information security as well as "PenQuest", a mature digital attack and defense *game implementation* based on said model. In PenQuest, an attacker aims to compromise an abstracted IT infrastructure, while the defender works to prevent or mitigate the damage. PenQuest functions as a virtual two-player board game, allowing users to engage with and dissect complex attacks without physically executing them. Most importantly, it empowers learners and risk managers to discover appropriate countermeasures at technical, organizational, and human levels.

The underlying model presented in this article has been fully reworked to accommodate gamified education and risk assessment in the digital domain while maintaining a high degree of realism. A very early version (sans implementation), which focuses more on game-theoretic aspects, has been published in 2020 (Luh et al., 2020). Another publication (Luh et al., 2022) focuses purely on the educational impact of utilizing an early game client in class, details of which are not reiterated here.

Specifically, this paper contributes by:

- assessing numerous security vocabularies, standards, and frameworks in regard to their suitability to creating gamified education and risk assessment solutions in the domain of cybersecurity;
- presenting a metamodel that combines said data sources, including different mappings and probability benchmarks in the context of IT/OT attacks and defense;
- discussing in depth the gamification aspects of said model;
- showcasing an advanced implementation of a serious game ("PenQuest"), which implements the model as part of a digital two-player board game.

The remainder of this paper is structured as follows. Section 2 provides related work in the area of gamified cybersecurity, with a focus on serious games. In Section 3, we discuss the metamodel in its entirety and explain its various data sources, components, and underlying algorithms. Section 4 revolves around the playable game that was built around the model and how it works, as well as educates and motivates the players. The technical implementation is discussed as well. In Section 5 we present a use case, discuss utilizing the model as a game, and list limitations and future enhancements of our approach. Lastly, Section 6 concludes the paper.

2. Related work

In existing literature, three distinct approaches to games within the security context have emerged:

1. Game-theoretic and formal modeling research: This approach focuses on game-theoretic aspects and formal models.
2. Serious games: These are used in awareness training and other contexts.
3. AI-enabled adversarial games: This emerging field utilizes machine learning to facilitate tasks such as malware detection or specific threat responses.

In this overview, we focus on the category of *serious games*, as it aligns most closely with our own approach. The differences to other models as well as synergies to other works are discussed after that.

2.1. Serious games

Several papers have surveyed the landscape of serious games in the context of information security: Batzos et al. (2023) highlight the importance of such games for awareness and incident response training in both the digital and physical domains. Prümmer et al. (2024) conducted a systematic literature review, offering a comprehensive overview of the methods used in cybersecurity training. Among these, game-based approaches were the most prevalent, representing 30% of the reviewed studies. While a significant portion of the literature, particularly on social engineering, is well-covered, relatively few works address a broader spectrum of cybersecurity topics — a gap that PenQuest aims to fill. Other research (Manos and Faragallah, 2019) focuses more on building educational games by combining learning contents with game mechanics. Here, the authors evaluate a number of games to determine which best serves as pedagogic tool. Ranked highest is "Space Fighter", a 3D shooter game created by the authors themselves, which teaches students about the difference between phishing and benign URLs.

Another paper, titled "Cybersecurity training for Web Applications through Serious Games" by Gaurav et al. (2021), investigates the use of serious games to enhance cybersecurity awareness and training related to web-based applications. While the paper focuses mainly on SQL injection, broken authentication, XSS, and information disclosure as defined in the OWASP Top 10,[1] it generally highlights the potential of serious games as a valuable tool for cybersecurity training.

Another notable serious game in the domain of IT security is "Elevation of Privilege" (Shostack, 2014). This physical print-at-home card game aims to assist individuals in understanding threat modeling and helps aspiring analysts identify general threats to their IT and software systems. The game is based on Microsoft's STRIDE mnemonic (Kohnfelder and Praerit, 1999; Swiderski and Snyder, 2004), which associates threats with desired security properties such as confidentiality or availability. Notably, "Elevation of Privilege" lacks mechanics for cyber-defense, and topological factors are not considered.

"Riskio" (Hart et al., 2020) also builds upon the STRIDE framework. Here, players engage in attack and defense gameplay facilitated by a human game master who assesses the success of actions based on player arguments. The game employs three distinct diagrams representing game boards: an office map, a network diagram, and a data flow diagram. The tabletop game aims to increase cybersecurity awareness for people with no technical background.

"SherLOCKED" (Jaffray et al., 2021) is a serious game designed for undergraduate computer science students to enhance their cybersecurity knowledge. The game includes scenarios that require players to identify and mitigate security risks, thereby reinforcing theoretical knowledge through practical application. Unlike PenQuest, SherLOCKED takes the shape of a quiz-heavy 2D puzzle game that has been

[1] https://owasp.org/www-project-top-ten/.

built solely for student education. The project clearly shows the benefit in using serious games in a modern learning environment.

"Operation Digital Chameleon" (Rieb and Lechner, 2016) takes the form of a red-team exercise in the shape of a board game. Players collaborate to construct attack and defense strategies for specific scenarios. However, similar to "Riskio", the proposed solutions are evaluated by a game master. While this workshop-style approach offers flexibility and suits large group events, it lacks a distinct security model or computational resolution methods. Other solutions, such as "Backdoors & Breaches" (Black Hills Information Security, 2024), combine physical game components with team-based decision-making. In this game, defenders aim to reveal the attacker's cards within a turn limit, following a kill chain similar to PenQuest. However, it requires human facilitation akin to a role-playing game and does not incorporate advanced mechanics like modeling different assets, interconnections, threat prevention, or specific compromise types.

In contrast, "OWASP Cornucopia" (OWASP Foundation, 2024) follows a different premise. This game provides cards to assist software development teams in identifying the security requirements of their projects by expediting discussion. The game links technology-agnostic concepts to exemplary weaknesses (CWE) and attack patterns (CAPEC).

Additionally, there are educational and commercial games designed to raise user awareness. Most of these games prioritize entertainment and do not delve into complex security models. For instance:

- "Keep Tradition Secure" (Texas A&M University, 2024), the US Department of Defense's Cyber Awareness Challenge (Defense Information Systems Agency, 2024), "Targeted Attack: The Game" (Trend Micro, 2024), and "The Weakest Link" (Decisions, 2024) can be considered quizzes or interactive decision-making games. Users must choose options that are least likely to lead to a compromise (e.g., avoiding publicizing certain information or clicking suspicious links) or identify potential security violations in a virtual room (LivingSecurity, 2024b).
- "CyberEscape Online" (LivingSecurity, 2024c) incorporates team-building aspects and includes several security best practices without focusing on technical backgrounds.
- Phishing-specific solutions like "Craft that Phish" (LivingSecurity, 2024a), "What.Hack" (Wen et al., 2017), and "Jigsaw" (Google Inc, 2024) explore specific threats and how to spot them. While these games address human and some organizational vulnerabilities, they do not provide comprehensive IT security education.

There are additional games, often intended for younger audiences, listed in Adam Shostack's repository (Shostack, 2024).

2.2. Security models

The primary difference between our abstraction model and existing frameworks lies in its focus on gamification and its comprehensive integration of attack and defense dynamics. While existing tools excel in specific domains, our model uniquely combines narrative-driven serious gaming with flexible threat modeling to address both educational and practical applications. This approach allows users to engage with cybersecurity concepts interactively, fostering learning and real-world applicability.

Unlike MulVAL (Tayouri et al., 2023), which primarily uses attack graphs to evaluate system vulnerabilities, our model integrates defense mechanisms, resource allocation, and cascading effects within a unified framework, creating a dynamic interplay between attackers and defenders. Similarly, CyberSage (Vu et al., 2014) focuses on quantitative risk assessment, whereas our model emphasizes flexibility in incorporating various data sources and abstraction levels, enabling both high-level scenario modeling and granular technical analysis.

PenQuest is first and foremost a serious game. However, it also provides a means to *plan penetration tests* as well as *model cascading effects* on IT infrastructures. In this context, the work by Rak et al. (2024) uses graphs to model target system infrastructures similar to our approach. CAPEC and ATT&CK tactics are used for modeling attacks and tools are taken directly from penetration testing distributions and other sources. While the defense side is not considered, the paper shows how PenQuest can become an asset in planning pentests with only minor adaptations.

In terms of modeling cascading effects on (critical) infrastructures, Grafenauer et al. (2018) demonstrate the usefulness of simulating the interdependencies between systems. Unlike PenQuest, where asset dependencies are modeled as a minor, custom-coded feature without the temporal component (i.e., outages currently impact linked assets immediately without delay), the authors' work uses OMNeT++[2] for modeling and visualizing the propagation of consequences.

Although reinforcement learning (RL) is outlined as future work in this paper, we are actively developing an RL environment for PenQuest, with early results showing promise. Tang et al. (2024) have explored RL-based decision-making for network attack-defense scenarios, implementing their approach within the Cyber Operations Research Gym (CybORG) (Standen et al., 2022). While their experiment, environment, and model differ significantly from ours, their work highlights the potential of RL in cybersecurity applications.

To the best of our knowledge, no existing work offers the level of scope, versatility and maturity that PenQuest does. At the same time, existing research provides many possible synergies; be that the incorporation of educational quizzes, a more web app centered variant, or penetration test planning using a less abstract, tool-enriched model.

3. Metamodel

In this section, we present an overview of the model created for the purpose of subsequent gamification. The goal was to create a flexible means to depict the many facets of IT/OT cyberattacks and the corresponding defense measures based on established industry standards with respect to data sources and general vocabulary. Generally, we define the following criteria for our gamified model:

- **Scope**: The model offers a wide information security scope without sacrificing depth. Specifically, we seek to include a narrative context, (threat) actors, organization assets and their dependencies, actions taken (i.e., actor behavior such as countermeasures implemented), realistic attack stages (kill chain), procurement decisions, as well as applied risk assessment in the form of impact and likelihood of success.
- **Flexibility**: The model is flexible in terms of content and utilization:
  - Data sources: The model can be populated by different data content taken from different data sources without having to be restructured or rebuilt. This means that users can either use our custom-built variant based on multiple sources or opt to incorporate established data sources such as MITRE D3FEND as-is, e.g., to shift the focus from organizational security to technical security.
  - Abstraction level: The model is flexible in terms of abstraction, allowing users to sculpt high-level scenarios as well as model highly technical settings close to the physical implementation of an attack or defense measure.
  - Utilization: Thanks to its design and the aforementioned data sources of various origin, the model has the potential to be utilized flexibly in various areas ranging from IT infrastructure security to physical security or application-specific (e.g., web app) security. Note that this paper focuses on the former.

[2] https://omnetpp.org/.

- **Gamification**: The model lends itself to gamification, meaning that it can be implemented as a physical or digital game app that can be used for education and risk assessment.

In the following, the most important data sources are discussed, and the components of the model are explained in detail.

3.1. Data sources

To satisfy our flexibility requirements, we have evaluated over 20 individual frameworks and knowledge bases on different levels of abstraction. In addition, several vocabularies were taken into consideration for later populating the model with actors and assets, as well as to establish narrative context. These data sources were narrowed down based on the following criteria:

- Abstraction: The data source uses abstracted information that does not directly correspond to the technical implementation or command(s) entered.
- Access: The data source is generally considered open access or at least provides sufficient information to the public for applied modeling.
- Content: The data source contains attack and/or defense techniques, patterns, behavior, tactics, methodologies, weaknesses, flaws, vulnerabilities, or controls and provides at least high-level information about mapping that data to their adversary counterpart.
- Industry standard: Data source is established in the industry and/or published by an organization considered an established vendor or provider.

We opted to exclude best-practice guides that are largely based on security standards, which, in turn, were included in the selection. Near-duplicates and national variants were omitted in favor of the original (i.e., more established) source – with the exception of ISO 27002 and NIST SP 800-53, where we retained both. We limited our result set to the top five for each domain, counting the two standards only once. This yielded a total of 10 frameworks with 20 individual components listed in Tables 1 and 2.

In the process of narrowing down our results, we assign an abstraction level to each source:

○○○: No abstraction. Data sources that do not abstract at all were summarily excluded from consideration. These include technical databases that provide specific commands for certain IT systems, like the Exploit Database[3] or various threat intelligence sources. While it is ultimately helpful to link these resources to our more abstract model on a use case basis (e.g., serving as an example), the initial version presented in this paper is independent of specific, short-lived technical attacks or the concrete technical implementation of a defense measure on e.g., a single workstation or web app.

●○○: Low abstraction. We consider data sources of "low abstraction" if they contain references to specific technical systems or include implementation suggestion in the form of e.g., pseudocode. These data sources were investigated for their suitability for inclusion.

●●○: Medium abstraction. Data sources that describe attacks and specific defensive controls independent from their technical implementation or specific software/hardware are considered to be of medium abstraction. This category is most relevant for our metamodel.

●●●: High abstraction. Data sources of this category provide high-level attack or defense categorization or taxonomies. For our model, we reviewed such sources for their usefulness in modeling attack stages (see Section 3.2.4), for attack–defense mapping purposes (Section 3.3), and as basis for computing success and detection probabilities (Section 3.4).

[3] https://www.exploit-db.com/.

Note that while we seek to retain the ability to use different data sources as-is with only minor adaptations, we also provide a ready-to-use solution that combines several data sources into one set of "actions" that are ultimately used within our game. This addresses the drawbacks of the various data sources and closes the gaps caused by some frameworks' incompleteness or audience-focused approach. In the following, we discuss the specific data sources considered in and incorporated into our model. These are frameworks and databases within the *attack* domain, the *defense* domain, as well as *generic* sources such as vocabularies and kill chain models.

3.1.1. Attack

Table 1 lists all data sources from low to high abstraction and provides references as well as short descriptions. Each data source by itself can be used to populate our model in the form of game actions. Our default variant of the populated model, which has been implemented in its entirety and is referenced in this paper, uses MITRE ATT&CK (MITRE Corporation, 2023d) techniques as well as CAPEC (MITRE Corporation, 2023a) meta and standard patterns as primary data sources. CVE (MITRE Corporation, 2023b) is used in concert with "Exploit" equipment discussed in Section 3.2.5. An OWASP (Open Web Application Security Project, 2023b) web application variant is currently in the works.

Going into detail, we argue that our choice is grounded in the fact that ATT&CK, CAPEC, and CVE offer highly relevant information that lends itself well to an IT/OT security game. These include, but are not limited to:

- Skill requirement: Level of skill needed to conduct the attack.
- Prerequisites: Requirements for the attack to be successful, e.g. valid credentials, administrator privileges, or physical access (attack vector).
- Likelihood and severity: Potential damage of the attack (CIA impact metrics (Stoneburner et al., 2002)) and its likelihood of occurrence.
- Attack phase: Tactic or attack stage the technique/pattern can be a part of.
- Platforms: Operating systems affected by the attack technique.

As mentioned above, most data sources are not complete or flaw-free when it comes to these variables. Since we built our game rules around this information, a lot of data had to be amended. Section 3.2.4 goes into more detail in regard to our game's actions for which the data sources serve as foundation.

3.1.2. Defense

Table 2 presents the list of defense domain frameworks, knowledge bases, and standards considered in our model. Like with the attack domain, every data source could be adopted to serve as basis for our game's actions if adapted accordingly.

The variant presented and evaluated in this paper uses MITRE D3FEND (MITRE Corporation, 2023f) techniques in combination with NIST SP 800-53 (MITRE Corporation, 2023h) controls to encompass both technical and organizational measures. D3FEND in particular offers valuable descriptions and considerations as well as the occasional link to its ATT&CK counterparts as part of its ontology. Unfortunately, neither framework offers information like skill prerequisites, mitigation effectiveness, or other values that would help to turn them into game actions. This information was added as part of the action modeling process discussed in Section 3.2.4.

The Cyber Analytics Repository (MITRE Corporation, 2023e) and Engage (MITRE Corporation, 2023g) mostly serve as means to evaluate and improve the mapping between attack and defense actions that were modeled after the attack and defense data sources. This mapping is discussed in Section 3.3. ASVS (Open Web Application Security Project, 2023a) will be key when implementing an OWASP web application variant.

Table 1
Overview of data sources associated with the attack domain. Abstraction level (Abstr.) ranges from ●○○ (specific to certain technical products, concrete information about technical implementation) to ●●● (high level overview or generic taxonomy).

Data source	Org.	Abstr.	Description
ATT&CK Tactics (MITRE Corporation, 2023d)	MITRE	●●●	Common stages of an attack (e.g., Reconnaissance), split into Enterprise, Mobile, and ICS.
ATT&CK Techniques	MITRE	●●○	Specific means of achieving an attack objective, associated to one or several Tactics.
ATT&CK Subtechniques	MITRE	●●○	Specific variant of a more general technique, such as a specific type of scan associated with "Active Scanning".
CAPEC Mechanisms (MITRE Corporation, 2023a)	MITRE	●●●	Overview of mechanisms used when exploiting a vulnerability such as "Subvert Access Control".
CAPEC Meta Pattern	MITRE	●●○	Abstraction of attack methodology or technique, e.g. "Content Spoofing".
CAPEC Standard Pattern	MITRE	●●○	More specific attack associated to a meta pattern, e.g. "Counterfeit GPS Signal".
CAPEC Detailed Pattern	MITRE	●○○	Attack pattern describing a specific execution flow that may reference a specific piece of technology.
CWE Class (MITRE Corporation, 2023c)	MITRE	●●●	Generic design weakness, e.g. "Improper Access Control"
CWE Base	MITRE	●●○	More specific weakness still mostly independent of a product, such as "Exposed Dangerous Method or Function".
CWE Variant	MITRE	●○○	Very specific weakness linked to a certain product, e.g. "Exposed Unsafe ActiveX Method", often in concert with a CVE example.
CVE (MITRE Corporation, 2023b)	MITRE	●○○	Database of product-specific vulnerabilities including various scores to measure impact and ease of exploitation.
OWASP Top 10 (Open Web Application Security Project, 2023b)	OWASP	●●○	List of web application flaws categorized into a top 10 list. Provides a link with CWE.

Table 2
Overview of data sources associated with the defense domain. Abstraction level (Abstr.) ranges from ●○○ (specific to certain technical products, concrete information about technical implementation) to ●●● (high level overview or generic taxonomy).

Data source	Org.	Abstr.	Description
D3FEND Tactics (MITRE Corporation, 2023f)	MITRE	●●●	Categories of cyber defense, e.g., including hardening, detection, and isolation.
D3FEND Technique	MITRE	●●○	Knowledge graph of technical countermeasures associated to one or several tactics as well as ATT&CK techniques.
Cyber Analytics Repository MITRE Corporation (2023e)	MITRE	●○○	Knowledge base providing a link between ATT&CK techniques and D3FEND countermeasures, as well as monitoring tool suggestions with pseudocode implementation examples.
Engage (MITRE Corporation, 2023g)	MITRE	●●○	Framework for planning and discussing the detection, prevention, and eliciting of adversary behavior. Provides a link to MITRE ATT&CK.
SP 800-53 Control Family (MITRE Corporation, 2023h)	NIST	●●●	High-level categories corresponding to cybersecurity controls such as "Awareness and Training" and "Access Control".
SP 800-53 Controls	NIST	●●○	Specific technical or organization-wide controls associated to a control family, e.g. "Concurrent Session Control".
ISO 27002 Control Type (International Organization for Standardization, 2023)	ISO	●●●	High-level categories separating types of controls such as "People Controls" and "Technological Controls".
ISO 27002 Controls	ISO	●●○	Specific technical or organization-wide controls associated to a control type, e.g. "Information Backup". Comparable to NIST SP 800-53 controls.
ASVS Control Category (Open Web Application Security Project, 2023a)	OWASP	●●●	High-level web app control categories like "Authentication" and "Session Management". Similar to NIST's control families.
ASVS Controls	OWASP	●●○	Controls associated with a category and mapped onto three levels of application security verification levels. Similar to NIST's controls, but for web apps.

3.1.3. Generic

Finally, we incorporated additional data sources that primarily include vocabularies that are part of the Structured Threat Information eXpression (STIX) language (OASIS Cyber Threat Intelligence Technical Committee, 2023), which describes cyber threat information for the purpose of sharing, storing, and analyzing information about attacks in a consistent manner. These vocabularies[4] include:

- `Industry Sector` vocabulary, including industries like "construction" and "education".
- `Threat Actor Type` vocabulary with values like "activist", "competitor", and "crime-syndicate". Note that our threat actors are of `Threat Actor Role` "agent" or "independent"; we do not model background roles such as "directors" or "sponsors".
- `Threat Actor Sophistication` describes the level of skill and resources an attacker has, ranging from "none" to "strategic".
- `Attack Motivation` vocabulary that contains attacker motivations such as "notoriety", and "personal-gain".

[4] https://docs.oasis-open.org/cti/stix/v2.1/cs01/stix-v2.1-cs01.html.

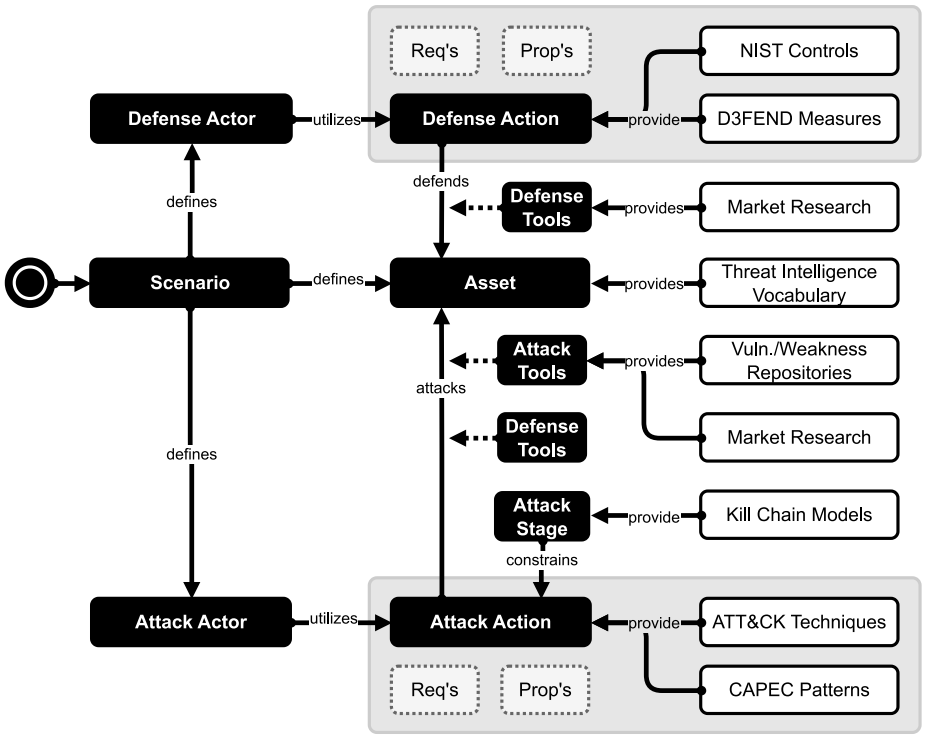

Fig. 1. Overview of the metamodel's components. Actions come with a set of requirements and properties as per the rules derived from our various data sources. Note that defense tools either support defense actions or hinder the use of attack actions (dotted arrows), mostly by altering success and detection probabilities.

We use these vocabularies to provide narrative context and as general source of information. Specifically, we modeled our game's actors and their motivation after STIX' vocabularies and loosely base our actor attributes off the `Sophistication` vocabulary. These actors are key components of our narrative scenarios that serve as storyline for individual games and play a key role in gamification. See Sections 3.2.1, 3.2.2, 3.2.4 and 4 for more information.

3.2. Components

This subsection summarizes the individual components of the model and how they work together in the context of the game. Fig. 1, which updates the version found in Luh et al. (2022), provides an overview.

In short, we model each cyberattack as a so-called *scenario* that provides the narrative framework and defines the participating *actors* (i.e. adversaries) discussed in Section 3.2.2, as well as an organization's *assets*. Assets are typically IT systems connected as part of an infrastructure that may be dependent of each other and may change state during a scenario by means of *actions*. Both attackers and defenders are provided a set of requirements-constrained actions derived from aforementioned data sources, which translate to specific attack techniques and defense measures. An asset's *attack stage* measures how far the attack has progressed and which actions can be used to proceed. *Tools* ranging from scripts and vulnerability scanners to various security appliances that support or hinder said actions and may weaken or protect assets.

In the following, each of the mentioned components is exemplified and explained in more detail.

3.2.1. Scenarios

Scenarios provide context, define the *mission* and specific *goals*, and set up actors and assets. Simply put, they define which attack story is to be explored. For example, a scenario might define the attacker as "skilled hacktivist who is out to go up against an evil food corp who allegedly abuses their labor force in third world countries" with the mission to "let the world know by manipulating their website, leaving damning evidence on their exposed web server".

The defender is similarly outlined and typically aims to keep their infrastructure secure against certain or all attacks.

In game terms, the scenario then quantifies the actors' attributes (see Section 3.2.2) and defines the game's scope through the number and type of assets modeled (refer to Section 3.2.3 for more on the topic), lists which of the assets are being targeted, and determines the repertoire of actions available to both sides. In our straightforward example, the attacker seeks to compromise the "Web Server" asset's integrity; the defender works to prevent just that.

Note that scenarios can have multiple goals that – in addition to said asset integrity compromise through data or configuration manipulation – may encompass data theft (loss of confidentiality), denial of service (loss of availability), or simply the accumulation of wealth or insight (see actor attributes in Section 3.2.2). Reaching these CIA goals (Stoneburner et al., 2002) not only sets up a game's victory conditions, it also directly links to the gamification aspects of *narration* and *motivation*, both of which are discussed in Section 4.

3.2.2. Actors

Actors are the protagonists in a modeled scenario. We use the aforementioned STIX `Threat Actor Type` vocabulary as a source for the nomenclature. Each attack and defense actor – let us use a generic "hacker" as an example – comes with a number of *attributes* that describe their properties:

- **Skill** [1..5]: Level of an actor's technical abilities. Skill constrains action use and influences success and detection chances (see Section 3.4). Practically speaking, a higher Skill rating unlocks more complex actions and makes it easier for actors to succeed in their task. A hacker with skill rating 3 can attack known and some unknown vulnerabilities and is proficient in various tools. However, they are not able to perform supply chain attacks or develop zero-day attacks.

- **Determination** [1..5]: Measure of actor motivation that determines the number of actions available. High Determination translates to more tactical choice in planning and executing an attack or implementing an effective defense. In addition, it influences how much time (i.e., Initiative, see below) an attacker is willing to invest into an undertaking before calling it quits. A hacker with a Determination attribute of 2 will only perform rudimentary research into new attack techniques and will abort their efforts after a relatively short time.
- **Wealth** [1..5]: Amount of financial and human resources available to the actor. More Wealth means more options when it comes to procuring tools or implementing policies. A hacker with a Wealth rating of 4 is able to invest significant resources into attack tools and may procure expensive exploits and malware on the black market.
- **Insight** [0..15]: Measure of knowledge gained about the opponent. Akin to Skill, Insight improves success and detection chances. Unlike Skill, Insight is accumulated during play and may even serve as a scenario's victory condition. An exemplary hacker with an accumulated Insight of 7 has learned a lot about their victim's MO and is able to adapt their attack to remain mostly undetected.
- **Initiative** [0..n] (attacker only): Maximum number of turns available to achieve the scenario's goal. This value represents the ticking clock and determines how much time and resources the attacker is willing to invest. Initiative is only reduced once the defender becomes aware of their opponents' endeavor (see Stealth and Conflict phase in Section 4.1). If Initiative reaches zero before the attacker meets their objective, the defense is considered too effective to beat, awarding victory to the defender. Skill, Determination as well as the number of assets in play contribute to initial Initiative.
- **Action Points** [0..$3 * n$]: Every point of Initiative (i.e., game turn) translates to 3 action points (AP) that can be invested into actions at will. Low-AP actions are quickly executed (e.g., a malicious command sent to a compromised server), while more expensive actions require a significant investment in regard to implementation time. See Section 3.2.4 for more information.

3.2.3. Assets

In our model, assets are tangible IT system (components) that actors attack and defend. They can be nested, vary in granularity, and are possibly dependent on each other. These dependencies and interconnections are the means by which the model captures complex attack patterns such as *cascading attacks*, *stepping-stone attacks*, and *multi-layered attacks*. Cascading attacks (Palleti et al., 2021) are addressed by modeling asset dependencies, where a compromise in one asset can propagate to others, reflecting the ripple effects seen in real-world scenarios. Stepping-stone attacks (Zhang and Paxson, 2000) are incorporated by defining access vectors and allowing attackers to exploit one compromised asset as a gateway to target others, enabling lateral movement. Multi-layered attacks (Hahn et al., 2015) are captured through (a) the nested structure of assets, which permits the modeling of progressive attacks that penetrate through layers of defense, and (b) through the incorporated kill chain model based on ATT&CK tactics, which are explained in the context of attack actions in Section 3.2.4.

Fig. 2 depicts an example of all possible relationships and interconnections. In the example, element (A) depicts an asset that is parent to two other assets. Area (B) shows the assets that operate within its context and that may inherit the parent's status (i.e., level of compromise), if so desired. Element (C) is an individual asset physically connected through network link (D) to another asset. These connections are access vectors an attacker has to follow in their lateral movement. Connection (E) depicts an external (exposed) access vector attackers may exploit from the outside (i.e., through the asset's public internet connection). Some connections, such as (F), only allow for certain types of attack (e.g., confidentiality/data theft), since the exposed asset itself may not have the required level of access. Arrow (G) symbolizes a dependency; here, the asset on the left will inherit some or all of its statuses to the asset on the right.

Said statuses are represented by an impact indicator (1) that keeps track of a single asset's Confidentiality, Integrity, and Availability damage. Like CAPEC, this impact ranges from 1 to 3, with 3 signifying the highest attainable level of compromise in our model. For example, 3 points of Integrity damage means that the attacker has completely taken over an asset or successfully altered a system's configuration or data to reflect their ultimate goal. Maximum Confidentiality damage equals a successful data theft and awards 1 point of Insight, while 3 points of Availability damage translates to a system that has been taken offline.

The final indicator, labeled (2), tracks the attacker's progress along the simplified kill chain. See Section 3.2.4 for more information on attack stages.

Note that we are able to designate any asset as 'critical' in the context of a scenario. Such assets will have additional adverse effects on the defender if taken offline. Next to various monetary penalties, they may even cause the defender to go out of business within a specified time frame.

3.2.4. Actions

Actions are at the core of the model. They represent the aforementioned attack techniques/patterns and defensive controls and describe what each actor does to compromise or protect the assets in play. There are 315 actions currently modeled, 195 of which are attack actions. In this subsection, we take a closer look at main and support actions for both attack and defense, the underlying attack stage modeling, types of defense actions, as well as our take on depicting asset damage, prevention, and recovery.

Main & support actions. In our model, there are two fundamental types of actions: main and support. Main actions describe techniques that can be utilized on their own, such as a "Brute Force" attack action or the "Limit Logon Attempts" preventative defense action. Most actions fall into this category.

Support actions, on the other hand, do not constitute an attack or mitigating control — at least not on their own. Instead, support actions are used in combination with compatible main actions to provide bonuses for one actor or introduce detrimental effects that hinder the other. For example, the action "Random Domains" can support the "Drive-by Compromise" attack by reducing the likelihood of detection. In our model, we use this approach to establish context between actions. Referring to MITRE ATT&CK, we define all actions belonging to the "Defense Evasion", "Command and Control", and "Collection" tactics as support actions. We argue that evasive (e.g., obfuscating) efforts should be contextually linked to an action (i.e., the activity being hidden), while command and control activity is used to trigger other attacks. Similarly, collection of data on a target system is always done in concert with data exfiltration, being a prerequisite ("support") for data theft.

If desired, our model can also be used to translate ATT&CK's techniques to main actions and the corresponding sub-techniques to support actions. In this case, support actions could be understood as a more fine-grained view on an attack. For example, the "Script Execution" action (MITRE ATT&CK: "Command and Scripting Interpreter"[5]) could have support actions specifying whether the script in questions is executed in PowerShell, JavaScript, or Python.

All main actions have an inherent base *success chance* that is modified through various means. The algorithms for how these chances

[5] https://attack.mitre.org/techniques/T1059/.

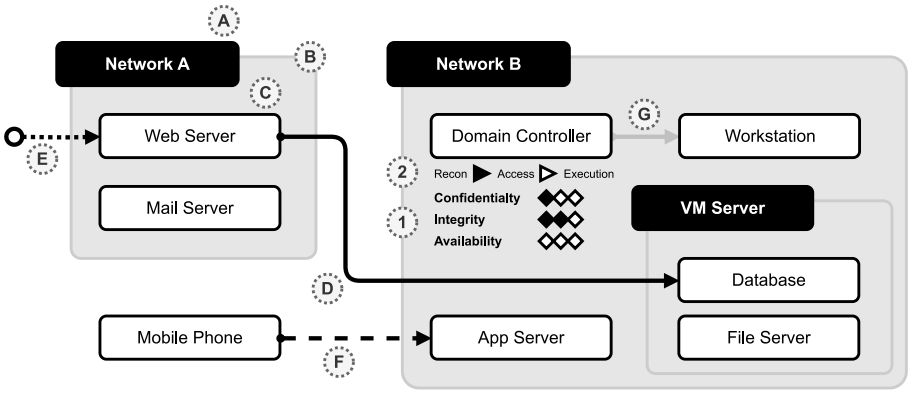

Fig. 2. Example infrastructure with several (nested and standalone) assets. Black arrows denote connections between assets while the gray line exemplifies a dependency.

are computed can be found in Section 3.4.1. Similarly, actions have a base detection chance that is similarly boosted or penalized (see Section 3.4.2).

The model considers that actions have a different *implementation time*. Specifically, we incorporated a currency called "action points", ranging from 1 (near instantaneous) to 5 (long-term activity such as the implementation of a company's "Least Privilege"[6] policy). Each action is assigned such a temporal cost. As stated above, one game turn translates to 3 AP — this means that actions may take more than one turn to complete.

Lastly, all actions have *constraints* that determine when and how they can be used:

- Skill requirement: Minimum actor Skill required to use the action (see Section 3.2.2). The algorithm for assigning this value to an action is detailed in Section 3.5.
- Elevation requirement: Some actions may require the attacker to be administrator or root on the given asset.
- Equipment requirement: There are actions that cannot be used without certain equipment. For example, certain file-based monitoring may need a "Host-based IDS", while the attacker's "Vulnerability Scan" requires a "Vulnerability Scanner".
- Asset type: Actions may only be compatible with certain types of assets, such as web servers, network appliances, or industrial control systems.
- Operating system: Some actions will only work if the asset has a specific operating system installed (e.g., Windows, Android, etc.)

In the following, we specifically discuss attack and defense actions.

Attack actions. Attack actions represent the attacking actor's means of compromising an asset. Each attack action is linked to an *attack stage*, potentially deals *damage* represented as CIA impact, and comes with additional *effects*.

Each attack action is assigned an *attack stage* that denotes when and how it can be used on an asset. Asset by asset, each stage must be successfully completed through an associated attack action in order to enable the use of actions belonging to the next stage. Refer to Fig. 3 for an overview of how our attack stages are linked to MITRE ATT&CK's tactics. In short, we differentiate the following 3 attack stages:

- **Reconnaissance**: When completed, the attacker has gathered information about the asset and is ready to launch their initial attack. Recon actions include various scans and techniques like open-sources intelligence (OSINT) gathering.

[6] https://csf.tools/reference/nist-sp-800-53/r5/ac/ac-6/.

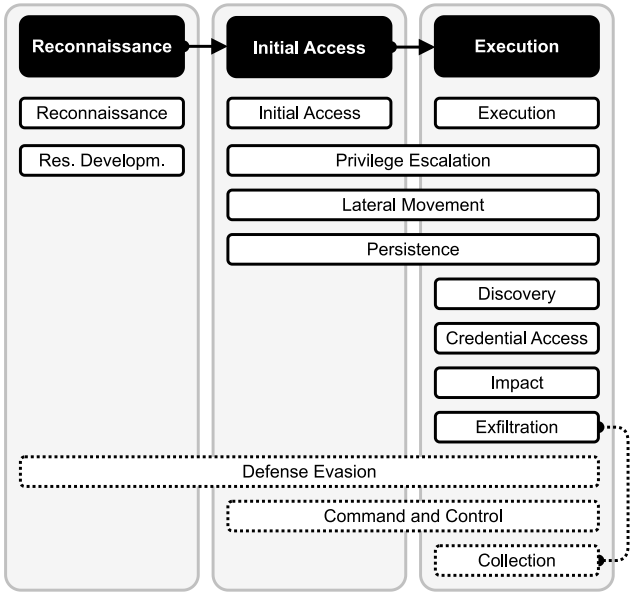

Fig. 3. Overview of the model's attack stages. ATT&CK techniques can be associated with one or several attack stages. Dotted boxes denote "support actions" that can be combined with actions from other tactics (see 3.2.4 for more information).

- **Initial Access**: Once successful, the attacker has gained a foothold in the system, enabling a wide range of subsequent (Execution) attacks. Access actions include remote connections, social engineering attacks, and exploiting technical vulnerabilities in an app.
- **Execution**: The attacker has brought to bear their payload, which seeks to further their main objectives. Execution actions may encompass website defacement, configuration changes, and data theft.

Preceding attack stage unlocks can be removed from an asset through successful defense actions if the means of initial access is thereby eliminated by the defender.

Many attack actions have a direct *impact* on an asset if successfully executed. As referenced in Section 3.2.3, this level of asset compromise ("damage") dealt follows the CIA triad (Stoneburner et al., 2002) and can reach up to 3 points. Actions deal anything ranging from zero points (those will have alternative effects such as the ones listed below) to 3 points in one, two, or all the damage tracks. In many cases, the player can choose which type of damage should be dealt. The assignment of damage values to actions is explained in Section 3.6.1.

Attack actions may have additional *effects* on an asset. Since these effects may also apply to defense actions and tools, we discuss them

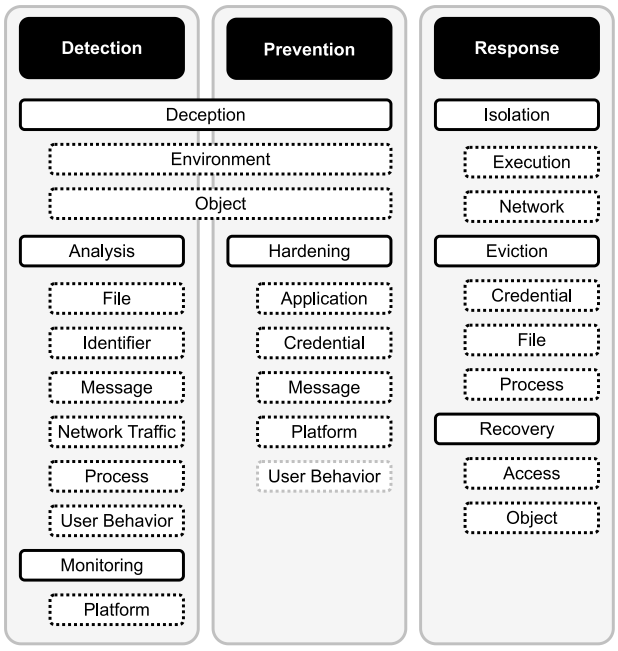

Fig. 4. Overview of the model's defense types. D3FEND categories are linked to our 3 basic types. Dotted boxes denote the different techniques contained therein. The gray dotted box is a custom technique that was not part of the original ontology.

separately in Section 3.2.6.

Additional properties associated with attack actions mostly revolve around two limiting values: *Detection* time and *counter* time limit. The first models actions that can only be detected by the defender within a certain number of turns. Such attacks are rare, however, since most of them leave traces that can be spotted at a later point in time through e.g., a forensic investigation. The counter time limit, on the other hand, is commonly seen: It describes how long after the fact the defender can mitigate a detected attack. Some attacks, such as "Multi-Factor Interception",[7] can only be countered immediately, lest the effect becomes irreversible. This limit applies in particular to data theft: once the information has been stolen, it is gone for good.

Defense actions. Defense actions are the defending actor's primary means of detecting, preventing, and responding to an attacker's actions. Although defense actions do not have a stage as attack actions do, they are of a certain that denotes their main purpose. Defense actions help recover from damage or prevent it entirely. They also may have additional effects.

Our model uses the three pillars of information security defense (LaPiedra, 2000) to depict what actors can do to mitigate attacks. As shown in Fig. 4, we map these pillars to the tactics and techniques of MITRE D3FEND (MITRE Corporation, 2023f). Specifically, our three *types* of defense actions are:

- **Detection**: This type of defense action is directly linked to the "Detect" tactic.[8] Here, we model all defense activity that revolves around the analysis of files, messages, traffic, various OS and user behavior in addition to platform monitoring facilitated by intrusion detection solutions. In the game, these actions typically increase the chance of detecting certain attacks (see Effects in Section 3.2.6).
- **Prevention**: These actions are synonymous with the "Harden" tactic[9] and depict preventative measures that are designed to stop an attack before it can unfold. We also include actions linked to deception ("Deceive" in D3FEND[10]) in this category.
- **Response**: In our model, response-type defense actions are about reacting to an attack by isolating[11] the cause, evicting[12] the attacker, and/or restoring[13] an asset. In game terms, response actions help recover from damage dealt by a prior attack.

Damage recovery (i.e., healing) directly negates damage dealt by attacks. This way, the e.g., "Backup Recovery" response (recovery) action can mitigate the damage caused by the "User Data Encryption" attack. This mechanic is the closest our approach comes to a zero-sum game.

In contrast, *damage prevention* is handled by a mechanic we dubbed a "shield": This shield is unique to prevention-type defense actions as it absorbs future incoming damage of certain attacks and can be configured to be limited in time and efficiency. For example, the defense action "Security Awareness Training" based on the control with the same name found in NIST SP 800-53[14] will negate 2 points of damage of all associated attacks with diminishing returns, signifying the effect that users will over time forget the lessons imparted. Shields can be either an exhaustible resource spent negating attacks, or an infinitely regenerating resource that will simply deduct damage from all related attacks. If all damage dealt by an attack is absorbed by a shield, the attack fails.

Effects related to all actions and tools are centrally discussed in Section 3.2.6.

3.2.5. Tools

We use the terms *tool* and *equipment* interchangeably to describe various, mostly technical appliances and software that may enable, support, or hinder an action. Tools have attack stage, asset type, and operating system constraints similar to actions (see Section 3.2.4) and may provide beneficial or detrimental effects (see below). They often serve to satisfy action constraints themselves, meaning that a certain piece of equipment may be required to use a certain action.

Equipment is either granted by an action (e.g., credentials) or procured for money. For this reason, each item is given a monetary value that models both the initial cost of procurement as well as implementation and training effort.

Attack tools. Our model currently incorporates the following categories of tools for the attacking side:

- **Attack Tool**: Attack tools are solutions typically implemented in software that aid the attacker in their malicious task. Examples include vulnerability scanners, exploit databases, keyloggers, and rented DDoS botnets. Once procured, attack tools permanently provide their bonuses to all related actions.
- **Malware**: In this category, we bring together tools that are directly 'attached' to an attack action to provide their benefits once. This includes malicious software such as ransomware and cryptojackers.
- **Exploit**: Other than malware, exploits are designed to not only support a linked action upon use, but to remain attached to an asset in order to provide persistent bonuses for future attacks. Examples include abstracted versions of known exploits such as Heartbleed[15] as well as generic zero-day exploits or backdoors.

[7] https://attack.mitre.org/techniques/T1111/.
[8] https://d3fend.mitre.org/tactic/d3f:Detect/.
[9] https://d3fend.mitre.org/tactic/d3f:Harden/.
[10] https://d3fend.mitre.org/tactic/d3f:Deceive/.
[11] https://d3fend.mitre.org/tactic/d3f:Isolate/.
[12] https://d3fend.mitre.org/tactic/d3f:Evict/.
[13] https://d3fend.mitre.org/tactic/d3f:Restore/.
[14] https://csf.tools/reference/nist-sp-800-53/r4/at/at-2/.
[15] https://heartbleed.com/.

- **Credentials**: This unique category is used to model ATT&CK's "Valid Accounts" technique[16] in various variants (also see Section 3.2.6 below). It contains credentials (or hashes thereof) that can be used to gain legitimate access to a system.

Defense tools. On the defending side, we model the following categories of tools:

- **Security System**: The direct equivalent of attack tools, security systems provide permanent bonuses for certain defense actions, increase detection chances, and may even hinder certain attacks. Examples include various intrusion detection and prevention systems, packet filters, or honeypots.
- **Analysis Tool**: These tools are directly attached to an appropriate detection action to generate Insight. Detonation chambers, API monitors, and debuggers are found in this category.
- **Fix**: Fixes are the opposite of exploits. They neutralize exploits placed on assets and may provide persistent bonuses for current or future defense measures.
- **Policy**: Policies are the organizational equivalent of security systems. They are required for certain actions to be effective (e.g., the "Separation of Duties"[17] action will require an access control policy to be in place) and provide a wide range of bonuses through their effects.

3.2.6. Effects

Actions as well as equipment (see Section 3.2.5) may have additional effects on other actions, assets, or the actor directly. These effects include:

- Attack Boost & Penalty: This effect modifies the success chance of attack actions by increasing or decreasing their base likelihood of success. Example: Most exploit equipment items in the game, such as the "API Remote Execution" exploit, increase the attacker's success chance for compatible actions (e.g., the "Buffer Overflow"[18] attack). Note that the defender's actions and tools may also decrease the opponent's success chance.
- Defense Boost & Penalty: Similar to attack chance, the chance of successfully defending against certain attacks can also be modified. For example, the defender's "Backup Server" equipment boosts defense success chance for the associated "Backup Recovery" action (techniques found within D3FEND's "Restore" tactic[19]) by a notable margin.
- Detection Boost & Penalty: Actions and equipment with this effect alter the base detection chance of an attack. For example: The "App Icon Hiding" attack support action based on ATT&CK's T1628 technique[20] decreases the detection chance of compatible main actions by a significant percentage, since the malicious app is no longer visible to the user.
- Budget Increase & Decrease, Budget Transfer: Next to cost of procurement, actions and equipment may also alter a player's current budget. For example, ransomware modeled as game equipment will increase the attacker's funds while decreasing the defender's. On the other hand, the "Information Procurement" reconnaissance action[21] will provide additional effects for a monetary cost upon use.
- Insight Boost & Insight Prevention: Reconnaissance actions in particular, such as "Endpoint Info Gathering",[22] grant Insight to the actor. As mentioned in Section 3.2.2, Insight measures the level of knowledge about the opponent and in turn provides accumulating attack, defense, and detection boosts. On the defense side, the player can use actions such as "Decoy Service"[23] which hinder intelligence gathering and, in game terms, prevent Insight gain.
- Asset Discovery: Discovery-type actions like "Network Service Discovery"[24] will unveil hitherto unknown assets on the game board. This mechanism allows us to designate certain assets as initially hidden and provide players with the means to discover them using certain actions.
- Grant Equipment: This super-class of effects awards the actor with a piece of equipment, which we currently use in 3 ways:
  - Credential Gathering: Actions modeled after the techniques within the "Credential Access" tactic[25] typically provide a "Valid Account" item, which enables attacks that require genuine user credentials.
  - Hash Dumping & Hash Cracking: Some actions, such as the "Credential Dumping" action modeled after the T1003 technique[26] provide a "Valid Hash" card instead. This item can be converted into a valid account through a hash cracking action.
  - Detect Vulnerability: Attack actions like the "Vulnerability Scan" have a certain chance to provide exploit-type equipment cards of varying quality.
- Privilege Escalation & Remediate Privilege Escalation: Attacks linked to ATT&CK's "Privilege Escalation" tactic[27] grant admin privileges on an asset, which may be a constraint for other actions. The defender has means to remediate this through actions like "Access Invalidation" modeled after D3FEND's "Authentication Cache Invalidation" technique.[28]
- Persistence & Remove Persistence: We model persistence[29] as damage that is repeatedly applied to an asset each turn until fully mitigated by an appropriate defense action such as "Process Isolation".[30]

Additional effects that further increase the accuracy of the gamified model are constantly being added.

3.3. Mappings

A key aspect of the model is the link between actions and tools. Each defense action should only detect, prevent, or mitigate certain attacks. Similarly, the various tools mentioned above should provide their bonuses only when it is contextually appropriate.

To achieve this goal, we have experimented with different approaches, many of which have been discarded in earlier evaluations. This included a natural language based mapping, matching through proxy categories akin to what MITRE ATT&CK is doing with its mitigations, and more. Neither approach proved to be accurate enough for our purpose.

With BRON (Hemberg et al., 2024) and its openly accessible database[31] we found a springboard that combines many of our chosen

[16] https://attack.mitre.org/techniques/T1078/.
[17] https://csf.tools/reference/nist-sp-800-53/r5/ac/ac-5/.
[18] https://attack.mitre.org/techniques/T1190/.
[19] https://d3fend.mitre.org/technique/d3f:RestoreObject/.
[20] https://attack.mitre.org/techniques/T1628/001/.
[21] https://attack.mitre.org/techniques/T1597/.
[22] https://attack.mitre.org/techniques/T1592/.
[23] https://d3fend.mitre.org/technique/d3f:DecoyEnvironment/.
[24] https://attack.mitre.org/techniques/T1046/.
[25] https://attack.mitre.org/tactics/TA0006/.
[26] https://attack.mitre.org/techniques/T1003/.
[27] https://attack.mitre.org/tactics/TA0004/.
[28] https://d3fend.mitre.org/technique/d3f:AuthenticationCacheInvalidation/.
[29] https://attack.mitre.org/tactics/TA0003/.
[30] https://d3fend.mitre.org/technique/d3f:Kernel-basedProcessIsolation/.
[31] http://bron.alfa.csail.mit.edu:8529/_db/_system/_admin/aardvark/.

data sources and attempts to map them as a whole. In addition, we considered the link between MITRE ATT&CK and D3FEND that is provided for some of the techniques in the latter's ontology.

Neither source provides a complete mapping, however. To fill the gaps and map attack to defense actions as well as actions to tools, we revisited every data source mentioned in Section 3.1, followed by several mapping passes performed by a group of security experts. For actions, this included assessing the 120 defense actions currently in the model and mapping them to a total of 195 attacks. The result was double-checked against BRON and the ATT&CK–D3FEND mapping, where available, and finally evaluated with a different group of information security specialists. The same was done for the 145 equipment items currently modeled.

The results of these mappings can be viewed in the playable alpha version of our game (see Section 6) via the action info screen depicted in Fig. 10.

3.4. Probabilities

The model allows users to set the base probability of action success and its likelihood of detection. By default, we provide a set of rules that determine said probabilities for what we consider an average setting, while factoring in the actor attributes of *Skill* and *Insight*. Since it makes sense for organizations to adapt these probabilities in accordance to their own risk assessment, our numbers are likely to be used as a baseline only.

3.4.1. Success probability

Success probability determines the likelihood that an attack or defense action is used successfully. Algorithm 1 in Appendix A summarizes our approach for attack actions. In short, we split ATT&CK tactics into four categories that modify the base success chance of 50%. In general, Reconnaissance actions (see Section 3.2.4) are most likely to succeed. Initial access is considered hardest in terms of difficulty, while most Execution-type actions are close to the aforementioned base value – the exception being credential gathering and privilege escalation attacks, as well as attacks requiring physical access, which are considered more difficult to perform.

Defense actions work in a similar manner. Here, actions are categorized as Prevention, Detection, and Response (with a distinction of immediate response and recovery). See Algorithm 2 for details on defense actions.

In real-world situations, these base values are heavily influenced by an organization's risk assessment metrics. Future iterations of the model (also see Section 5.3) will incorporate a success probability factor based on an organization's risk matrix. The current model, however, focuses on highlighting which attack from a large set of techniques is more likely to succeed in the given infrastructure.

Armed with these base values, different approaches can be taken to quantify the odds in real world attack scenarios (Tavolato et al., 2024) – such as modeling scenarios as Discrete-Time Markov Decision Processes (DTMDP), or the approach for blockchain-based edge networks demonstrated by Halgamuge (2022).

3.4.2. Detection probability

Detection probability defines how likely it is that the opponent spots an attack or notices an implemented defense measure. As with success chances, we work off a baseline value and modify it in respect to its attack stage or defense type. For attack actions, the detection probability increases with the progressing attack stage and is lowered by attacker-side privilege escalation. Algorithm 3 provides details.

Defense actions – response-type actions in particular – can also be detected by the adversary. While some are obvious to the attacker (e.g., Eviction actions used in response to an attack), others are less visible – or even undetectable. See Algorithm 4 for the specifics.

3.5. Skill requirement

Similarly to probabilities, each action has certain requirements — most notably a level of Skill required to execute it.

Each attack action is assigned a Skill requirement depending on its inherent complexity, required level of access, and other factors. As with probabilities, this value can be adjusted to better depict the difficulty of e.g., successfully conducting an attack on a specific asset. Algorithm 5 serves as a baseline.

Defense actions are assigned a minimum required Skill rating as well. See Algorithm 6 in Appendix A for more information. Here, we primarily consider action complexity, implementation time, and its type (detection, prevention, or response), resulting in a generally lower skill requirement, which mirrors real-world situations: Unlike attacks, means of mitigation are generally well documented and do not rely on undesired functionality.

3.6. Impact values

Actions often come with damage and recovery values which represent the potential impact of a specific technique in regard to asset compromise or attack mitigation. Below, we briefly explain how said values are computed.

3.6.1. Damage

Damage values, which range from 0 to 3, refer to the potential impact an attack action has on one or several of the defender's assets (see also Section 3.2.3). Since this value is derived from properties (attack complexity, permissions needed, attack stage) similar to the Skill requirement detailed above, we use it as additional input to our Algorithm 7. Refer to the appendix for additional details.

3.6.2. Recovery & prevention

Recovery and prevention impact values are simpler in their calculation. Unlike damage values, they consider neither attack stage nor skill requirement and initialize their initial recovery to 3 (damage - 3) instead of 0. Applicability serves to nuance the final score. Only "Response"-type actions can recover damage. Damage prevention is offered by "Prevention" actions that negate future incoming damage.

4. The game

In this section, we discuss how the overall model is molded into the format of an online two-player board & card game, PenQuest, and which key aspects of gamification were considered in its conception.

4.1. Game rules

For reasons of gameplay and to enable AI-driven learning, we opt to use a turn-based as opposed to a real-time approach. This allows us to model a cause-and-effect loop as well as match attacks to existing prevention or detection measures.

Generally, the game alternates between attacker and defender, giving each side the opportunity to play a number of actions (limited by their action point cost) and to acquire new tools. Used actions are replaced at the end of the turn, signifying the process of research and preparation on both sides. Note that each actor is required to play at least one action per turn, but may string together several actions worth a combined maximum of 5 AP.

Overall, the game has three main phases: an optional *Setup* phase, *Stealth* phase, and *Conflict* phase. In Setup phase, the defender implements his or her baseline security by procuring tools and playing a number of actions for free. After initial shopping, the game starts off in Stealth phase and only progresses to Conflict stage once the defender has detected at least one of the attacker's hostile actions.

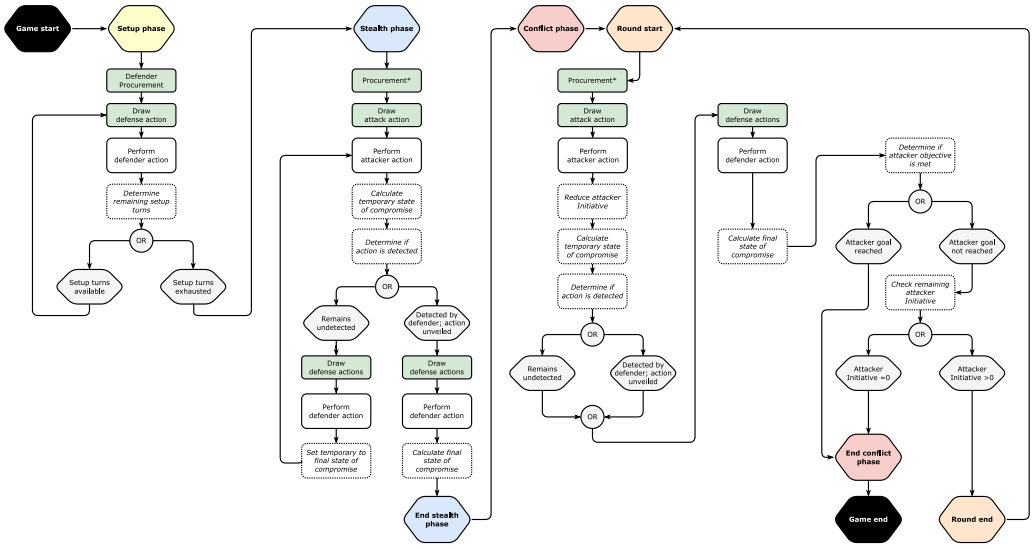

Fig. 5. Game sequence encompassing all stages. The main game loop includes the procurement of tools (procurement, marked by an asterisk, can be triggered on demand anytime during the player's turn), playing of both attack and defense actions (with different rules depending on whether an attack was detected), action replenishment, and numerous decision points that help determine the game's overall progress. Dotted boxes represent computations performed by the game engine. Setup phase can be changed in length or skipped entirely to depict the defender's preparedness level.

As mentioned in Section 3.2.2, the attribute Initiative serves as a time limit. If the attacker meets their goal before Initiative runs out, they win. If the defender manages to hold off the attacker until Initiative reaches zero, victory goes to the defender. At this point, we assume that the attacker will not manage to defeat their opponent's defenses given their current skill and resources.

Refer to Fig. 5 for an in-depth look at the game loop. Note that the scenario designer can define additional defeat conditions in the form of offline penalties (not pictured): It is possible for the defender to lose the game if certain assets remain offline for a specified number of turns after a successful availability attack.

4.2. Gamification

Gamification in general can be contemporarily defined as "the application of gaming mechanics to non-gaming environments to make difficult tasks more palatable" (Growth Engineering, 2023). These difficult tasks – understanding the fundamentals of information security and cyber threats, as well as the complex process of risk assessment – are at the core of our approach.

For our gamified implementation, we define the following requirements:

- **Narration**: The game tells a story in regard to its actors and assets that can be freely defined and adapted to suit an organization's setting.
- **Education & Dissemination**: The game imparts information security knowledge to intermediate IT users and security novices. It can serve as a learning platform in the context of the data sources chosen. In addition, the game tracks the actors' behavior and the effect of every action taken, enabling the dissemination of tailored mitigation suggestions for risk assessment (i.e., lessons learned).
- **Motivation**: The game utilizes reward mechanisms to motivate users to play and learn.

In the following, we discuss how we meet each of these requirements. To provide a measure of player motivation, we link each relevant feature to Bartle's model (Bartle, 1996) as well as an updated version called ACE2 (van Dam and Bakkes, 2019), which focuses more heavily on the creation of new content as a driving factor for player engagement. These and similar models often serve as a test of gamer psychology and have been widely used to measure a game's success with the various player types (Yee, 2007; Kapp, 2012).

See Section 4.2.3 for more information about motivating different player types and Fig. 6 for an overview of how our take on gamification can be mapped onto the different player types.

4.2.1. Narration

The first requirement for gamification we determined as important is *narration*. In PenQuest, there are a number of concepts and features that lend themselves to this key item, all of which are detailed below.

Scenarios. As mentioned in Section 3.2.1, scenarios provide objectives as well as the narrative glue of the game. The latter is key to helping players engage with the setting. We outline the game board in story terms and define why the attacker seeks to compromise a system in addition to presenting an overall plot. For example, we may give the attacker moral justification to steal certain information from a dubious firm or introduce urgency by telling the defender that they need to hold down the fort until a certain event in their organization's development.

Providing background information and story is one of the strongest motivational factors for 'Explorer' types as defined by Bartle's taxonomy of players (Bartle, 1996). ACE2 (van Dam and Bakkes, 2019) further differentiates between 'Explorers' and 'Engagers', who are similarly driven by experiencing a game's narrative. No matter the taxonomy, story and lore keeps these players engaged and helps them remember their experience for a longer period of time.

This and additional background information is provided via the scenario information screen displayed upon starting the game and accessible at any time during a session, as well as through informational messages that can be triggered at certain points in the game.

Avatars & personas. Directly linked to a scenario, personas – and their visual avatar representation – assign each player an identity, appearance, skills, background, and the aforementioned personal motivation. The goal is to have players personify a character as they would in a role-playing game, allowing them to experience the story as a different person/organization. This targets 'Engagers' (van Dam and Bakkes, 2019) in particular.

The concept of personas can be helpful in different applications, including human-centered AI (Holzinger et al., 2022): They map users' mental models to specific contexts, which is necessary to develop new approaches for future human-AI interfaces. Human-AI interfaces differ

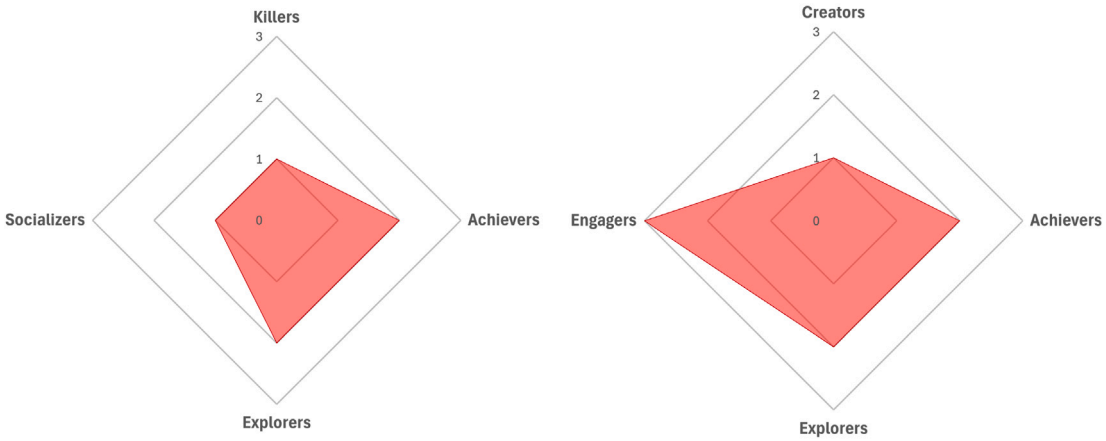

Fig. 6. Number of game features considered a strong motivational factor for player types modeled after (Bartle, 1996) on the left and van Dam (ACE2) (van Dam and Bakkes, 2019) on the right. Each mentioned and implemented gamification component that lends itself to motivating the respective player type increases the score by 1.

from classic human–computer interfaces, because AI today has some degree of human-like cognitive, self-executing and self-adapting abilities and also generates unexpected results that require non-deterministic interactions (Holzinger and Mueller, 2021).

In the game, players could play as a disgruntled employee and novice IT user who was fired because of a misunderstanding, or a high-profile intelligence agent who seeks to sabotage another nation-state's uranium enrichment program out of a sense of patriotism.

The player avatar and information about the persona is visible in the game's lobby, the aforementioned scenario information screen, and may be further detailed via information pop-up messages within the game.

4.2.2. Education & dissemination

Educating users and providing risk scenario exploration in addition to decision-making practice and quantitative assessment (i.e., lessons learned) are the game's primary objectives. This is achieved by three means: the various *information screens* offered to the player, detailed *logs and summaries* provided by the front-end, and scenarios using educational (e.g., physical-world) *metaphors*.

Information screens. The game's front-end (see Section 4.3) implements on-demand pop-up windows for each asset as well as action. The asset information screen includes a description in scenario terms in addition to gameplay-related information about the asset's current attack stage, type, operating system, active effects, and an attack/defense action history. This allows us to explain what certain systems do and how they interact with others.

The action information window (see Fig. 10 in Section 5.2) is even more vital. It presents an official description/definition taken from e.g., MITRE ATT&CK. In addition, it will inform the player which action of the adversary serves as a counter; i.e., which attack is detected, prevented, or mitigated by which defensive measure, or vice versa. Compatible support actions and tools are listed as well. This not only allows the players to optimize their strategy, but ultimately teaches them how actions and tools are linked together in both the game and the real world.

Logging & summarization. The game's in-built log records every action played, effect triggered, and damage caused/healed. This allows users to keep track of relevant events and which consequences their actions have on assets and actors in general.

In addition, the game presents the player with a summary at the end of a session. Here, statistics for actions and various effects are provided. Furthermore, it visualizes the entire attack/defense history and how each action contributed to the outcome of the game. This way, players and risk managers alike learn which techniques and countermeasures were particularly effective. An example for this summary is presented as part of a use case in the Evaluation section, Fig. 9, Section 5.1.

In future iterations, this feature will be expanded with an LLM-based explication of the collected (log) events, eventually turning the tabular presentation into human-readable text.

Educational metaphors. As mentioned in the introduction of the model, both actions and assets can be freely named and altered to suit the stakeholders' needs. This allows us to fully change a scenario to e.g., serve as a real-world metaphor. For example, we transformed a data theft scenario into a break-in where the attacker has to infiltrate a castle and steal an item from the treasury. Instead of phishing, drive-by compromises, and vulnerability exploitation, they attempt to con guards, hide themselves in a delivery cart, or use hidden structural weaknesses to scale the wall.

This approach is especially well suited for beginners and allows us to better teach basic concepts without relying too much on complex IT terminology.

4.2.3. Motivation

Ultimately, educating learners requires continuous engagement and, by extension, means of motivation. Many of the aforementioned aspects (e.g., scenario story-telling) already contribute to this goal, but are tailored to specific player types. PenQuest uses the following motivational features in regard to its educational use:

- Damage points: Introduced in Section 3.2.4, this numeric representation of system compromise serves as an important currency in tracking the game's progress. They are visually represented and highly motivate 'Achievers' and – to a certain extent when representing the attacker, 'Killers' – in Bartle's model or 'Engagers' in ACE2.
- Kill chain unlocks: Progressing along a kill chain for unlocking more effective attacks hits the same note as damage points and is visually shown on each asset as well as corresponding actions.
- Uncovering hidden assets and unlocking new access vectors: Using actions that have 'Asset Discovery' effect will visually unveil new assets to interact with, expanding the game board and keeping 'Explorers' interested. The same applies to unlocking new connections to assets that were previously unreachable by compromising an asset within its attack path.
- Experience points and leaderboards: The game awards points for each victory as well as for specific in-game accomplishments (e.g., number vulnerabilities found or assets compromised) and allows players to compare their progress with the community. Future iterations of PenQuest will also award achievements to the

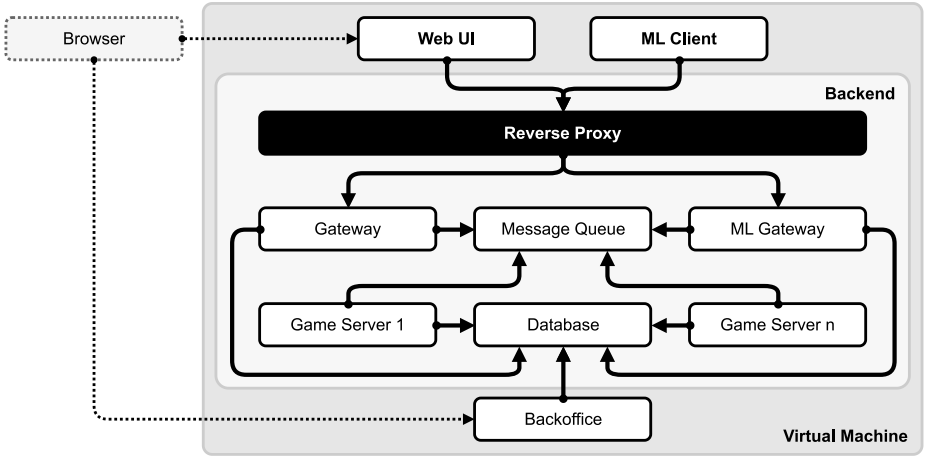

Fig. 7. Overview of the PenQuest architecture. Arrows denote the request direction.

player for reaching certain in-game milestones. Such mechanisms of competitive engagement are major motivational factors for 'Achievers' and 'Socializers' (Bartle, 1996).
- Community scenarios (WIP): Allowing players to contribute to the game by supplying their own scenarios and narratives is key to keep 'Creators' (van Dam and Bakkes, 2019) interested and helps the game grow. Currently, scenario building requires access to the game's backoffice application (see Section 4.3). Future iterations of PenQuest will make this feature public.

Fig. 6 visualizes the game's features in conjunction with the player types they are strongly associated with.

4.3. Implementation

This subsection introduces the technology stack used in our implementation of PenQuest and shows how the individual components are designed and connected. Refer to Fig. 7 for an overview.

The PenQuest software in its entirety is currently deployed as individual Docker containers within an Ubuntu virtual machine sporting 8 CPU cores and 8 GiB of memory. The containers correspond to the following (sub-)components that make up the *backend*:

- **Game Server** (1..*n*): The game server, written in C# within .NET,[32] is at the heart of PenQuest. It manages active games and lobbies, contains all logic corresponding to the features and mechanics explicated in Section 3, and handles errors and logging. Depending on demand, multiple game servers can be run as individual containers.
- **Database**: A PostgreSQL[33] database stores the specifics of our scenarios, actions, and tools, as well as active games and logs for later analysis.
- **Gateway**: The custom-built gateway orchestrates communication between the UI (via the reverse proxy), the message queue, and the database. It also handles the startup of new game server instances, if required by the operator.
- **ML Gateway**: Similarly, the machine learning (ML) gateway serves as an intermediary between the ML client (see below) used for reinforcement learning (RL).
- **Message Queue**: RabbitMQ[34] facilitates asynchronous communication between services in our architecture. It acts as a buffer for workload spikes and ensures reliable communication by storing messages to and from the game servers until they are processed and acted upon.
- **Reverse Proxy**: The NGINX reverse proxy[35] directs client requests to the appropriate backend server, in our case the two gateways. It provides an additional level of abstraction and response buffering, ensuring the smooth flow of network traffic between components.

Connecting to the reverse proxy from the client side, the *web user interface* (UI) or *front-end*, shown in Fig. 8, visualizes assets and their interconnections, as well as detected adversary behavior and own actions taken — including their individual effects on said assets. It provides all necessary means of interaction with the game's elements by enabling action drag-and-drop, action and asset information windows, and a means to procure tools and draw new cards that fit the player's strategy. In short: The UI visually incorporates all narrative, educational, and gamification aspects introduced in Section 4.2. Technology-wise, the front-end has been realized in Vue.js.[36] and is delivered to the browser by an NGINX[37] web server. User authentication is handled by Keycloak.[38]

The *ML client* is solely used by RL agents as part of our ongoing effort to infer optimal defense strategies. It enables autonomous training and testing for said RL agents. Since this implementation is functional but incomplete at the time of writing, we consider it future work (see Section 5.3).

Lastly, the *backoffice* (BO), a UIkit[39] web application, provides advanced tools for managing users, game server (backend) instances, for creating or altering scenarios, actions and equipment, for specifying tutorial and game log messages, and more. The BO is currently restricted to internal users but will eventually be forked and adapted into a public feature that enables the creation of aforementioned community scenarios.

5. Discussion

In this section, we sketch an exemplary use case and present a means of continuous evaluation for our model. In particular, we focus on the game's risk exploration capabilities (see Section 5.1) as well as the mappings between attack and defense actions (Section 5.2) that are at

[32] https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet-framework.
[33] https://www.postgresql.org/.
[34] https://www.rabbitmq.com/.
[35] https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/.
[36] https://vuejs.org/.
[37] https://www.nginx.com/.
[38] https://www.keycloak.org/.
[39] https://getuikit.com/.

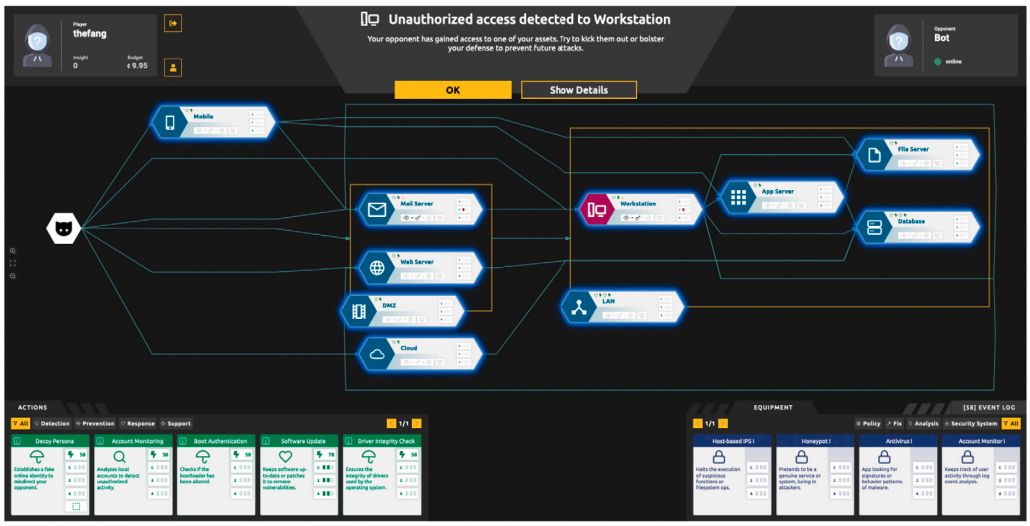

Fig. 8. UI screenshot depicting an example infrastructure, as seen by the defender. Actions and available tools are displayed below the assets. Actor, objective and game phase information is displayed on top. The attacker (i.e., outside world) is represented as icon on the far left and maintains a connection (line) to each asset that is accessible at the depicted moment in the game.

the heart of both the educational and risk assessment application of PenQuest. We conclude this section with an in-depth view on current limitations and planned future work (Section 5.3).

5.1. Use case

One of the more conventional use cases covered by a PenQuest scenario is the topic of ransomware. The game incorporates an example attack against the fictitious company of "AT Microsystems", a small player in the tech sector with average attributes (3 out of 5) in terms of Skill, Determination, and budget. In the scenario, they are threatened by a hired higher-skill (4), average motivation (3) actor with low monetary funds (2) who seeks to encrypt the firm's ERP system by attacking privileged workstations with ransomware tailored to spread to said asset.

The scenario assumes an architectural weakness in the infrastructure that allows unimpeded lateral movement from the workstations to the ERP system — something that, in a real-world application, may have been inferred as part of the asset identification stage of an early risk assessment.

Using PenQuest, we now want to explore the remaining risks and associated threats that may impact the firm. While focusing on availability attacks, we want to determine which MITRE ATT&CK techniques represented by our game actions would constitute the greatest threat to the infrastructure in question. For this purpose, the scenario uses a subset of attack and defense actions that are all related to data availability attacks and their possible prevention, detection, and response measures.

As part of a workshop-style risk exploration exercise, real-world security professionals have played the scenario against each other six times, with one additional game against a bot defender. The games lasted 19 min on average. In the following, we discuss these games in more detail, focusing on common decisions:

During Setup stage, the defender typically opted for organization-level prevention measures such as "Least Privilege"[40] or "Software Install Restriction"[41] as security baseline. Detection measures often revolved around URL scanning or mail reputation analysis. Both actors roughly used half their budget from the get-go to prepare for a number of eventualities. The remainder was typically kept in reserve to react to new developments or to explore additional attack vectors.

In the early game, the attackers often decided to use undetectable reconnaissance actions (e.g., open-source intelligence gathering) to increase their Insight, followed by vulnerability scans or information phishing. For initial access, a number of attacks were attempted, including but not limited to whaling, screen share abuse, and the placement of a malicious USB drive. In the few cases the attacking players managed to get access to one or both of the vulnerable workstations, they used actions like "System Data Encryption"[42] "Firmware Corruption"[43] to achieve their objective.

Out of the seven games, the defender won five, while the attacker surrendered the fifth and only managed to win the final game. The outcome was increasingly close: The attacker managed to encrypt a workstation (and, as a consequence, the ERP system) a total of three times. It was only thanks to well-placed response actions that the defender could usually undo the attack by restoring a backup in the same turn, minimizing the damage.

When a PenQuest game ends, a summary screen is shown (exemplified in Fig. 9. In addition to a full game log (see the Supplementary information for an example), this information can be used to better understand the sequence of events and enables risk exploration.

In summary, the following lessons were learned regarding this particular use case:

- Reconnaissance conducted by the attacker was always successful, thereby enabling additional stages of attack. The defender mostly neglected to prevent information disclosure, which would have reduced the attacker's Insight gain.
- Vulnerability scanning yielded knowledge of exploits, but was usually detected by the defender, ending Stealth phase and commencing open conflict (i.e., starting the 'ticking clock'). This cost the attacker valuable turns; more patient use of additional OSINT actions would have increased the attacker's chances, further emphasizing the need of information disclosure mitigation.

[40] https://csf.tools/reference/nist-sp-800-53/r5/ac/ac-6/.
[41] https://csf.tools/reference/nist-sp-800-53/r5/cm/cm-11/.
[42] Variant of, https://attack.mitre.org/techniques/T1486/.
[43] https://attack.mitre.org/techniques/T1495/.

- While the defender successfully used "Traffic Filtering (Inbound)"[44] to block the attacker's access after successful initial access, they missed to remedy the privilege escalation effect of an account compromised through "Whaling".[45] This credential was later used to attempt a firmware corruption attack while bypassing local system defenses with an appropriate support action.[46]
- The restriction of user's rights to install and use software proved to be very effective in preventing Initial Access and Execution-type attacks.
- "Sender Reputation Analysis",[47] "URL Analysis"[48] and "Sender Authentication"[49] detected or prevented most phishing-style attacks.
- Games where the defender used "User Account Monitoring"[50] and "User Access Pattern Analysis"[51] saw the most effective response to attacks like "Malicious USB Drive"[52] and "Screen Share Abuse".[53]
- The defender's investment into host- and network-based IDS/IPS was always rewarded, with a packet filter and mail gateway as close second. A dedicated backup server was vital in restoring an encrypted workstation before the game was lost.

If AT Microsystems were a real company, their information security personnel might now be incentivized to plan a real-world penetration test or launch a security awareness program to counter the most successful attacks seen during the PenQuest sessions. This could include educational phishing campaigns, vulnerability scanning, checking for information disclosure, and more.

While many countermeasures or recommended investments could likely be determined through conventional risk assessment in this simple scenario, playing the game significantly sped up the process and helped identify controls that the players might have otherwise missed. The educational effect of gamifying the exploratory process was highlighted by the participants as well as students (Luh et al., 2022), and was echoed by many others in the PenQuest community of almost 200 alpha testers at the time of writing.

Playing the game in this fashion, however, also highlighted the need for better scalability in terms of gameplay and session log interpretation; i.e. the ability to play more games in a shorter period of time with even smarter attackers and/or defenders, combined with the automated inference of lessons learned. Refer to Section 5.3 for upcoming AI updates that will address said shortcomings.

5.2. Continuous evaluation

In this subsection, we present a means of a community-driven evaluation for the mappings between attack (Recon, Initial Access, Execution) and defense actions (Detection, Prevention, Response), which is arguably the key component of our model. To allow the scoring of as many of the over 10,000 mappings currently defined, we implemented a 5-star rating system directly into the game (via the action information screen introduced in Section 4.2.2 and pictured in Fig. 10). The aforementioned alpha community as well as selected security experts are asked to leave their verdict on as many mappings as possible.

Since this community scoring approach is an iterative process that is part of the continuous improvement loop of the model/the game (scores are reviewed regularly and mappings are quickly updated or removed entirely), extracting an average score has proven to be limited in its significance. Generally speaking, we have never fallen short of an average score of 3.86 out of 5 points at any given point in time, with a median action mapping score of 4 stars.

We hypothesize that this community-driven approach will keep improving the model over time, especially considering our constantly growing user base of testers and contributors.

It stands to mention that the currently implemented evaluation feature does not yet allow users to suggest new mappings; only existing ones can be scored. Additions are currently handled in an expert group of both the authors and lecturers/researchers from two universities with dedicated security study programs.

PenQuest has, despite its alpha status, been used in various workshop and class settings revolving around risk exploration and general IT security education:

- Information Security Fundamentals classes for bachelor students,
- Company workshops for SMB's to explore risks in custom-built IT infrastructures,
- Parts of the IT security education of the Austrian cyber forces,
- "Teach the teacher" initiatives for high schools, and
- High school workshops introducing IT security concepts.

In general, the game can be tested in real-world scenarios through controlled penetration testing exercises or as part of cybersecurity training programs. By simulating realistic attack and defense scenarios aligned with actual infrastructure layouts and threat models, the game's effectiveness in identifying vulnerabilities and enhancing defensive strategies can be evaluated. Metrics such as detection time, accuracy in threat identification, and quality of mitigation responses can provide actionable insights, ensuring the game's applicability in practical settings.

Refer to Luh et al. (2022) for a preliminary evaluation of PenQuest's educational aspects.

5.3. Limitations & future work

Both the model and its programmatic implementation have limitations in various areas that offer room for improvement. In general, the versatility of PenQuest opens many avenues of research — be it for education, formal modeling, decision support, or automated pentesting and cyber range scenarios. Specifically, we want to focus on:

- **Model**
  - **Attack–defense and action-tool mapping quality**: The mappings between actions and tools can be improved in many ways, be it by better incorporating BRON (Hemberg et al., 2024) or ontologies like UCO (Casey et al., 2015), or by expanding on our own, community-driven scoring approach.
  - **Improvement of success and detection chance algorithms**: Currently, the algorithms presented in Section 3.4 are rather simplistic. Here, we see two main areas of improvement: Converting reports such as CISA's Risk and Vulnerability Assessment (Cybersecurity and Infrastructure Security Agency (CISA), 2023) into a success chance algorithm, or creating a means to automatically transform an organization's risk matrix into in-game values — something that has to be done manually at the moment.
  - **Improvement of cascading effects modeling**: While our model is capable of modeling various cascading effects in which compromising an asset can have effects on other systems, the possible delay of such effects is currently not considered. We plan to incorporate a mechanism to better depict the temporal component of e.g., outages in future versions of the model.

[44] https://d3fend.mitre.org/technique/d3f:NetworkTrafficFiltering/.
[45] Variant of https://attack.mitre.org/techniques/T1566/001/.
[46] https://attack.mitre.org/techniques/T1562/.
[47] https://d3fend.mitre.org/technique/d3f:SenderMTAReputationAnalysis/.
[48] https://d3fend.mitre.org/technique/d3f:URLAnalysis/.
[49] https://d3fend.mitre.org/technique/d3f:TransferAgentAuthentication/.
[50] https://d3fend.mitre.org/technique/d3f:LocalAccountMonitoring/.
[51] https://d3fend.mitre.org/technique/d3f:ResourceAccessPatternAnalysis/.
[52] https://attack.mitre.org/techniques/T1091/.
[53] https://attack.mitre.org/techniques/T1219/.

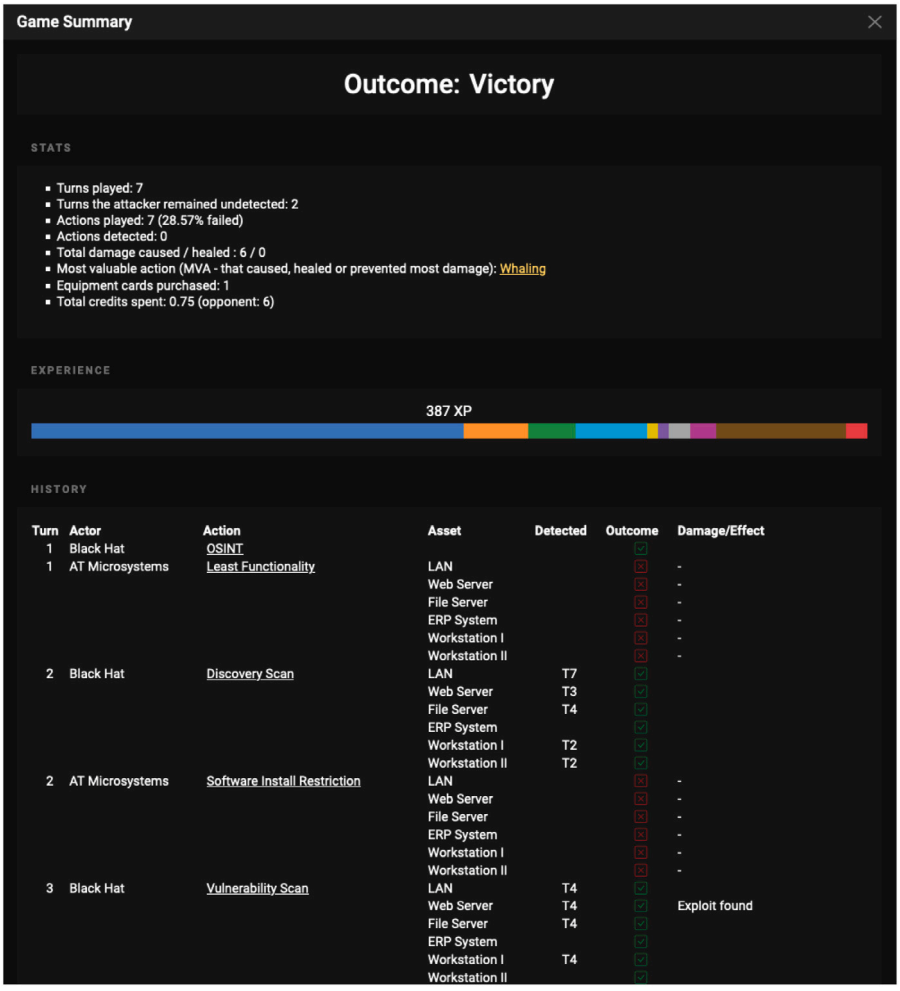

Fig. 9. Example summary window of a game won by the attacker. The summary shows all actions played in order (even the ones not detected during play) and returns a few additional metrics.

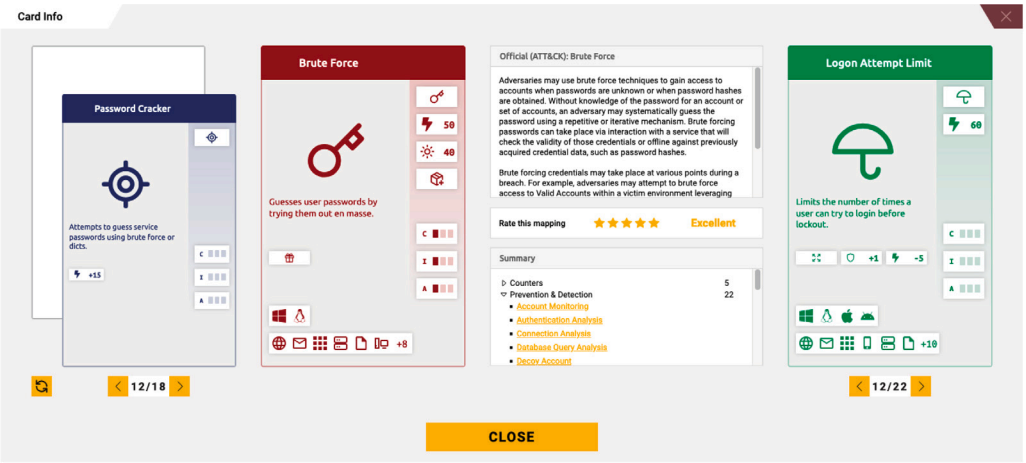

Fig. 10. Example action information screen showcasing the in-built rating function. Here, the link between the attack action "Brute Force" and the preventative defense action "Logon Attempt Limit" is shown.

- **Incorporation of asset (data) importance**: The current model does not explicitly incorporate asset importance as a factor for non-availability attacks. Asset importance determined by additional factors, such as data sensitivity, could significantly influence attack priorities and defense resource allocation. Future iterations of the model will integrate new importance metrics to better reflect real-world scenarios. This enhancement would align with existing literature, such as Halgamuge's analysis (Halgamuge, 2022) of success probabilities for attackers in critical systems, or Abdallah's behavioral decision-making framework (Abdallah et al., 2020).
- **Use of formal languages for threat modeling and scenario definition**: In order to better facilitate collaboration, we plan to store our scenarios as well as actions in accepted formats such as MAL[54] for threats or CRACK[55] for scenario definitions.
- **Enrichment of actions and tools**: To close the gap to penetration test planning and other practical implementations of our abstracted take on cybersecurity, we endeavor to append details such as command syntax and implementation examples to each action and tool, akin to Rak et al. (2024).

• **PenQuest**

- **Improved mechanics for exploits and credentials**: Exploits and valid credentials are currently modeled as global piece of equipment not tied to a specific asset. This simplification will in future be replaced by more granular mechanics.
- **Additional usability and educational features**: Next to UI/UX design in general, we aim to further improve upon PenQuest's educational benefits by adding e.g., quiz-based action success, observer functionality, as well as the means to inject events into active games.
- **Community contributions, scenario building**: The aforementioned scenario creator will help streamline the process of creating new attack–defense campaigns, thereby significantly increasing the scope of the game.
- **Additional model variants**: The framework of the game lends itself to modeling a multitude of (security) domains. New action sets for OWASP/ASVS, physical security, and purely organizational information security (different standards) are in the works.
- **Automated import pipeline from data sources**: Currently, our data sources are being imported and converted semi-automatically via JSON files (ATT&CK) or by parsing websites such as https://csf.tools/. This makes updates to newer versions slower — something that will be addressed by an import pipeline in the future.
- **Automated report & guideline generation**: The reports and logs produced by the game offer a lot of room for improvement in regard to detail and automated interpretation. In the first step, we will use an LLM-based approach to interpret the outcome of each game with the goal of creating a report that serves as human-readable decision support.
- **Reinforcement learning (RL) agents and AI-based strategy inference**: Training of RL agents autonomously playing the game against each other has already commenced. In combination with the previous item, we endeavor to infer optimal defense strategies at scale: It will no longer be required to play games against each other, but instead task the ML Client to play n games with arbitrary constraints and objective, such as the optimization of budget, impact mitigation, or time. This will eventually yield concrete suggestions for a given IT/OT infrastructure. RL either implements policy-driven learning (Sutton and Barto, 2018) or can be viewed as a human-in-the-loop paradigm, where human feedback improves human-agent interaction as it does for systems like ChatGPT (Retzlaff et al., 2024).
- **Investigation of formal game-theoretic aspects**: Due to the inherent complexity of PenQuest, deriving an analytical solution for the game-theoretic model is infeasible. As an alternative, we propose using a probabilistic model checking approach to identify optimal strategies for specific scenarios. The outcomes from this approach can be compared with the RL-based strategy inference mentioned above. If the results align, this will offer an interpretation (and not an explanation in the sense of Broniatowski (2021)) of the strategy suggested by the RL component. In cases of divergence, the model checking results will help identify the underlying reasons for these discrepancies.
- **Pentesting, cyber range tie-in**: One of our long-term goals for the project is to use PenQuest as full-fletched pentest planning and cyber range scenario deployment tool. We envision that assets placed on the game board will automatically create respective containers with a pre-defined security baseline. Through the link to real-world tools (see above), the appropriate tools/commands for executing an attack will be readily available, translating to the techniques abstracted in the game.

Work on many of these aspects has already begun. Future work includes a large-scale evaluation of PenQuest's educational features, as well as a long-term evaluation of the model's additional aspects introduced here.

6. Conclusion

PenQuest marries a number of individual features found in various areas of security research while providing a novel combined approach to risk exploration, education, pentest planning, and asset dependency simulation. In this paper, we present the model powering our ready-to-use serious game client and provide a glimpse at its many features and possible applications.

The current version of PenQuest has been successfully employed as part of university classes, school and company workshops, as well as military IT education. We have created the means for an ever growing security community to help improve the attack–defense action mappings and created an accurate baseline for training reinforcement learning agents that, in the near future, will help organizations to not only explore risks and possible threats in a workshop setting, but compute optimal defense strategies for any given infrastructure.

CRediT authorship contribution statement

Robert Luh: Writing – review & editing, Writing – original draft, Project administration, Methodology, Funding acquisition, Conceptualization. **Sebastian Eresheim:** Writing – review & editing, Software, Methodology, Conceptualization. **Paul Tavolato:** Writing – review & editing, Writing – original draft, Formal analysis. **Thomas Petelin:** Visualization, Software, Resources, Conceptualization. **Simon Gmeiner:** Software, Methodology, Conceptualization. **Andreas Holzinger:** Writing – review & editing. **Sebastian Schrittwieser:** Supervision, Funding acquisition.

[54] https://mal-lang.org/.
[55] https://github.com/enricorusso/CRACK.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded in whole, or in part, by the Austrian Science Fund (FWF) P 33656. The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association, Austria is gratefully acknowledged.

Appendix A. Algorithms

A.1. Success probability

Algorithm 1 shows the base attack success probability calculation for actions across all ATT&CK tactics for a given infrastructure that does not implement any meaningful security controls. All chances are based off of a 50% baseline. We argue that Reconnaissance actions as well as Resource Development and Discovery Actions are more likely to succeed (+20, and +10%, respectively) due to open source information and benign discovery tools utilizing existing APIs. Execution, Lateral Movement, Collection, and Impact actions rely on existing access and often employ living-off-the-land (LOTL) techniques. We such actions as baseline compared to Initial Access, Persistence, Privilege Escalation, and Credential Access techniques, which, in our model, are harder to perform (−10%).

Algorithm 1 Attack Success Chance Calculation

```
procedure SetAttackSuccessChance(action, skill, skill_req, long_term, insight, admin, physical_access)
    successChance ← 0.5                                   ▷ Initialize success chance to 0.5
    tactics ← [Reconn., Resource Dev., Initial Access, Execution, Persistence, Priv. Escalation, Credential Access, Discovery, Lateral Movement, Collection, Impact]
    for tactic in tactics do
        switch tactic do
            case Reconnaissance
                successChance ← successChance + 0.2
            case Resource Development, Discovery
                successChance ← successChance + 0.1
            case Execution, Lateral Movement, Collection, Impact
            case Initial Access, Persistence, Privilege Escalation, Credential Access
                successChance ← successChance − 0.1
                continue
    end for
    if skill_req = 2 then
        successChance ← successChance + 0.1               ▷ simple actions
    end if
    if skill_req = 1 then
        successChance ← successChance + 0.2               ▷ basic actions
    end if
    if long_term = true then
        successChance ← successChance + 0.1               ▷ action costs ≥ 4 AP
    end if
    if physical_access = true then
        successChance ← successChance − 0.2               ▷ attacker needs phys. access
    end if
    if admin = true then
        successChance ← successChance + 0.2               ▷ attacker has admin privileges
    end if
    supportTactics ← [Defense Evasion, Exfiltration, Command and Control]
    for tactic in supportTactics do
        if tactic in tactics then
            successChance ← successChance + 0.1           ▷ associated support tactics
        end if
    end for
    successChance ← successChance + (0.05 × skill)        ▷ Skill modifier
    successChance ← successChance + (0.02 × insight)      ▷ Insight modifier
    return successChance
end procedure
```

Further modification of the general tactic success probability is done via the action's Skill requirement, which is in turn based on action complexity, required user privileges, the need for user interaction, and time required (see Algorithms 5 and 6). Straightforward (low-skill) actions are 10 or 20% easier to perform. Long-term actions that cost more that one turn's available action points (≥4 AP) without time pressure are considered slightly easier, while actions requiring physical access are less likely (−20%) to succeed.

If the attacker has previously gained admin privileges or uses appropriate support actions, the chance of success is higher by a small to significant margin. The actor's overall Skill and current Insight also factor into the calculation.

Defense success chance (Algorithm 2) is handled similarly. Analysis, Monitoring, Hardening, and Deception actions are considered the baseline, while Eviction, Isolation, and Recovery is slightly easier to perform, given that these actions require prior knowledge and often rely on certain appliances. An attacker's admin privileges are considered counterproductive to implementing a defensive measure.

Algorithm 2 Defense Success Chance Calculation

```
procedure SetDefenseSuccessChance(action, skill, skill_req, long_term, insight, admin)
    successChance ← 0.5                                   ▷ Initialize success chance to 0.5
    tactics ← [Eviction, Isolation, Recovery, Analysis, Monitoring, Hardening, Deception]
    for tactic in tactics do
        switch tactic do
            case Eviction, Isolation
                successChance ← successChance + 0.2       ▷ Response type 1
            case Recovery
                successChance ← successChance + 0.1       ▷ Response type 2
            case Analysis, Monitoring
                continue                                  ▷ Detection
            case Hardening, Deception
                continue                                  ▷ Prevention
    end for
    if skill_req = 2 then
        successChance ← successChance + 0.1               ▷ simple actions
    end if
    if skill_req = 1 then
        successChance ← successChance + 0.2               ▷ basic actions
    end if
    if long_term = true then
        successChance ← successChance + 0.1               ▷ action costs ≥ 4 AP
    end if
    if admin = true then
        successChance ← successChance − 0.1               ▷ attacker has elevated priv.
    end if
    successChance ← successChance + (0.05 × skill)        ▷ Skill modifier
    successChance ← successChance + (0.02 × insight)      ▷ Insight modifier
    return successChance
end procedure
```

A.2. Detection probability

The base detection chance for infrastructures without dedicated monitoring capabilities beyond that of on-board tools is initialized at 20%. Depending on the corresponding tactic, more obvious actions (ranging from Discovery to Impact) are modified by +10 up to +30%, with stealthy actions (as per their description in MITRE ATT&CK) offsetting this to a certain degree. Obvious attacks such as file encryption double the detection chance in our model, while some actions are entirely undetectable. As with success chance, actor Skill and Insight provide bonuses, as do supporting tactics.

Defense actions have a higher base detection chance of 50%. Eviction, Isolation, and certain Recovery techniques, which are typically apparent immediately (disconnection, process termination, etc.), receive a significant detection bonus for the attacker. Analysis and Monitoring actions are typically harder to spot (−30%). Hardening and Deception measures sit at the aforementioned baseline. Skill and Insight are considered as before. Note that defense actions do not currently have

supporting techniques due to how MITRE D3FEND is translated to our game rules.

Algorithm 3 Attack Detection Chance Calculation

```
procedure SetAttackDetectionChance(action, actionType, skill, insight)
    if action is detectable then
        detectionChance ← 0.2                                   ▷ Initialize detection chance to 0.2
        tactics ← [Reconnaissance, Resource Development, Initial Access, Execution, Persistence, Privilege Escalation, Credential Access, Discovery, Lateral Movement, Collection, Impact]
        for tactic in tactics do
            switch tactic do
                case Reconnaissance
                    continue                                    ▷ Recon stage 1
                case Resource Development, Discovery
                    detectionChance ← detectionChance + 0.1     ▷ Recon stage 2
                case Initial Access, Execution, Lateral Movement
                    detectionChance ← detectionChance + 0.2     ▷ Attack stage 1
                case Persistence, Priv. Escal., Credent. Access, Collection, Impact
                    detectionChance ← detectionChance + 0.3     ▷ Attack stage 2
        end for
        if actionType = "stealthy" then
            detectionChance ← detectionChance − 0.1             ▷ stealthy actions
        end if
        if actionType = "visible" then
            detectionChance ← min(1.0, 2 × detectionChance)     ▷ obvious actions
        end if
        supportTactics ← [Defense Evasion, Exfiltration, Command and Control]
        for tactic in supportTactics do
            if tactic in tactics then
                detectionChance ← detectionChance + 0.1         ▷ support tactics
            end if
        end for
        detectionChance ← detectionChance + (0.05 × skill)      ▷ Skill modifier
        detectionChance ← detectionChance + (0.02 × insight)    ▷ Insight modifier
        return detectionChance
    else
        detectionChance ← 0
        return detectionChance
    end if
end procedure
```

Algorithm 4 Defense Detection Chance Calculation

```
procedure SetDefenseDetectionChance(action, actionType, skill, insight)
    detectionChance ← 0.5                                       ▷ Initialize detection chance to 0.5
    tactics ← [Eviction, Isolation, Recovery, Analysis, Monitoring, Hardening, Deception]
    for tactic in tactics do
        switch tactic do
            case Eviction, Isolation
                detectionChance ← detectionChance + 0.3         ▷ Response type 1
            case Recovery
                detectionChance ← detectionChance + 0.2         ▷ Response type 2
            case Analysis, Monitoring
                detectionChance ← detectionChance − 0.3         ▷ Detection
            case Hardening, Deception
                continue                                        ▷ Prevention
    end for
    if actionType = "stealthy" then
        detectionChance ← detectionChance − 0.1                 ▷ stealthy actions
    end if
    if actionType = "visible" then
        detectionChance ← min(1.0, 2 × detectionChance)         ▷ visible actions
    end if
    detectionChance ← detectionChance + (0.05 × skill)          ▷ Skill modifier
    detectionChance ← detectionChance + (0.02 × insight)        ▷ Insight modifier
    return detectionChance
end procedure
```

A.3. Skill requirement

We derive the value that determines the level of required skill (ranging from 1 to 5) from factors similar to those used in the CVSS base score: Attack complexity (low, medium, or high), the required level of local permissions (user, admin/root, or system/kernel), and the level of user interaction required to successfully conduct an attack. In addition, we define actions early in the kill chain as generally easier to conduct, with untargeted actions (like OSINT gathering) as being most trivial.

Algorithm 5 Attack Skill Requirement Calculation

```
procedure SetAttackSkillRequirement(complexity, target, permissions, interaction)
    skillRequirement ← 1                                        ▷ Initialize skill requirement to 1
    stages ← [Reconnaissance, InitialAccess, Execution]
    for stage in stages do
        switch stage do
            case Reconnaissance                                 ▷ Recon actions
                if target = none then
                    skillRequirement ← skillRequirement         ▷ untargeted actions
                else
                    skillRequirement ← skillRequirement + 1     ▷ targeted actions
                end if
            case InitialAccess, Execution
                skillRequirement ← skillRequirement + 1         ▷ Access & Execution
    end for
    switch complexity do
        case complexity = "low"
            skillRequirement ← skillRequirement                 ▷ simple actions
        case complexity = "medium"
            skillRequirement ← skillRequirement + 1             ▷ average actions
        case complexity = "high"
            skillRequirement ← skillRequirement + 2             ▷ complex actions
    switch permissions do
        case permissions = "user"
            skillRequirement ← skillRequirement − 1             ▷ user privileges
        case permissions = "admin"
            skillRequirement ← skillRequirement                 ▷ admin privileges
        case permissions = "system"
            skillRequirement ← skillRequirement + 1             ▷ system privileges
    if interaction = true then
        skillRequirement ← skillRequirement − 1                 ▷ user interaction required
    end if
    return skillRequirement
end procedure
```

Algorithm 6 Defense Skill Requirement Calculation

```
procedure SetDefenseSkillRequirement(complexity, long_term)
    skillRequirement ← 1                                        ▷ Initialize skill requirement to 1
    types ← [Detection, Prevention, Response]
    for type in types do
        switch type do
            case Detection
                skillRequirement ← skillRequirement             ▷ Detection actions
            case Prevention, Response
                skillRequirement ← skillRequirement + 1         ▷ Prev. & Resp. actions
    end for
    switch complexity do
        case complexity = "low"
            skillRequirement ← skillRequirement                 ▷ simple actions
        case complexity = "medium"
            skillRequirement ← skillRequirement + 1             ▷ average actions
        case complexity = "high"
            skillRequirement ← skillRequirement + 2             ▷ complex actions
    if long_term = true then
        skillRequirement ← skillRequirement − 1                 ▷ action costs ≥ 4 AP
    end if
    return skillRequirement
end procedure
```

A.4. Impact values

If the action is based on CAPEC and comes with both a 'Consequences' field (listing a C, I, and/or A scope) and a 'Typical Severity' (from low, medium, to high), we adopt this value as-is.

For ATT&CK-type actions, the attack action's stage determines the baseline damage value, which is initialized to 1 or 2. A particularly low or high skill requirement (see Algorithm 5) may adjust this value up or down. We also introduced an "applicability" rating (ranging from 0 – not applicable, to 3 – fully applicable) that measures how well suited an action is to deal damage in regard to one of the three CIA tracks. This value, like the aforementioned 'Typical Severity' rating in CAPEC,

is derived from domain knowledge provided by contributing security experts.

We consider a damage track applicable to an action if either CAPEC or ATT&CK list it in their 'Consequences' or 'Impact Type' fields, respectively.

Algorithm 7 Damage Impact Calculation

```
procedure ConfidentialityDamageImpact(skillRequirement, applicability)
    damage ← 0                                              ▷ Initialize damage to 0
    if applicability > 0 then
        stages ← [Reconnaissance, InitialAccess, Execution]
        for stage in stages do
            switch stage do
                case Reconnaissance                         ▷ Recon actions
                    damage ← damage
                case InitialAccess
                    damage ← damage + 1                     ▷ Access actions
                case Execution
                    damage ← damage + 2                     ▷ Execution actions
        end for
        switch skillRequirement do
            case skillRequirement = 1                       ▷ low skill actions
                damage ← damage − 1
            case skillRequirement = 2 or 3                  ▷ medium skill actions
                damage ← damage
            case skillRequirement = 4 or 5                  ▷ high skill actions
                damage ← damage + 1
        switch applicability do
            case applicability = 1                          ▷ barely applicable
                damage ← damage − 2
            case applicability = 2                          ▷ moderately applicable
                damage ← damage − 1
            case applicability = 3                          ▷ fully applicable
                damage ← damage
        return damage
    else
        return damage                                       ▷ not applicable (no damage)
    end if
end procedure
procedure IntegrityDamageImpact(skillRequirement)           ▷ see above
end procedure
procedure AvailabilityDamageImpact(skillRequirement)        ▷ see above
end procedure
```

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cose.2024.104287.

PenQuest can be accessed via https://play.pen.quest/. Please contact the authors for an account.

Data availability

Data will be made available on request.

References

Abdallah, M., Cason, T., Bagchi, S., Sundaram, S., 2020. The effect of behavioral probability weighting in a sequential defender-attacker game. In: 2020 59th IEEE Conference on Decision and Control. CDC, pp. 3255–3260. http://dx.doi.org/10.1109/CDC42340.2020.9304311.

Abt, C., 1987. Serious Games. University Press of America, Maryland, U.S..

Bartle, R., 1996. Hearts, clubs, diamonds, spades: Players who suit MUDs.

Batzos, Z., Saoulidis, T., Margounakis, D., Fountoukidis, E., Grigoriou, E., Moukoulis, A., Sarigiannidis, A., Liatifis, A., Karipidis, P., Bibi, S., Filippidis, A., Kazanidis, I., Nifakos, S., Kasig, T., Heydari, M., Mouratidis, H., 2023. Gamification and serious games for cybersecurity awareness and first responders training: An overview. http://dx.doi.org/10.36227/techrxiv.22650952.v1.

Black Hills Information Security, 2024. Backdoors & Breaches. URL ttps://www.blackhillsinfosec.com/projects/backdoorsandbreaches/. (Accessed 19 April 2024).

Broniatowski, D., 2021. Psychological foundations of explainability and interpretability in artificial intelligence. http://dx.doi.org/10.6028/NIST.IR.8367.

Bundesamt für Sicherheit in der Informationstechnik (BSI), 2023. IT-Grundschutz-Kompendium. URL https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2023.html.

Casey, E., Back, G., Barnum, S., 2015. Leveraging CybOX to standardize representation and exchange of digital forensic information. Digit. Investig. 12, S102–S110. http://dx.doi.org/10.1016/j.diin.2015.01.014, URL https://www.sciencedirect.com/science/article/pii/S1742287615000158, DFRWS 2015 Europe.

Cybersecurity and Infrastructure Security Agency (CISA), 2023. FY22 risk and vulnerability assessment analysis. URL https://www.cisa.gov/sites/default/files/2023-07/FY22-RVA-Analysis%20-%20Final_508c.pdf.

Cybersecurityventures.com, 2024. Global Ransomware Damage Costs Predicted To Exceed $265 Billion By 2031. URL https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/.

Decisions, I., 2024. The Weakest Link. URL https://www.isdecisions.com/user-security-awareness-game/. (Accessed 19 April 2024).

Defense Information Systems Agency, 2024. Cyber Awareness Challenge 2022. URL https://public.cyber.mil/training/cyber-awareness-challenge/. (Accessed 19 April 2024).

Gaurav, D., Kaushik, Y., Supraja, S., Khandelwal, A., Negi, K., Gupta, M.P., Chaturvedi, M., 2021. Cybersecurity training for web applications through serious games. In: 2021 IEEE International Conference on Engineering, Technology & Education. TALE, IEEE, pp. 390–398.

Google Inc, 2024. Jigsaw Phishing Quiz. URL https://phishingquiz.withgoogle.com/. (Accessed 19 April 2024).

Grafenauer, T., König, S., Rass, S., Schauer, S., 2018. A simulation tool for cascading effects in interdependent critical infrastructures. In: ARES 2018 Proceedings of the 13th International Conference on Availability, Reliability and Security. ARES Conference ; Conference date: 27-08-2018 Through 30-08-2018.

Growth Engineering, 2023. The ultimate definition of gamification. URL https://www.growthengineering.co.uk/definition-of-gamification/. (Accessed 21 July 2023).

Hahn, A., Thomas, R.K., Lozano, I., Cardenas, A., 2015. A multi-layered and kill-chain based security analysis framework for cyber-physical systems. Int. J. Crit. Infrastruct. Prot. 11, 39–50.

Halgamuge, M.N., 2022. Estimation of the success probability of a malicious attacker on blockchain-based edge network. Comput. Netw. 219, 109402. http://dx.doi.org/10.1016/j.comnet.2022.109402, URL https://www.sciencedirect.com/science/article/pii/S1389128622004364.

Hart, S., Margheri, A., Paci, F., Sassone, V., 2020. Riskio: A serious game for cyber security awareness and education. Comput. Secur. 95, 101827.

Hemberg, E., Turner, M.J., Rutar, N., O'reilly, U.-M., 2024. Enhancements to threat, vulnerability, and mitigation knowledge for cyber analytics, hunting, and simulations. Digit. Threats 5 (1), http://dx.doi.org/10.1145/3615668.

Holzinger, A., Kargl, M., Kipperer, B., Regitnig, P., Plass, M., Müller, H., 2022. Personas for artificial intelligence (AI) An open source toolbox. IEEE Access 10, 23732–23747. http://dx.doi.org/10.1109/ACCESS.2022.3154776.

Holzinger, A., Mueller, H., 2021. Toward human-AI interfaces to support explainability and causability in medical AI. IEEE Comput. 54 (10), 78–86. http://dx.doi.org/10.1109/MC.2021.3092610.

International Organization for Standardization, 2023. ISO/IEC 27002:2022: Information security, cybersecurity and privacy protection — Information security controls. URL https://www.iso.org/standard/75652.html. (Accessed 20 July 2023).

Jaffray, A., Finn, C., Nurse, J.R., 2021. SherLOCKED: A serious game for cybersecurity education. arXiv URL https://arxiv.org/pdf/2107.04506.

Kapp, K.M., 2012. The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education. Pfeiffer, San Francisco, CA, URL https://doi.org/10.1002/9781119208897.

Kohnfelder, L., Praerit, G., 1999. The threats to our products. In: Microsoft Interface.

LaPiedra, J., 2000. The Information Security Process Prevention, Detection and Response. SANS Institute.

LivingSecurity, 2024a. Craft the Phish. URL https://phishing.livingsecurity.com/. (Accessed 19 April 2024).

LivingSecurity, 2024b. Hotspot. URL https://hotspot.livingsecurity.com/. (Accessed 19 April 2024).

LivingSecurity, 2024c. Living Security Teams: CyberEscape Online. URL https://www.livingsecurity.com/cyberescape-online/. (Accessed 19 April 2024).

Luh, R., Eresheim, S., Größbacher, S., Petelin, T., Mayr, F., Tavolato, P., Schrittwieser, S., 2022. PenQuest reloaded: A digital cyber defense game for technical education. In: 2022 IEEE Global Engineering Education Conference. EDUCON, pp. 906–914. http://dx.doi.org/10.1109/EDUCON52537.2022.9766700.

Luh, R., Temper, M., Tjoa, S., Schrittwieser, S., Janicke, H., 2020. PenQuest: a gamified attacker/defender meta model for cyber security assessment and education. J. Comput. Virol. Hack. Tech. 16, 19–61. http://dx.doi.org/10.1007/s11416-019-00342-x, URL http://eprints.cs.univie.ac.at/6681/.

Manos, A., Faragallah, O.S., 2019. Development of serious games for teaching information security. In: IEEE International Conference on Serious Games. URL https://ieeexplore.ieee.org/iel7/6287639/8600701/08911357.pdf.

MITRE Corporation, 2023a. Common Attack Pattern Enumerations and Classifications. URL https://capec.mitre.org/. (Accessed 20 July 2023).

MITRE Corporation, 2023b. Common Vulnerability Enumeration. URL https://www.cve.org/. (Accessed 20 July 2023).

MITRE Corporation, 2023c. Common Weakness Enumeration. URL https://cwe.mitre.org/. (Accessed 20 July 2023).

MITRE Corporation, 2023d. MITRE ATT&CK. URL https://attack.mitre.org/. (Accessed 20 July 2023).
MITRE Corporation, 2023e. MITRE Cyber Analytics Repository. URL https://car.mitre.org/. (Accessed 20 July 2023).
MITRE Corporation, 2023f. MITRE D3FEND. URL https://d3fend.mitre.org/. (Accessed 20 July 2023).
MITRE Corporation, 2023g. MITRE Engage. URL https://engage.mitre.org/. (Accessed 20 July 2023).
MITRE Corporation, 2023h. NIST SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations. URL https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final. (Accessed 20 July 2023).
Natalucci, F., Qureshi, M.S., Suntheim, F., 2024. Rising Cyber Threats Pose Serious Concerns for Financial Stability. URL https://www.imf.org/en/Blogs/Articles/2024/04/09/rising-cyber-threats-pose-serious-concerns-for-financial-stability.
OASIS Cyber Threat Intelligence Technical Committee, 2023. Introduction to STIX. URL https://oasis-open.github.io/cti-documentation/stix/intro.html. (Accessed 20 April 2023).
Open Web Application Security Project, 2023a. OWASP Application Security Verification Standard. URL https://owasp.org/www-project-application-security-verification-standard/. (Accessed 26 April 2023).
Open Web Application Security Project, 2023b. OWASP Top 10:2021. URL https://owasp.org/Top10/. (Accessed 26 April 2023).
OWASP Foundation, 2024. OWASP Cornucopia. URL https://owasp.org/www-project-cornucopia/. (Accessed 19 April 2024).
Palleti, V.R., Adepu, S., Mishra, V.K., Mathur, A., 2021. Cascading effects of cyber-attacks on interconnected critical infrastructure. Cybersecurity 4, 1–19.
Prümmer, J., van Steen, T., van den Berg, B., 2024. A systematic review of current cybersecurity training methods. Comput. Secur. 136, 103585. http://dx.doi.org/10.1016/j.cose.2023.103585, URL https://www.sciencedirect.com/science/article/pii/S0167404823004959.
Rajbhandari, L., Snekkenes, E., 2018. Utilizing game theory for security risk assessment. In: Game Theory for Security and Risk Management: From Theory To Practice. Springer International Publishing, Cham, pp. 3–19. http://dx.doi.org/10.1007/978-3-319-75268-6_1.
Rak, M., Moretta, F., Granata, D., 2024. Advancing ESSecA: a step forward in Automated Penetration Testing. In: Proceedings of the 19th International Conference on Availability, Reliability and Security. ARES '24, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/3664476.3670459.
Rebah, H.B., et al., 2019. The educational effectiveness of serious games. Médiat. Médiat. (2), 131–155.
Retzlaff, C.O., Das, S., Wayllace, C., Mousavi, P., Afshari, M., Yang, T., Saranti, A., Angerschmid, A., Taylor, M.E., Holzinger, A., 2024. Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities. J. Artif. Intell. Res. (JAIR) 79 (1), 349–415. http://dx.doi.org/10.1613/jair.1.15348.
Rieb, A., Lechner, U., 2016. Operation digital chameleon: towards an open cybersecurity method. In: Proceedings of the 12th International Symposium on Open Collaboration. pp. 1–10.
Shostack, A., 2014. Elevation of privilege: Drawing developers into threat modeling. In: 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14).
Shostack, A., 2024. Tabletop Security Games & Cards. URL https://adam.shostack.org/games.html. (Accessed 19 April 2024).
Standen, M., Bowman, D., Hoang, S., Richer, T., Lucas, M., Van Tassel, R., Vu, P., Kiely, M., Konschnik, N., Collyer, J., 2022. Cyber operations research Gym. https://github.com/cage-challenge/CybORG.
Stoneburner, G., Goguen, A.Y., Feringa, A., 2002. SP 800-30. Risk Management Guide for Information Technology Systems. Technical Report, NIST.
Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction, second ed. MIT Press, Cambridge, MA.
Swiderski, F., Snyder, W., 2004. Threat modeling.
Tang, Y., Sun, J., Wang, H., Deng, J., Tong, L., Xu, W., 2024. A method of network attack-defense game and collaborative defense decision-making based on hierarchical multi-agent reinforcement learning. Comput. Secur. 142, 103871. http://dx.doi.org/10.1016/j.cose.2024.103871, URL https://www.sciencedirect.com/science/article/pii/S016740482400172X.
Tavolato, P., Luh, R., Eresheim, S., Gmeiner, S., Schrittwieser, S., 2024. Quantifying the odds in real world attack scenarios. In: 2024 IEEE International Conference on Cyber Security and Resilience (CSR). pp. 845–852, URL https://api.semanticscholar.org/CorpusID:272858582.
Tayouri, D., Baum, N., Shabtai, A., Puzis, R., 2023. A survey of MulVAL extensions and their attack scenarios coverage. IEEE Access 11, 27974–27991. http://dx.doi.org/10.1109/ACCESS.2023.3257721.
Technology.org, 2024. Global Costs of Cyber Attacks in 2024 and The Future. URL https://www.technology.org/2024/02/13/global-costs-of-cyberattacks-in-2024/.
Texas A&M University, 2024. Keep Tradition Secure. URL https://keeptraditionsecure.tamu.edu/. (Accessed 19 April 2024).
Trend Micro, 2024. Targeted Attack: The Game. URL http://targetedattacks.trendmicro.com/. (Accessed 19 April 2024).
van Dam, T., Bakkes, S., 2019. The ACE2 Model: Refining Bartle's Player Taxonomy for Creation Play.
Vu, A.H., Tippenhauer, N.O., Chen, B., Nicol, D.M., Kalbarczyk, Z., 2014. CyberSAGE: A tool for automatic security assessment of cyber-physical systems. In: Norman, G., Sanders, W. (Eds.), Quantitative Evaluation of Systems. Springer International Publishing, Cham, pp. 384–387.
Wen, Z.A., Li, Y., Wade, R., Huang, J., Wang, A., 2017. What. hack: Learn phishing email defence the fun way. In: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems. pp. 234–237.
Wilkinson, P., 2016. A Brief History of Serious Games, vol. 9970, International GI-Dagstuhl Seminar, Wadern, Germany, pp. 17–41. http://dx.doi.org/10.1007/978-3-319-46152-6_2.
World Economic Forum, 2024. Global Cybersecurity Outlook 2024. URL https://www.weforum.org/publications/global-cybersecurity-outlook-2024/.
Yee, N., 2007. Motivations of Play in MMORPGs: Results from a Factor Analytic Approach. CyberPsychol. Behav. 9 (6), 772–775. http://dx.doi.org/10.1089/cpb.2006.9.772.
Zhang, Y., Paxson, V., 2000. Detecting stepping stones. In: USENIX Security Symposium. Vol. 171, p. 184.

Robert Luh is a lecturer and researcher at St. Pölten University of Applied Sciences, where he also serves as the head of the IT Security bachelor's program in the Department of Computer Science & Security. He earned his Ph.D. in Cyber Security from De Montfort University, Leicester, in 2019, focusing on machine learning-enabled intrusion detection and gamified security modeling. His research interests encompass digital forensics, serious games, and risk assessment.

Sebastian Eresheim is a Junior Researcher at St. Pölten University of Applied Sciences, where he graduated in 2018. In addition to earning a Master's degree in Information Security, he studied Technical Mathematics at the Vienna University of Technology. His research focuses primarily on Machine Learning and Artificial Intelligence, with a particular emphasis on Reinforcement Learning. Sebastian's work aims to enhance the applicability of these algorithms across a broad range of domains.

Paul Tavolato: After earning a Ph.D. in Computer Science from the Vienna University of Technology, he spent several years teaching and conducting research at the university before transitioning to industry, where he gained 20 years of experience. In 2010, Paul returned to academia, focusing on cybersecurity research, with expertise in malware analysis, anomaly detection, and AI applications in industrial control systems, smart grids, and weather forecasting. Since 2020, he has been affiliated with the University of Vienna, working on a research project that explores the application of serious gaming and game theory to threat and risk analysis.

Thomas Petelin completed his Bachelor's degree in Data Engineering & Statistics and his Master's degree in Computational Intelligence at Vienna University of Technology in 2014. He has over 10 years of experience as a Fullstack Developer, specializing in microservice-based backend systems. His research interests include distributed systems, enterprise computing, and software development.

Simon Gmeiner received his Bachelor's degree in Data Science from St. Pölten University of Applied Sciences in 2024, during which he also worked as a student employee at the University of Vienna. His research focuses on machine learning and artificial intelligence, with a particular emphasis on reinforcement learning.

Andreas Holzinger is a full professor of digital transformation at BOKU University Vienna and a pioneer in interactive machine learning with a focus on the human-in-the-loop paradigm. He advocates for a synergistic approach to Human-Centered AI (HCAI), aiming to empower humans by aligning AI systems with human intelligence, social values, ethical principles, and legal requirements to ensure safe and secure AI. He earned his Ph.D. in Cognitive Science from Graz University in 1998 and a second Ph.D. in Computer Science from Graz University of Technology in 2003. Andreas has served as a Visiting Professor at institutions including the University of Verona, RWTH Aachen University, University College London, and the Alberta Machine Intelligence Institute in Edmonton.

Sebastian Schrittwieser earned his Ph.D. in Technical Sciences with a focus on Information Security from the Vienna University of Technology in 2014. From 2015 to 2020, he headed the Josef Ressel Center for Unified Threat Intelligence on Targeted Attacks. In April 2024, he assumed leadership of the newly established Christian Doppler Laboratory for Assurance and Transparency in Software Protection at the University of Vienna, within the Security & Privacy research group. His current research interests include software protection mechanisms and the security of large language models (LLMs).