Cole-Puli International Conference Publishing Group

Business and Social Sciences Proceedings

Article

2025 International Conference on Education, Economic Management, Law and Humanities and Social Sciences (MELSS 2025)

Design and Utility of a Graphical User Interface for Hierarchical Attack Representation Models

Ruoqi Guo [1,*]

[1] School of Electrical Engineering & Computer Science, The University of Queensland, UQ Brisbane, QLD 4072, Australia

* Correspondence: Ruoqi Guo, School of Electrical Engineering & Computer Science, The University of Queensland, UQ Brisbane, QLD 4072, Australia

Abstract: The increasing complexity of cyber threats poses significant cognitive challenges for security analysts and creates communication barriers between technical experts and non-technical stakeholders. While graphical security models like the Hierarchical Attack Representation Model (HARM) offer a scalable solution for analysis, their practical utility is often hindered by the lack of intuitive interfaces. This paper presents the design, implementation, and evaluation of a novel web-based Graphical User Interface (GUI) for HARM, built to enhance network security analysis through effective visualization. Grounded in human-computer interaction (HCI) principles, the interface integrates the HARM model with the Harmat analysis engine, allowing users to interactively build, visualize, and analyze multi-layered attack paths. We detail the system's architecture and key design choices, such as the dual-layer canvas for attack graphs and attack trees, visual iconography, and a logical layout aimed at reducing cognitive load. Furthermore, we discuss the broader implications of this tool beyond technical analysis, exploring its potential as an educational platform for cybersecurity training and as a communication medium to facilitate risk-based decision-making in organizational contexts. The results demonstrate that a well-designed visual interface not only improves the efficiency of security analysis but also makes complex security concepts more accessible to a wider audience.

Keywords: cybersecurity visualization; human-computer interaction; graphical security models; HARM; attack graph; cybersecurity education; visual analytics

Received: 18 November 2025
Revised: 05 January 2026
Accepted: 16 January 2026
Published: 19 January 2026

1. Introduction

With the rapid proliferation of information technology, the Internet has become deeply integrated into every facet of social and economic life, fundamentally altering human production and daily activities [1]. This pervasive digitization, while offering unprecedented convenience, has also exposed individuals and organizations to an increasingly complex and evolving landscape of cyber threats. The economic ramifications are staggering, with global costs of cybercrime estimated to be in the hundreds of billions of dollars annually and projected to rise significantly [2]. Consequently, developing innovative and effective technologies to enhance network security has become a paramount task in the modern era. However, the challenge of cybersecurity extends beyond purely technical defenses. As network systems grow in scale and complexity, security analysts face significant cognitive overload when trying to comprehend vast amounts of data to identify potential threats. Traditional text-based

reports and raw data logs are often insufficient for rapid and intuitive risk assessment. This creates a critical communication gap between technical experts, who understand the intricate details of vulnerabilities, and non-technical decision-makers, such as managers and executives, who must allocate resources for security investments [3]. There is a clear need for methods that can bridge this gap by making complex security information more understandable and actionable. Visualization offers a powerful solution to this challenge. Graphical Security Models (GSMs) have emerged as a class of tools that use visual formalisms to represent and analyze key concepts in system security [4]. By abstracting complex security scenarios into intuitive graphical elements, GSMs facilitate clearer communication among stakeholders and enable a more systematic analysis of potential attack vectors. This visual approach helps security personnel and developers to identify, classify, and quantify threats more effectively compared to purely textual descriptions. Among the various GSMs, traditional models like Attack Graphs (AGs) and Attack Trees (ATs) have been widely used [5]. However, they often encounter a significant scalability problem, known as state-space explosion, when applied to large-scale enterprise networks, making them computationally inefficient and difficult to manage [6,7]. To address this limitation, the Hierarchical Attack Representation Model (HARM) was proposed as an innovative, multi-layered approach [8,9]. HARM ingeniously separates the security model into two layers: an upper-layer AG that models network-level reachability and a lower-layer AT that details the vulnerability information for each individual host. This hierarchical design dramatically improves the model's scalability and manageability in complex environments. Despite the theoretical elegance and scalability of the HARM model, its practical utility can be severely hindered without an intuitive and user-friendly interface. A powerful analytical model remains inaccessible to many potential users if it requires extensive technical expertise to operate via command-line interfaces. This work addresses this crucial gap by presenting the design and implementation of a graphical user interface (GUI) specifically for the HARM model. The primary contributions of this work are threefold:

The design and implementation of a functional and interactive GUI that allows users to visually construct, modify, and analyze HARM-based security models.

The integration of core Human-Computer Interaction (HCI) principles into the interface design to enhance usability, reduce cognitive load, and create a more intuitive user experience.

An exploration of the tool's broader applications beyond technical analysis, discussing its potential value as an educational platform for cybersecurity training and as a communication medium for organizational risk management.

The remainder of this research is structured as follows. Section 2 reviews the background of GSMs, the HARM model, and the role of visualization in cybersecurity. Section 3 details the system design methodology and key features of the implemented GUI. Section 4 discusses the implications of the tool from an HCI perspective and its potential applications in education and organizational communication. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Background and Related Work a Comprehensive Understanding of Our Work Requires Knowledge of Graphical Security Modeling, the Specifics of the HARM Framework, and the Broader Context of Visualization in the Cybersecurity Domain

2.1. Graphical Security Models

Graphical Security Models are formal, visual tools used to systematically describe and analyze potential security threats to a system. They provide a structured way to reason about how an attacker might compromise a target. The two most foundational types of GSMs are ATs and AGs.

An Attack Tree (AT), first conceptualized by Schneier, is a top-down, hierarchical model that deconstructs an overall attack goal into a series of smaller, more manageable

sub-goals [10]. The main goal is the root of the tree, and the various means to achieve it are represented as child nodes, or leaves. These nodes are connected by logical AND/OR gates, indicating whether multiple steps must be performed in combination or if alternative paths exist [11]. ATs are highly effective for modeling and quantifying the effort required for specific attack scenarios on a single target [12].

An Attack Graph (AG), in contrast, focuses on modeling how an attacker can chain together multiple vulnerabilities across a network to achieve an objective [13]. In an AG, nodes represent system states like attacker privileges on a host, and directed edges represent actions that transition the system from one state to another. AGs are powerful for providing a holistic view of all possible multi-step attack paths within a network environment [14]. However, their primary drawback is a lack of scalability. As the size of the network and the number of vulnerabilities grow, the number of states and paths in the AG can increase exponentially, leading to a "state-space explosion" that makes the graph too large to generate or analyze effectively [15].

2.2. The Hierarchical Attack Representation Model (HARM)

To overcome the scalability limitations of traditional AGs while retaining their analytical power, Hong and Kim proposed the HARM [8,9]. HARM achieves scalability by abstracting the security model into a two-layer hierarchy, which significantly improves performance and manageability for large networks [16].

The upper layer of HARM consists of an Attack Graph that models only the network-level connectivity or reachability between hosts [17]. In this view, each node represents a host, and an edge from Host A to Host B signifies that an attacker who has compromised Host A can reach and potentially attack Host B. This layer provides a macro-level overview of the network topology from an attacker's perspective.

The lower layer uses Attack Trees to model the specific vulnerabilities present on each individual host. For every host node in the upper-layer AG, there is a corresponding AT in the lower layer [18]. This AT details how an attacker could gain control of that host by exploiting one or more local vulnerabilities, using logical AND/OR gates to represent the conditions for a successful compromise.

By decoupling network topology from host-specific vulnerability details, HARM effectively contains the complexity [19]. The size of the upper-layer AG grows linearly with the number of hosts, while the complexity of vulnerability analysis is encapsulated within individual, manageable ATs. This modular and hierarchical structure makes HARM a highly scalable and powerful framework for modern network security analysis.

2.3. Visualization in Cybersecurity

The effective visualization of security data is crucial for transforming raw information into actionable intelligence. As noted by Fink et al., well-designed visual workspaces can significantly improve the efficiency of security analysts by providing an intuitive platform for monitoring and managing security data [20]. Visualization tools help analysts to identify patterns, detect anomalies, and comprehend complex relationships that would be nearly impossible to discern from text-based logs alone. This has led to a growing interest in creating visual analytics tools for various cybersecurity tasks.

In the context of graphical security models, several visualization tools have been developed. For example, the Safelite framework was introduced to automate security analysis, and its accompanying GUI, Safeview, was designed to visualize the analysis results generated from models like AGs and HARMs [6]. These tools demonstrate the value of providing a visual front-end to complex analytical engines. However, the field is continuously evolving, with an ongoing need to create tools that are not only functionally powerful but also grounded in principles of user-centered design to maximize usability and accessibility [17]. This work builds upon this foundation by focusing on creating a

highly interactive and intuitive interface tailored specifically for the construction and analysis of the HARM model, with the broader goal of making sophisticated security analysis accessible to a wider range of users, including trainees and non-technical stakeholders.

3. System Design and Methodology

This chapter details the methodology employed in designing and implementing the interactive graphical user interface for the HARM. The process was guided by a set of user-centric design goals aimed at transforming the theoretical HARM framework into a practical and accessible analytical tool. We describe the system's architecture, key interface features, and the design decisions made to enhance usability and analytical efficiency.

3.1. Design Goals and Principles

The development of the GUI was driven by a series of core principles derived from both functional and non-functional requirements, ensuring the final tool is both powerful and easy to use. These guiding principles are:

1. Intuitive Interaction

The primary goal was to enable users to create, view, and modify complex HARM structures through direct and intuitive graphical manipulation. This principle eschews command-line dependency in favor of a point-and-click environment, lowering the barrier to entry for users who may not be security modeling experts.

2. Hierarchical Visualization

The interface must faithfully represent the dual-layer nature of the HARM model. It should provide a clear and seamless way for users to navigate between the high-level network topology and the detailed, host-specific vulnerability information, thereby reducing cognitive load by separating distinct analytical contexts.

Integrated Analysis and Feedback

The GUI should be tightly integrated with a backend analysis engine to provide on-demand risk assessment. Users must be able to trigger complex calculations with a single action and receive immediate, understandable feedback in the form of analytical reports and operational logs. This aligns with the HCI principle of system status visibility [21].

4. Data Persistence and Portability

To support long-term projects, iterative analysis, and team collaboration, the system must allow users to save their work and reload it in subsequent sessions. The design also considered cross-platform compatibility to ensure a consistent user experience across different operating systems, as demonstrated by the use of libraries intended to function on macOS, Windows, and Linux environments [22].

3.2. System Architecture

The system is built upon a two-part architecture including a front-end GUI and a back-end analysis engine. This decoupled design allows for modularity and clear separation of concerns.

The front-end GUI is developed using Python's native Tkinter library, chosen for its cross-platform capabilities. The GUI is responsible for rendering all visual elements, including the canvases, nodes, arcs, and control buttons. It captures all user interactions, such as mouse clicks and menu selections, and translates them into commands for the back-end.

The back-end analysis engine is powered by Harmat, a dedicated Python library for HARM analysis. During the initialization of the GUI, corresponding Harmat data structures are instantiated. When a user performs an action in the GUI, such as adding a host node, the front-end calls the appropriate Harmat function to create and store a corresponding Host object in the back-end data model. This ensures that the visual

representation on the canvas is always synchronized with the underlying analytical model. The analysis process itself is triggered by the GUI but executed entirely by the Harmat engine's flowup() method, with the results passed back to the front-end for display.

3.3. Interface Implementation and Key Features

The GUI's design directly reflects the principles outlined above, featuring several key components that work in concert to provide a fluid user experience.

1.The Dual-Canvas Interface

The core of the user experience is a dual-canvas design that mirrors the HARM structure. The main interface (see Figure 1) presents the top-level Attack Graph, where users construct the overall network topology by placing Attacker, Host, and Target nodes. By right-clicking a Host node, the user can open a secondary, low-level interface (see Figure 2). This canvas is dedicated to building the Attack Tree for that specific host, allowing the user to add vulnerability nodes and define their relationships using logical AND/OR gates. This separation allows users to focus on either the macro-level network flow or the micro-level host compromise without visual clutter.

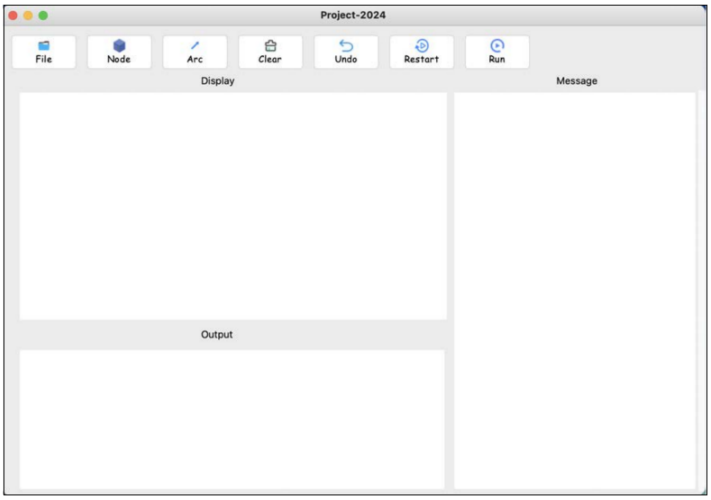

Figure 1. The main interface.

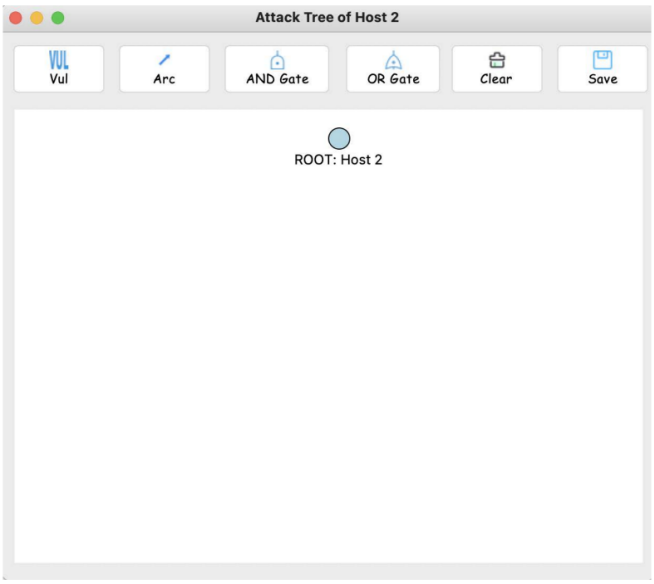

Figure 2. The low-level interface.

2. Interactive Model Construction

Users build the security model through direct manipulation. The interface features a mode-based system where users can switch between "Node Mode" for adding graphical elements and "Arc Mode" for connecting them. For instance, in "Node Mode," a click on the canvas creates a node, while in "Arc Mode," clicking on two sequential nodes draws a directed edge representing a potential attack step. This tactile, interactive process makes the abstract task of security modeling a concrete and visual activity.

3.Visual Enhancements for Usability:

Several features were implemented specifically to improve the interface's clarity and usability. To aid in quick identification, different node types such as Host, Attacker, Target and Vulnerability, are rendered with distinct visual icons and colors (see Figure 3). This use of visual encoding reduces the user's cognitive effort in parsing the graph. Furthermore, the layout of the operational buttons and information panels was logically organized to create a more intuitive workflow, placing frequently used functions in easily accessible locations.

Figure 3. Icons.

4.Integrated Analysis and Feedback Mechanism

A "Run" button serves as the trigger for the entire analysis process. Upon being clicked, the GUI passes the constructed HARM data structure to the Harmat engine. The engine performs the risk calculation, and the results are then displayed in a dedicated text area within the GUI (see Figure 4). This provides immediate feedback on the potential attack paths and their risk scores. In parallel, a real-time operation log records every action taken by the user. This log not only serves as a history tracker but also supports an "Undo" function, giving users the freedom to experiment and correct mistakes easily.

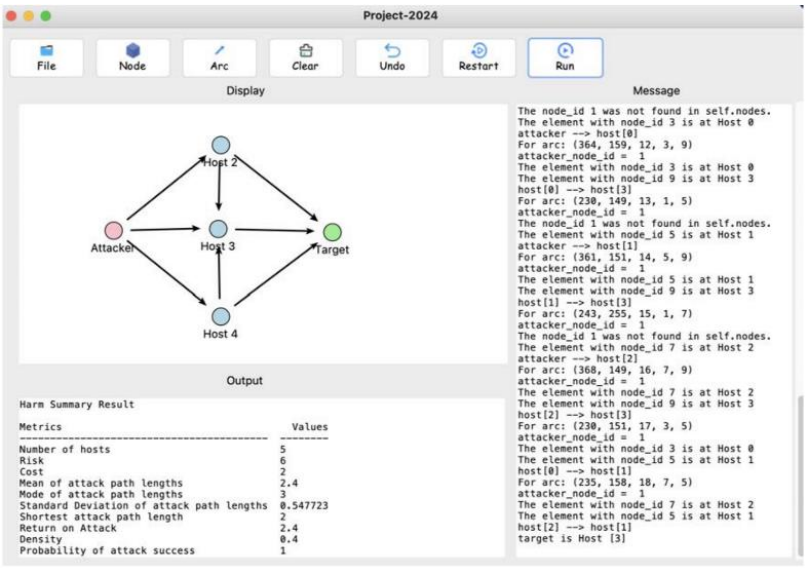

Figure 4. The result interface.

5.File Operations for Data Persistence

The system includes "Save" and "Load" functionalities, which are crucial for practical use. This feature utilizes Python's pickle module to serialize the entire state of the HARM model, including all nodes, arcs, attributes, and their positions on the canvas into a single file. Users can then save their progress and reload a complex attack graph at a later time, facilitating iterative analysis and the sharing of models among team members [23].

Through this combination of a principled design approach and key interactive features, the developed GUI successfully translates the powerful but abstract HARM model into a tangible and highly usable tool for network security analysis.

4. Discussion

The development of the HARM GUI is more than a technical implementation; it represents an effort to make complex cybersecurity analysis more accessible, intuitive, and effective. This chapter discusses the broader implications of our work by examining it through the lens of HCI principles. Furthermore, we explore its potential applications beyond traditional security analysis, specifically as a tool for education and as a medium for improving risk communication within organizational contexts. Finally, we address the current limitations of the system and propose directions for future research.

4.1. HCI Principles in Security Visualization

The usability of a security tool is as critical as its analytical power. A tool that is difficult to use will not be adopted, regardless of its sophistication. The design of our GUI was implicitly guided by several core HCI principles, which explains its potential for enhanced usability.

1.Visibility of System Status

The interface is designed to keep the user constantly informed. The real-time operation log provides an explicit history of every action performed, while the immediate display of analysis results (see Figure. 4) provides clear feedback on the consequences of the user's model. This adherence to system visibility ensures that users are never left guessing about the current state of the system or the outcome of their actions [24].

2.User Control and Freedom

Acknowledging that users will make mistakes, the system provides a clear "Undo" function. This feature, which leverages the operation log, acts as an "emergency exit," allowing users to reverse unintended actions without penalty. This fosters a sense of freedom and encourages exploration and experimentation, which is crucial for learning and complex problem-solving.

3.Consistency and Standards

The interface leverages conventional visual language to reduce cognitive load. By using distinct, recognizable icons and color-coding for different node types such as attackers, hosts, and vulnerabilities (Figure. 3) the system follows platform conventions. This allows users to apply their prior knowledge to quickly understand the visual information presented, rather than having to learn a new and arbitrary visual syntax.

Aesthetic and Minimalist Design: By separating the high-level Attack Graph from the low-level Attack Trees into a dual-canvas system, the interface avoids presenting the user with excessive information at once. This "chunking" of information helps to manage complexity, ensuring that each view contains only the information relevant to the task at hand. This minimalist approach prevents cognitive overload and allows the user to focus more effectively [25].

By embedding these HCI principles into the design, the tool moves beyond being a simple data-entry front-end to become a user-centric analytical environment.

4.2. Potential as an Educational Tool

The abstract nature of network attack paths and vulnerability chaining can be a significant hurdle for those new to cybersecurity. The interactive and visual nature of our tool positions it as a valuable asset for educational and training purposes [26].

In an academic setting, the GUI can serve as an interactive "sandbox" or virtual laboratory. Instead of just reading about attack graphs and trees, students can actively construct them. They can build hypothetical network scenarios, place vulnerabilities, and run the analysis to see firsthand how an attack path is formed and how risk metrics are calculated. This hands-on, constructivist approach can dramatically deepen their understanding of theoretical concepts and bridge the gap between abstract knowledge and practical application [27].

In a professional training context, the tool can be used to simulate real-world scenarios for new security analysts. Senior staff can create templates based on their organization's network architecture, tasking trainees with modeling specific threats such as a phishing attack leading to lateral movement. This provides a safe and controlled environment for junior analysts to develop their analytical skills, learn to identify critical assets and high-risk paths, and become familiar with their organization's specific security posture [28].

4.3. Facilitating Communication in Organizational Contexts

One of the most persistent challenges in cybersecurity is the communication gap between technical security teams and non-technical business leaders. Our tool can serve as a powerful instrument to bridge this divide and support data-driven decision-making.

A visual representation of an attack path is a potent communication artifact. The clear, graphical output of the tool can function as a "boundary object", a shared representation that both technical and managerial staff can understand, albeit from their own perspectives. For a CISO presenting to a board of directors, a single, compelling visualization showing a direct attack path to a "crown jewel" asset is far more impactful than pages of technical jargon and vulnerability scores. It translates abstract risk into a concrete, understandable narrative.

This enhanced communication directly enables more effective decision-making. When management can clearly see which vulnerabilities create the most dangerous attack paths, they are better equipped to make informed decisions about resource allocation. The visual analysis can help answer critical business questions: "Where should we invest our limited security budget to achieve the greatest risk reduction?" or "What is the security impact of delaying this patch?" By making the risk landscape transparent, the tool helps shift security investment from a reactive, fear-based model to a proactive, strategic, and justifiable one.

4.4. Limitations and Future Work

While this work provides a solid foundation, we recognize several limitations and opportunities for future enhancement, consistent with the findings in the original thesis.

The current tool is primarily designed for analyzing static network snapshots. A major avenue for future work is to extend its capabilities to support dynamic environments like cloud infrastructures and the Internet of Things, where network topologies and device states change frequently. This would involve developing mechanisms for real-time updates to the attack graph.

Although based on sound principles, the user interface could be further improved. Future versions could incorporate automatic graph layout algorithms to handle large-scale networks more gracefully, preventing visual clutter. Moreover, conducting formal usability studies with target users would provide valuable quantitative and qualitative data to guide further refinement of the user experience.

To augment the analyst's capabilities, machine learning models could be integrated into the tool. Such models could be trained on historical attack data to predict the most likely attack paths, automatically identify high-risk nodes that an analyst might overlook, and provide proactive security recommendations [22].

The current use of Python's pickle module for data persistence is functional but has limitations in cross-platform sharing and interoperability. Future work should include support for standardized data formats such as JSON or XML. This would not only improve data security and compatibility but also allow the tool to integrate more seamlessly with other security systems, such as Security Information and Event Management platforms [23].

5. Conclusion

This study has presented the design, implementation, and broader implications of a GUI for the HARM. We have demonstrated how a user-centric design approach can transform a powerful but abstract security model into a practical, intuitive, and accessible tool for network security analysis. By integrating a dual-canvas interface that mirrors HARM's two-layer structure with the Harmat analysis engine, the developed tool allows users to visually construct, analyze, and manage complex attack scenarios, effectively bridging the gap between theoretical modeling and real-world application. Key features such as interactive model building, integrated risk analysis, and data persistence provide a robust environment for both novice and expert users.

The core contribution of this work, however, extends beyond the mere implementation of a tool. We argue that in the increasingly complex domain of cybersecurity, a focus on human-centric design is not a luxury but a necessity. As discussed, the principles of effective HCI are critical for creating tools that reduce cognitive load and enhance analytical efficiency. Moreover, we have explored the significant potential of this visual tool to serve as an educational platform for training future cybersecurity professionals and as a vital communication bridge between technical experts and non-technical decision-makers. By translating abstract risks into clear, compelling visual narratives, such tools can facilitate more informed, data-driven security investments within organizations.

To further build upon this foundation, future work should focus on several promising directions. Extending the framework to support dynamic attack graph construction is essential for addressing the challenges of modern, fluid network environments like cloud and IoT systems. Further enhancement of the user interface through automated layout algorithms and formal usability testing would continue to improve the interactive experience. Finally, the integration of machine learning could introduce predictive capabilities, further augmenting the analyst's ability to identify and mitigate threats preemptively.

In conclusion, by placing the user at the center of the design process, we can forge a new generation of cybersecurity tools that are not only more powerful in their analytical capabilities but are also more intuitive, collaborative, and ultimately more impactful in the mission to protect our critical digital infrastructures.

References

1. W. S. Admass, Y. Y. Munaye, and A. A. Diro, "Cyber security: State of the art, challenges and future directions," *Cyber Security and Applications*, vol. 2, p. 100031, 2024. doi: 10.1016/j.csa.2023.100031
2. J. Lewis, "Economic impact of cybercrime, no slowing down," *McAfee, Center for Strategic and International Studies (CSIS)*, 2018.
3. A. Kuzior, "Cybersecurity and cybercrime: Current trends and threats," *Journal of International Studies*, vol. 17, no. 2, pp. 220-239, 2024. doi: 10.14254/2071-8330.2024/17-2/12
4. J. B. Hong, D. S. Kim, C. J. Chung, and D. Huang, "A survey on the usability and practical applications of graphical security models," *Computer Science Review*, vol. 26, pp. 1-16, 2017.
5. V. Shandilya, "Use of attack graphs in security systems," *Journal of Computer Networks and Communications*, vol. 2014, pp. 1-13, 2014. doi: 10.1155/2014/818957

6. F. Jia, J. B. Hong, and D. S. Kim, "Towards automated generation and visualization of hierarchical attack representation models," In *Proceedings of the 2015 IEEE International Conference on Computing and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 1689-1696. doi: 10.1109/cit/iucc/dasc/picom.2015.255
7. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002, pp. 273-284.
8. J. B. Hong, and D. S. Kim, "HARMs: Hierarchical attack representation models for network security analysis," In Proceedings of the 10th Australian Information Security Management Conference (Perth, WA, Australia). SRI Security Research Institute, Edith Cowan University., 2012.
9. J. B. Hong, and D. S. Kim, "Towards scalable security analysis using multi-layered security models," *Journal of Network and Computer Applications*, vol. 75, pp. 156-168, 2016. doi: 10.1016/j.jnca.2016.08.024
10. B. Schneier, "Attack trees," *Dr. Dobb's Journal of Software Tools*, 1999.
11. S. Y. Enoch, "Model-based cybersecurity analysis: Past work and future directions," *arXiv*, vol. 2, 2021. doi: 10.1109/rams48097.2021.9605784
12. H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Computer Science Review*, vol. 35, p. 100219, 2020. doi: 10.1016/j.cosrev.2019.100219
13. K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," In *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC)*, 2006, pp. 121-130. doi: 10.1109/acsac.2006.39
14. M. Mohammadzad, "MAGD: Minimal attack graph generation dynamically in cyber security," *Computer Networks*, vol. 236, p. 110004, 2023. doi: 10.1016/j.comnet.2023.110004
15. A. Palma, and S. Bonomi, "Behind the scenes of attack graphs: Vulnerable network generator for in-depth experimental evaluation of attack graph scalability," *Computers & Security*, vol. 157, p. 104576, 2025. doi: 10.1016/j.cose.2025.104576
16. J. B. Hong, and D. S. Kim, "Performance analysis of scalable attack representation models," In *Security, Privacy, and Information Processing Systems*, 2013, pp. 330-343. doi: 10.1007/978-3-642-39218-4_25
17. S. Y. Enoch, "HARMer: Cyber-attacks automation and evaluation," *IEEE Access*, vol. 8, pp. 129397-129414, 2020. doi: 10.1109/access.2020.3009748
18. G. A. Fink, C. L. North, A. Endert, and S. Rose, "Visualizing cyber security: Usable workspaces," In *Proceedings of the 6th International Workshop on Visualizing Cyber Security*, 2009, pp. 1-8.
19. M. Zipperle, "PARGMF: A provenance-enabled automated rule generation and matching framework with multi-level attack description model," *Journal of Information Security and Applications*, vol. 81, p. 103682, 2024. doi: 10.1016/j.jisa.2023.103682
20. S. Y. Enoch, Z. Huang, C. Y. Moon, D. Lee, M. K. Ahn, and D. S. Kim, "HARMer: Cyber-attacks automation and evaluation," *IEEE Access*, vol. 8, pp. 129397-129414, 2020. doi: 10.1109/access.2020.3009748
21. S. Y. Enoch, J. B. Hong, M. Ge, H. Alzaid, and D. S. Kim, "Automated security investment analysis of dynamic networks," In *Proceedings of Australasian Computer Science Week Multi-conference*, 2018, pp. 1-10. doi: 10.1145/3167918.3167964
22. J. A. Iman, "Refining UI/UX with minimalist design and AI: Towards sustainable and efficient digital experiences," *Procedia Computer Science*, vol. 269, pp. 669-680, 2025. doi: 10.1016/j.procs.2025.09.010
23. T. V. Sumithra, "Evolving usability heuristics for visualising augmented reality/mixed reality applications using cognitive model of information processing and fuzzy analytical hierarchy process," *Cognitive Computation and Systems*, vol. 6, no. 1-3, pp. 26-35, 2024. doi: 10.1049/ccs2.12109
24. N. Loftus, and H. S. Narman, "Use of machine learning in interactive cybersecurity and network education," *Sensors*, vol. 23, no. 6, p. 2977, 2023. doi: 10.3390/s23062977
25. A. Salman, "Integrating artificial intelligence in cybersecurity education: A pedagogical framework and case studies," In *2024 International Conference on Computer and Applications (ICCA)*, 2024, pp. 1-5. doi: 10.1109/icca62237.2024.10927933
26. W. Lazarov, "Lessons learned from using cyber range to teach cybersecurity at different levels of education," *Technology, Knowledge and Learning*, 2025. doi: 10.1007/s10758-025-09840-y
27. P. Sarlin, "Macroprudential oversight, risk communication and visualization," *Journal of Financial Stability*, vol. 27, pp. 160-179, 2016. doi: 10.2139/ssrn.2583762
28. M. Ge, J. B. Hong, W. Guttmann, and D. S. Kim, "A framework for automating security analysis of the Internet of Things," *Journal of Network and Computer Applications*, vol. 83, pp. 12-27, 2017. doi: 10.1016/j.jnca.2017.01.033