frontiers | Frontiers in Computer Science

TYPE Technology and Code
PUBLISHED 29 February 2024
DOI 10.3389/fcomp.2024.1304288

Check for updates

OPEN ACCESS

EDITED BY
Dileep Kumar Yadav,
Bennett University, India

REVIEWED BY
Prerna Agarwal,
Bennett University, India
Naween Kumar,
Bennett University, India
Vinay Saini,
Maharaja Agrasen Institute of Technology, India

*CORRESPONDENCE
Atul Rana
✉ 20cs9002@mvn.edu.in

RECEIVED 29 September 2023
ACCEPTED 06 February 2024
PUBLISHED 29 February 2024

CITATION
Rana A, Gupta S and Gupta B (2024) A comprehensive framework for quantitative risk assessment of organizational networks using FAIR-modified attack trees. *Front. Comput. Sci.* 6:1304288. doi: 10.3389/fcomp.2024.1304288

COPYRIGHT
© 2024 Rana, Gupta and Gupta. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

A comprehensive framework for quantitative risk assessment of organizational networks using FAIR-modified attack trees

Atul Rana[1]*, Sachin Gupta[2] and Bhoomi Gupta[2]

[1]MVN University, Faridabad, India, [2]Maharaja Agrasen Institute of Technology, Delhi, India

Attack trees are a widely used method for threat modeling and analyzing cyber-attacks in organizational networks. Assessing the risk associated with each individual node of an attack tree is crucial for understanding the overall risk of the attack. This article presents a comparative study of different threat modeling methods and risk assessment approaches in organizational networks. The article also presents a novel comprehensive approach for quantifying risk assessment of organizational networks based on attack trees modified according to the factor analysis of information risk (FAIR) approach. Our results demonstrate the effectiveness of the novel approach in capturing the unique characteristics of different assets and their dependencies in an attack tree, leading to quantitative risk assessment.

KEYWORDS

attack trees, threat modeling, risk assessment, organizational networks, FAIR approach, quantitative assessment, resource optimization

1 Introduction

The increasing reliance on information technology in our daily lives has led to a corresponding rise in the number and severity of cyber-attacks. There are a variety of threat modeling tools available for the analysis and management of organizational asset security, where each method has its own strengths and limitations. Attack trees are a commonly used graphical technique for modeling and analyzing cyber-attacks, allowing analysts to understand the different stages of an attack and identify vulnerabilities that can be exploited. Attack trees are primarily scenario-based and portray the potential steps an attacker could take to compromise a system or network. An attack tree is essentially a tree structure where each node represents a possible attack method or subgoal, and the edges represent the relationship between them. The paths in trees are used to represent the mindset of an attacker, to penetrate the network to gain access to sensitive information or disrupt services. Attack trees also provide a useful communication tool for security professionals to discuss and convey the risks associated with a particular system or network. By presenting the attack tree in a visual format, it is easier for stakeholders to understand the potential threats and make informed decisions about risk management strategies.

Assessing the risk associated with each node of an attack tree is a critical step in understanding the overall risk of the attack. This allows the organization to prioritize its security efforts and focus on the most critical areas of the system. Over the years, attack trees have been used by both blue teams and red teams for threat assessment and management.

This study presents a comprehensive review of various threat modeling approaches, including attack trees, along with the evolution of attack trees through the past few decades to understand the gaps in a scenario-based risk assessment presented by attack trees. Based on the gaps identified, the study presents a comprehensive approach for quantifying the risk assessment of organizational networks based on asset properties, including impact, the likelihood of an attack, and the depth of resources. This is a significant improvement over the non-quantified scenario-based approach of the traditional attack trees.

Red teams can leverage attack trees to simulate potential attack paths and identify vulnerabilities in an organization, which can then be exploited to test and improve the organization's defenses.

1. Map potential attack path to simulate: the goal of any red team is to simulate the attacks to exploit vulnerabilities in the target system. Working with attack trees helps with visualizing the various paths. This also helps the stakeholders to understand the risks and potential threats.

2. Identifying weaknesses and attack surface: one of the big advantages of using the red team lies in mapping out potential entry points to exploit and then using them for privilege escalation and lateral movement.

3. Simulate exploit: using the most vulnerable path identified, it can be used to simulate exploits and attacks on the target systems. It can be used by external attack surface management (EASM) to create related IPs and URLs.

4. Improve and retest: the attack trees can be reused during the periodic red teaming exercise to show the hardening and improving the defenses of the organization.

Blue teams, also known as defenders, use the attack trees to improve the security of their organization by identifying **potential attack scenarios** and developing proactive measures to prevent or mitigate them. Typically, blue teams can use attack trees for the following purposes:

1. Identify attack paths: by using attack trees, blue teams can visualize the potential attack paths that an attacker might take to reach their ultimate goal. This can help the blue team focus their efforts on hardening those points of entry and minimizing the risk of a successful attack.

2. Conduct risk assessments: attack trees can be used to identify and assess the risks associated with specific attack scenarios. Blue teams can use this information to prioritize their security efforts and allocate resources accordingly.

3. Develop defense strategies: attack trees can help blue teams develop proactive defense strategies by identifying potential vulnerabilities, attack vectors, and threat actors. This can include implementing security controls, such as access controls, intrusion detection systems, or network segmentation.

4. Plan incident response: in the event of a security breach, blue teams can use attack trees to quickly identify the attack path and develop an incident response plan. This can help to minimize the impact of the breach and prevent further damage to the organization.

The remainder of the article is organized as follows: Section 2 covers a comprehensive review of the popular threat modeling methods presented with their strengths and limitations. It also covers a comprehensive summary of the threat modeling research focused on attack tree-based methods to explore gaps in the studies. Section 3 includes a detailed research methodology for quantifying risks in the present research, followed by a discussion of the results in Section 4. The article is concluded in Section 5 with some brief directions for future research work in this domain.

2 Literature review

Threat modeling is a systematic approach for identifying and assessing potential security threats and vulnerabilities that could affect an organization or system. It involves analyzing the system's design, identifying potential security weaknesses, and evaluating the likelihood and impact of each threat.

The process of threat modeling has been gaining importance since the 1990s, as the increasing use of computers and the internet led to a rise in cyber-attacks and other security breaches. Threat modeling provides a way to proactively identify and mitigate potential security risks before they can be exploited by attackers. By taking a structured and methodical approach to security, organizations can reduce the likelihood of security breaches and minimize the impact of any that do occur.

Threat modeling has also become more important in recent years as organizations have become increasingly dependent on technology and data to operate their businesses. With the rise of cloud computing, the Internet of Things (IoT), and other new technologies, the attack surface for potential threats has increased, making it more important than ever for organizations to understand and manage their security risks. Threat modeling is a key tool for achieving this goal. Table 1 summarizes the most popular threat modeling tools that have remained in use over the last two decades.

The survey shows that there is a serious lack of automated tools except for IriusRisk, Microsoft Threat Modeling Tool, and VAST+. The majority of the available threat modeling tools are open source, with their roots tracing to attack trees. The present research is focused on attack trees in particular, as a majority of the other threat modeling techniques use some form of patterns originating from the attack tree methodology. There has been a lot of research specific to attack trees, and a summary of the comprehensive literature to review the security research focused on attack trees is presented in Table 2.

The history of the attack tree started with two important articles in which it was formalized by Weiss and Schneier. These research articles ensured that the concept of the attack tree required formalization. This was the first introduction of the AND/OR node, which was later converted to the root node and leaf node. As mentioned in the table above, there are different definitions and explanations of attack trees as graphical tools for modeling and analyzing threats and their importance in threat modeling for an organization. The article on attack trees also introduced the idea that security engineering processes need additional resources allocated to high-risk vulnerabilities to keep the organization secure. There have been multiple explanations regarding the methodologies used in the creation of attack trees, including formalisms, notations, and modeling techniques, i.e., AND/OR trees, STPA, AFT, and ADT. The attack tree

TABLE 1 Feature summary and limitations of different threat modeling tools.

Method	Working summary	Strengths	Limitations
Attack Trees (Schneier, 2015) 2015	Uses a tree-based structure to represent different ways in which an attacker could penetrate a system and the corresponding countermeasures that can be taken to prevent or mitigate those attacks	Can identify potential vulnerabilities and weaknesses in a system	Not suitable for complex systems
CAPEC (MITRE, 2023) 2023	Common attack pattern enumeration and classification, which consists of attack patterns and methods	Useful for identifying potential threats based on known attack patterns	Fails to capture unknown or novel attacks
DREAD (Microsoft Press, 2003) 2003	The risk assessment method is based on five factors: damage potential, reproducibility, exploitability, affected users, and discoverability, which quantifies risks and prioritizes them based on the impact	Simple and easy to use	Fails to consider environmental factors that may affect the likelihood of an attack
IriusRisk 2023[a]	Integrated platform for threat modeling and risk analysis, based on the PASTA methodology and includes integration with development tools	Provides a comprehensive and centralized approach to threat modeling, allows for integration with development tools	May require significant resources and time to fully implement
ISTR	Intel security threat report, which provides insights into current and emerging cyber threats and trends	Useful for identifying potential threats and vulnerabilities	Limited to current known threats
LINDDUN 2023[b]	Linking security requirements and threats with risk mitigation, which provides a structured approach to threat modeling	Provides guidance on risk management and compliance	Can be time-consuming and resource-intensive
Microsoft STRIDE (Archived docs, 2009) 2023	Microsoft's proprietary threat modeling approach, based on STRIDE threat classification	Widely used and supported within Microsoft products, includes detailed guidance and documentation	Limited to STRIDE threat classification, may not be suitable for non-Microsoft environments
Microsoft Threat Modeling Tool 2023[c]	An automated tool that implements the STRIDE threat classification approach includes automated report generation	Provides an efficient and standardized approach to threat modeling, allows for collaboration and team-based threat modeling	Limited to STRIDE threat classification, may not be suitable for non-Microsoft environments
PASTA (Morana and Uceda Vélez, 2015) 2015	An iterative and incremental process of threat modeling, based on seven stages: planning, scoping, data flow diagramming, threat analysis, risk ranking, countermeasures, and report, which are comprehensive, adaptable, and scalable	Considers both technical and non-technical aspects of security	May require significant time and resources
PTA 2023[d]	Penetration testing and assessment: it identifies attacks from a potential attacker's perspective	Can identify potential vulnerabilities and weaknesses in a system	May not be suitable for all environments
Security Cards 2023[c]	Method that uses a deck of cards to prompt brainstorming and discussion about potential security threats	Simple and easy to use, promotes collaboration and engagement among team members	May not provide a comprehensive approach to threat modeling and may require additional follow-up and analysis
Triangulation	It is a method that uses multiple perspectives to identify and assess security threats and includes a variety of methodologies and techniques	Provides a comprehensive approach to threat modeling, incorporates multiple viewpoints and expert opinions	May require significant resources and time to fully implement and may be more complex than other methods
Trike 2023[e]	A collaborative process of identifying threats and vulnerabilities using various flow diagrams and models	Considers both technical and non-technical aspects of security	May require significant time and resources

(Continued)

10.3389/fcomp.2024.1304288

TABLE 1 (Continued)

Method	Working summary	Strengths	Limitations
Trike 1.0 2023 [e]	Extension of the original Trike method includes additional threat modeling techniques such as attack trees and PASTA	Provides comprehensive threat modeling approach, customizable security controls list	Limited to STRIDE and DREAD threat classifications, requires expertise in PASTA
Trike+ 2023 [e]	Extension of the Trike method, including the use of additional diagrams and models and emphasizing business impact using context diagrams, misuse cases, and threat matrices	Considers both technical and non-technical aspects of security	May require significant time and resources
VAST (ThreatModeler, 2023)	Visual, Agile, and Simple Threat modeling methodology utilizes data flow diagrams and threat modeling templates	Provides a streamlined and easy-to-follow approach to threat modeling, allows for customization based on specific needs	May not provide as comprehensive approach as other methods, limited by the use of templates
VAST+ (ThreatModeler, 2023)	Extension of the VAST method, including testing for multiple scenarios and use cases to identify possible threats and generate possible attack path	Comprehensive and adaptable to different environments	May require significant time and resources

[a] The Automated Threat Modeling Platform. Available online at: https://www.iriusrisk.com (accessed April 03, 2023).
[b] Symantec Security Center. Available online at: https://www.broadcom.com/support/security-center? (accessed April 03, 2023).
[c] Microsoft Download Center. *Microsoft Threat Modeling Tool 2016*. Available online at: https://www.microsoft.com/en-us/download/details.aspx?id=49168 (accessed April 03, 2023).
[d] The Penetration Testing Execution Standard. Available online at: http://www.pentest-standard.org/index.php/Main_Page (accessed April 03, 2023).
[e] Trike. octotrike.org. Available online at: http://www.octotrike.org/ (accessed April 03, 2023).

has undergone significant transformation in its methodologies and construction and maintenance techniques depending on the complexity of the environment. Various techniques have emerged as fundamental frameworks for constructing attack trees, each offering unique advantages and challenges. Several approaches describe how to design an attack tree. A group of experts proposed designing a tree manually, analytically, and iteratively, considering all possible attack scenarios. There has been an evolution of automated and assisted attack tree generation techniques, which have emerged recently and help to create static scenarios. Despite these evolutions, challenges still exist, which require efforts to address scalability issues, dynamic threats, and integrating attack trees with other modeling methodologies, which is still lacking in the literature review. This has also been addressed below.

2.1 Gaps in the previous research

The most obvious patterns that are implied based on the various research and methods mentioned in Table 2 are as follows:

- Lack of practical translation to real-world scenarios till now; in this study, we are attempting to align attack trees with existing organization networks, which can enhance their practicality by connecting them to established network design and create better visibility within the environment.
- Priority classifications of assets based on priority using attack trees help assess the criticality and importance of various assets within an organization's infrastructure within the organization. Attack trees mentioned in this study describe that help by prioritizing assets, identifying potential threats, and their impacts on those assets.
- Scalability is an issue that has been mentioned in previous research articles and methods when dealing with attack trees, particularly in larger and more complex systems; in this study, we attempted to solve these methods by prioritizing the assets based on their values in the organization, using the modular approach by breaking down the complex attack tree in smaller and more manageable modules.
- The absence of a quantifiable framework for attack tree-based risk assessment using relevant metrics presents a challenge in cybersecurity within the current organization setup. Organizations lack quantitative risk assessment. In this study, we are introducing the factor analysis of information risk (FAIR) approach that balances between complexity and practicality and aims to provide actionable insights to support decision-making processes based on the crown jewels within the organization and their risk related to cybersecurity risk mitigation.
- Visualization techniques, such as color coding or network diagrams, can significantly aid in conveying complex information present in attack trees, and we have used these methods of visualization here in this attack tree.
- The methods focus on either attack defense trees (ADT) or attack fault trees (AFT); no method has been mentioned, and research does a combination where we can help the organization improve its defense using the method.

TABLE 2 Threat modeling research based on attack trees.

References	Main contributions	Limitations and gaps
Weiss (1991)	Introduced threat logic trees based on fault trees. Used to identify and assess potential security threats and vulnerabilities. Assigned negative impact and required effort levels to each leaf. Calculated risk for all nodes in the tree bottom-up.	Published over 30 years ago, focuses primarily on technical aspects of security engineering and not on creating and maintaining the tree.
Schneier (2015)	Introduced attack trees to represent the different ways in which an attacker could penetrate a system and corresponding countermeasures. Used to identify potential vulnerabilities and weaknesses in a system and evaluate the effectiveness of different security measures. Applied to real-world security scenarios.	Lack of real-world examples and practical implementation guidance. No comparison with other methods of security risk analysis. No standardization on creating and analyzing the attack tree.
Sheyner et al. (2002)	Introduced an automated approach for generating and analyzing attack graphs. Represented the various ways an attacker can gain access to a computer system. Proposed an attack graph generation algorithm.	No documentation on the maintenance of the attack tree in the future. The paper was designed on a very simple playground network which is far different from real-life scenarios that consist of complex environments and situations.
Kaiser et al. (2004)	Proposed a new component concept for fault trees based on components used in attack trees. Components allow an attack tree to be split up into more manageable parts.	Only provides a basic understanding of fault trees and does not cover the creation of attack trees, which is missing in this paper.
Bistarelli et al. (2007)	Proposed a new approach for modeling and analyzing strategic interactions between attackers and defenders in the context of cybersecurity. Modeled defense trees as strategic games to understand the dynamics of cyber-attacks and the effectiveness of different defense strategies.	Does not take into account the uncertainty and incomplete information present in real-world cyber-attacks. Assumes perfect rationality and knowledge of the attacker and defender. Does not provide a comprehensive evaluation of the proposed model or a comparison with existing approaches.
Mauw and Oostdijk (2006)	Proposed a formal framework for modeling and analyzing attack trees. Provided a comprehensive review of existing work on attack trees and proposed formal semantics for attack trees. Introduced a formal definition of attack trees based on a set of operators that represent different types of attacks and their relationships.	The formal semantics of attack trees assume the perfect knowledge and rationality of the attacker. Does not provide a comprehensive evaluation of the proposed approach or a comparison with existing approaches. Does not address the challenges of modeling and analyzing large and complex attack trees.
Edge et al. (2006)	Proposed a methodology for modeling and analyzing the security of critical infrastructure using attack and protection trees. Modeled the system's assets, threats, and vulnerabilities and developed a set of attack and protection trees. Analyzed the trees to identify critical vulnerabilities and develop effective defense strategies.	Lack of empirical validation, limited consideration of human factors, and limited guidance on implementation. May not be scalable to very large or complex systems and does not fully consider cost and resource constraints.
Yager (2006)	Proposed a new approach to modeling security using OWA trees and attack trees. Used OWA trees to model the uncertainty and imprecision associated with different security parameters and attack trees to model potential attack paths.	May be complex and difficult to implement as it requires a strong background in mathematical modeling. May not fully address scalability to large or complex systems. May not fully consider the human factors involved in security.
Bistarelli et al. (2008)	Proposed an approach for analyzing security scenarios using defense trees and answer set programming. Modeled the security of a system using defense trees and represented the trees	The model does not take into account the uncertainty and incomplete information that is often present in real-world cyber-attacks.
Zonouz et al. (2009)	Developed RRE intrusion-detection system using game theory and optimization techniques to predict and respond to attacks.	Limited scalability and complexity of models used to capture attacker's behavior may impact applicability in real-world scenarios.
Jürgenson and Willemson (2010)	Proposed a new algorithmic model for efficiently computing attack trees called the Serial Model.	Assumes that all leaf nodes in the attack tree are atomic events, does not provide a comprehensive evaluation of the proposed model, does not address the issue of scalability for large and complex attack trees, and does not consider the dynamic nature of attack scenarios.
Abdulla et al. (2010)	Proposed attack jungles, an extension of attack trees, as a tool for analyzing the security of the GSM radio network.	The complexity of the attack jungles, limited scope, and lack of detailed explanation of the process for creating the attack jungles.
Piètre-Cambacédès and Bouissou (2010)	Proposed a new approach to dynamic security modeling called BDMP that extends the traditional attack tree method.	BDMP model may be more complex and difficult to construct than traditional attack trees and does not provide a detailed evaluation of the effectiveness or efficiency of the BDMP model compared to other methods for dynamic security modeling.
Whitley et al. (2011)	Proposed a methodology for attributing attack trees to identify the group or individual responsible for creating them.	Relies on the availability of contextual information, is reliant on the accuracy and completeness of the attack tree itself, and does not address how to incorporate uncertainties and dependencies between attack steps in the analysis.

(Continued)

10.3389/fcomp.2024.1304288

TABLE 2 (Continued)

References	Main contributions	Limitations and gaps
Roy et al. (2012)	Proposed a new model called Attack Countermeasure Trees (ACT) that unifies the constructs of Attack Trees and Defense Trees.	Does not provide a comprehensive evaluation of the proposed model and does not provide a comparison of ACT with other existing models.
Poolsappasit et al. (2012)	Proposed a new approach to dynamic security risk management that uses Bayesian attack graphs.	The effectiveness of the approach is highly dependent on the accuracy of the data used to build the Bayesian networks, and the approach does not take into account the human factor, such as insider threats or social engineering attacks.
Poolsappasit et al. (2012)	Proposed a new approach to dynamic security risk management that uses Bayesian attack graphs.	The effectiveness of the approach is highly dependent on the accuracy of the data used to build the Bayesian networks, and the approach does not take into account the human factor, such as insider threats or social engineering attacks.
Ingoldsby (2013)	Presented a methodology for performing threat risk analysis using attack trees.	Does not provide a detailed discussion of the limitations and assumptions of the attack tree methodology, does not address how to incorporate uncertainties and dependencies between attack steps in the analysis, and does not provide guidance on how to assess the effectiveness of mitigation strategies or how to update the attack tree as new information becomes available.
Kordy et al. (2013a)	Presented ADTool, an open-source tool that supports the modeling and analysis of attack-defense trees for security analysis.	Depending on the accuracy of the input information, it does not provide sufficient information on the performance and scalability of the tool when dealing with large and complex systems and assumes that the attacker has complete knowledge of the system.
Kordy et al. (2013b)	Presented a formalism called "quantitative attack-defense trees" (QADT), which enables security analysts to reason about the likelihood and impact of security attacks.	QADTs do not take into account the dynamic nature of the system and its dependencies, which can lead to incorrect risk assessments
Pieters et al. (2014)	Proposes a framework for security risk analysis using "attacker profiles" to model different attack scenarios. Introduces the attacker profile library, the attack navigator, and the attack simulator. Provides a detailed example of using the framework to analyze a smart grid system.	Requires significant effort to create and maintain the attacker profile library. Assumes that the attacker's behavior is known in advance, which may not always be the case. Relies on the accuracy of the system model and the attacker profiles, which may be difficult to achieve in complex systems.
Kordy et al. (2014)	Proposes an extension to attack trees called attack-defense trees (ADTrees) that include modeling defenses. Provides a structured approach to modeling the security of a system and prioritizing defense measures. Presents a case study demonstrating the effectiveness of ADTrees in modeling and analyzing attacks and defenses.	Not easily scalable to large and complex systems. No formal method for incorporating uncertainties and dependencies in the analysis. Evaluation is only performed on a single case study, requiring further validation.
Kumar et al. (2015)	Proposes a quantitative approach to analyzing attack trees using priced timed automata (PTA). Presents an algorithm for translating attack trees into PTA and computing various quantitative measures, such as the probability of a successful attack. Demonstrates the effectiveness of the approach in several case studies.	Assumes a complete and accurate attack tree, which may not always be feasible. Does not account for the dynamic nature of attacks. The complexity of PTA may grow rapidly, making it challenging to compute desired measures.
Audinot and Pinchinat (2016)	Focuses on the formal verification of attack trees using transition systems and Petri nets. Provides a set of rules for constructing attack trees that satisfy certain properties. Presents a method for verifying the soundness of attack trees using model-checking	Only focuses on the soundness of attack trees and does not address other important aspects such as completeness, scalability, and efficiency. The method may not be practical for large-scale attack trees.
Fraile et al. (2016)	Demonstrates the use of ADTrees for threat analysis in an automated teller machine (ATM) system. Provides a step-by-step approach for constructing the ADTree and evaluating the effectiveness of countermeasures. Discusses the limitations of ADTrees in threat analysis.	ADTrees may not be suitable for large-scale systems. More research is needed to address this issue.
Kumar and Stoelinga (2017)	Proposes an extension to attack trees called "attack fault trees" (AFTs) to model security and safety threats in a unified manner. Shows how AFTs can be translated into Markov chains for quantitative analysis. Demonstrates the applicability of the AFT framework through several case studies.	Unable to capture all possible attack scenarios due to the difficulty of modeling attacker behavior accurately. More research is needed to improve the scalability of the analysis and integrate AFTs with other modeling frameworks.

10.3389/fcomp.2024.1304288

TABLE 2 (Continued)

References	Main contributions	Limitations and gaps
Audinot et al. (2017)	Proposes a formal verification approach to ensure the correctness of attack trees using logic and formal methods. Presents a tool, named ATREES, for checking the correctness of attack trees. Provides visual feedback to help users understand errors and how to fix them.	May not scale well to large and complex attack trees. Does not provide guidance on how to construct attack trees.
Lallie et al. (2002)	Provides a comprehensive review of visual syntax used in attack graphs and trees for threat analysis	Lacks discussion on the practical implications of these visual representations in real-world scenarios. Does not cover other visualization techniques, such as heat maps or network diagrams
Bolivar et al. (2019)	Proposes using attack trees for cloud security analysis and provides a structured approach to capture and analyze various attack scenarios	Does not provide a detailed evaluation of the effectiveness of the approach in real-world cloud environments, and scalability is a potential issue
Bryans et al. (2020)	Proposes a template-based approach for automatically generating attack trees that is faster and more efficient than manual methods	Relies on pre-existing knowledge of attack trees and does not account for the evolving nature of threats
Fila and Widel (2020)	Proposes an approach to finding an optimal set of countermeasures for a system using attack-defense trees and game theory	Assumes that the attack-defense tree is complete and accurate and does not address the issue of uncertainty in the system's state. Effectiveness and scalability in practice are unknown
Hyder and Govindarasu (2022)	Proposes a methodology for optimizing cybersecurity investment in smart grids using attack-defense trees and game theory	May not be applicable to other types of systems outside of smart grids and assumes that system components are independent and identically distributed. Does not address the issue of uncertainty in the system

In recent studies, a test lab setup for simulation and attack tree generation is demonstrated in Gupta et al. (2023), which can be used in conjunction with the MITRE ATT&CK framework with a provision to create and assess various attack scenarios while providing flexibility in subnet configuration and movement, addition or removal of networking devices. In Rana et al. (2023a), the authors have proposed that the attack trees risk assessment is no longer limited to blue team activity, and a dual attack tree-assisted command and control server activity has been proposed to ensure enhanced path coverage and test coverage by the red team during security validation and penetration testing.

3 Bridging the gap with FAIR-modified attack trees

We propose a novel approach for assessing risk quantitatively in modified attack trees based on the FAIR (Freund and Jones, 2015) approach in security. The FAIR approach is a risk management framework that is used to evaluate and analyze the value of data, the potential impact of a security incident, and the cost of implementing security measures.

The FAIR approach involves four steps as follows:

1. **Identify assets**: the first step is to identify the assets that need to be protected, including data, systems, and processes. This is a standard part of an attack tree assessment as well.

2. **Evaluate threats**: the second step is to evaluate the threats that could impact those assets. This includes assessing the likelihood and impact of different types of security incidents, such as data breaches, cyber-attacks, or natural disasters. We have modified the threat evaluation to suit the attack tree methodology.

3. **Assess vulnerabilities**: the third step is to assess the vulnerabilities of the assets and determine how easy they are to exploit. This includes evaluating technical vulnerabilities and human factors that could contribute to security incidents. This is again a standard part of the attack tree methodology, but we have introduced quantified values for vulnerabilities.

4. **Determine the risk**: the final step is to determine the level of risk associated with each asset. This involves calculating the likelihood and impact of security incidents and the cost of implementing security measures to mitigate those risks. We have added another step to calculate the overall risk score of the organization based on risks calculated for individual assets in the modified attack tree.

To apply the FAIR approach in the context of the attack trees, we have identified and included the following terminology from both domains.

3.1 Countermeasures

The availability of possible countermeasures (such as firewalls, intrusion detection systems, encryption, network segmentation, and access control) for safeguarding critical resources from potential attackers. We have used a range of 0 to 1 to demonstrate the relative values of countermeasure effectiveness (Figure 1).

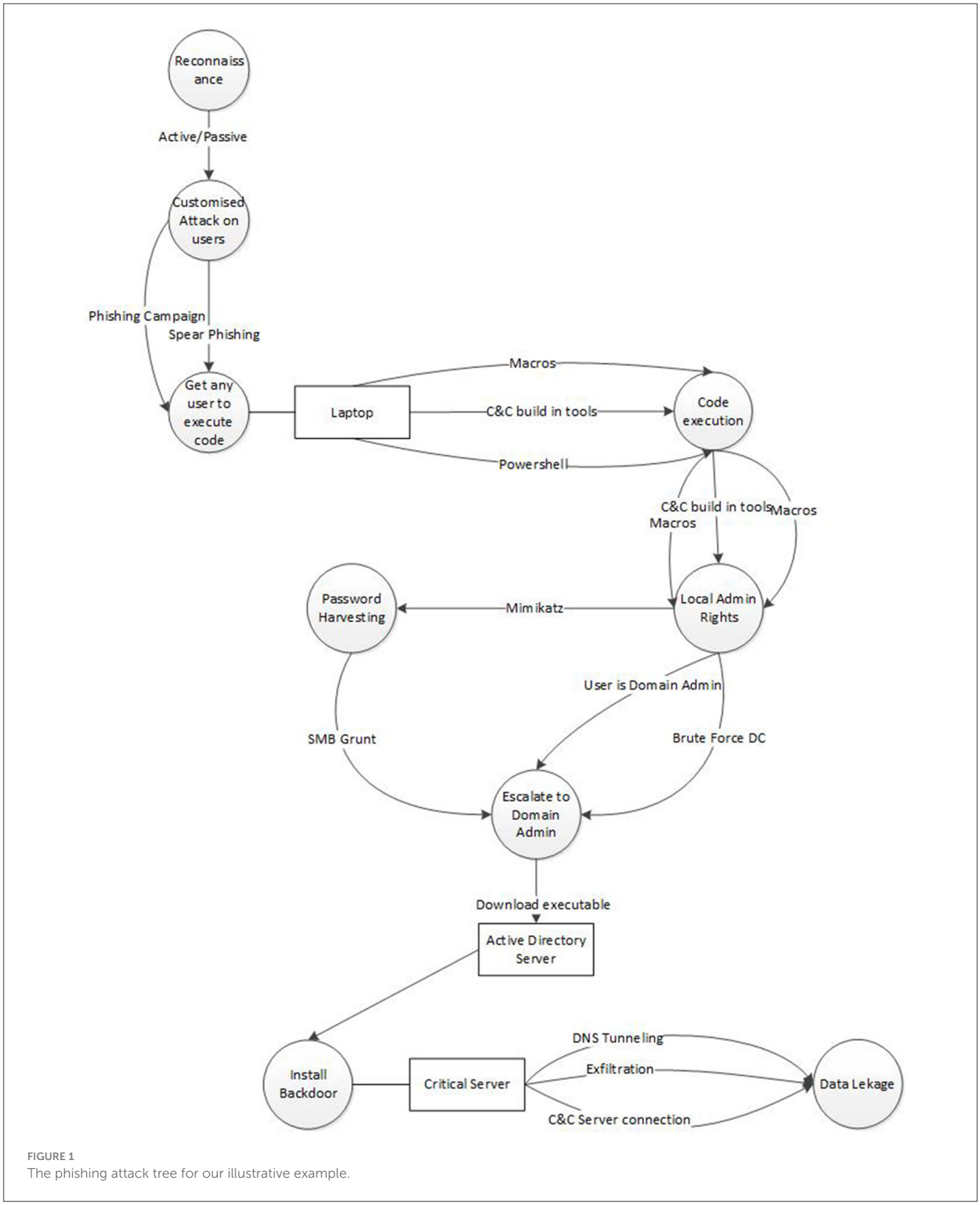

FIGURE 1
The phishing attack tree for our illustrative example.

3.2 Depth

This parameter indicates the number of countermeasures leading to a resource within the organizational network from the single point of access to the network. It may also indicate the distance in segments from the single point of access based on the implementation. The higher the value of depth, the more secure the resource may be considered.

3.3 Impact

An organization may associate some numerical value with the potential impact of an attack on its operations as per the relative importance of each resource. In our implementation, we have chosen impact to range from 0 to 1,000 in increments of 100. Crown Jewels within the organization will have a higher significance (Figures 2, 3).

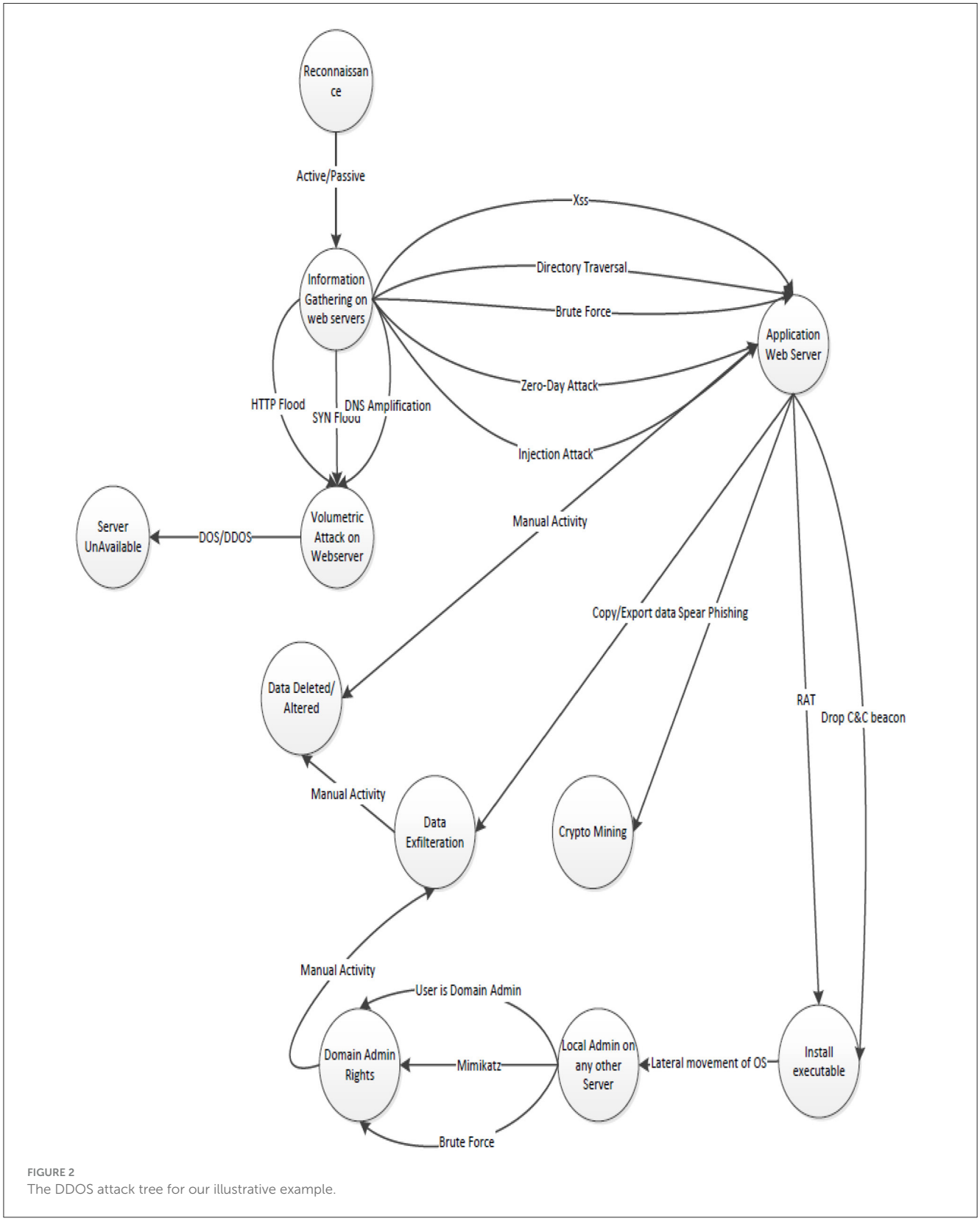

FIGURE 2
The DDOS attack tree for our illustrative example.

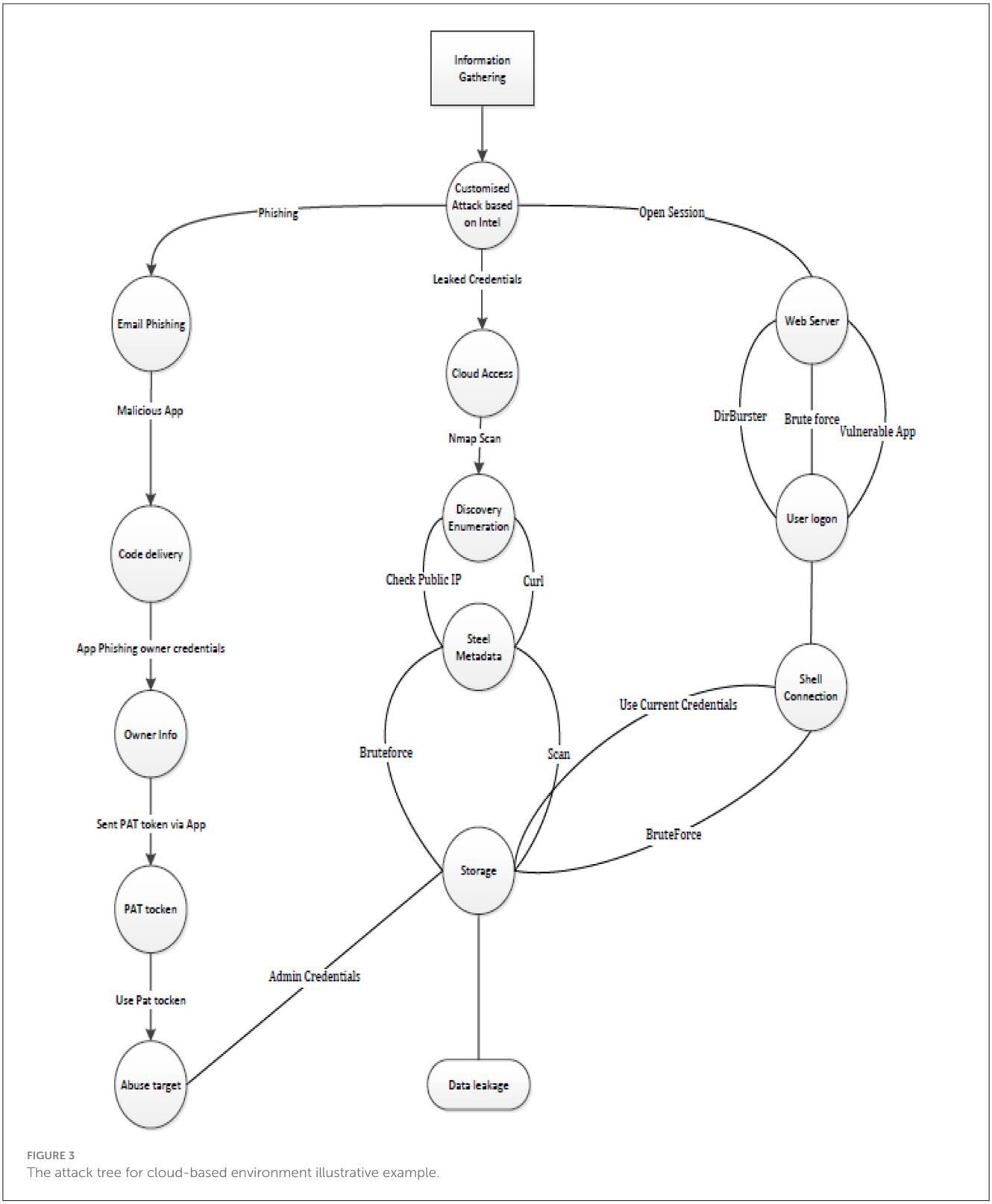

FIGURE 3
The attack tree for cloud-based environment illustrative example.

3.4 Likelihood

This parameter signifies how likely the attackers are to target a specific resource based on its criticality/usefulness. We have used a range of 0–1 for the likelihood of an attack for each resource, with a higher value indicating that a resource is more likely to be attacked (Figures 4, 5).

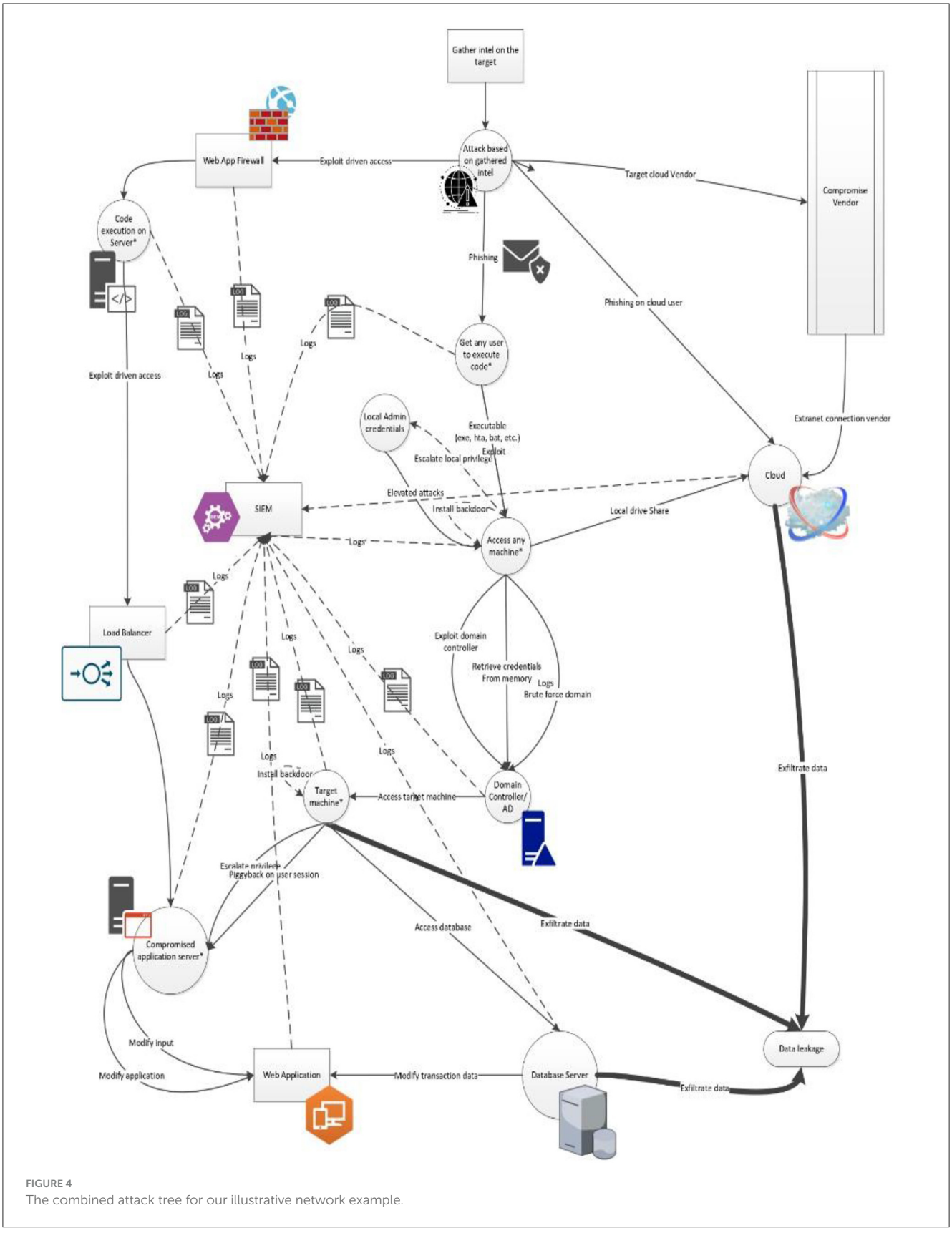

FIGURE 4
The combined attack tree for our illustrative network example.

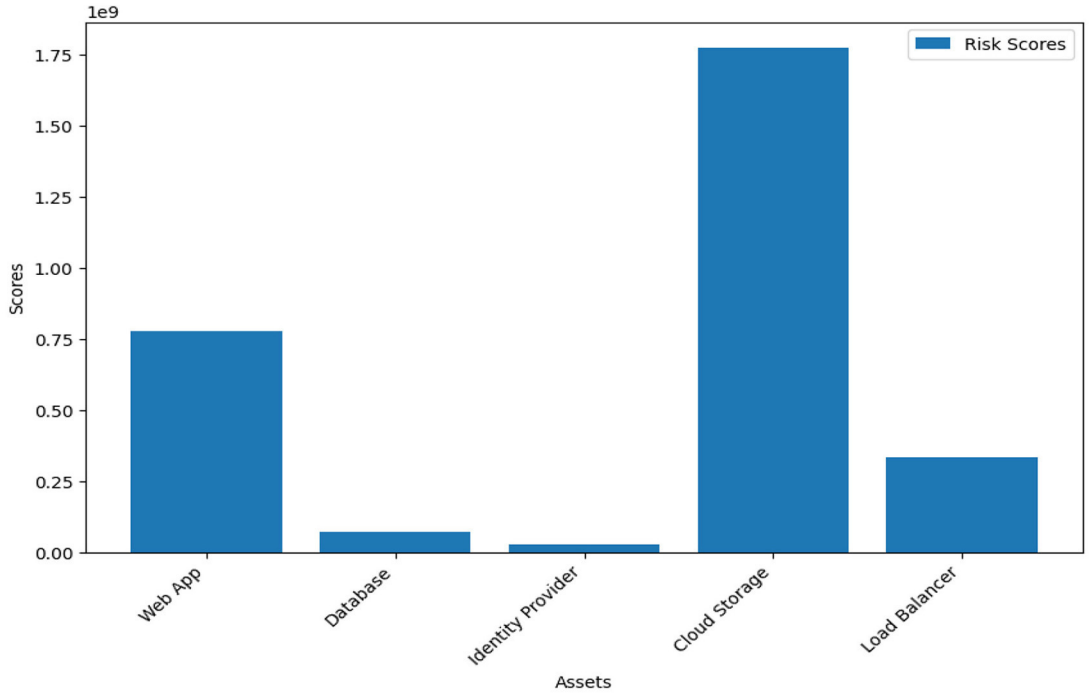

FIGURE 5
Individual asset risk scores based on impact, likelihood, and depth.

3.5 Asset value

The asset value has been used to indicate the tangible monetary value that an organization attributes to the worth of an asset. Please note that we have not used the "reputation" value for assessment, but it is a very crucial requirement, which we would like to address in our future work.

4 Methodology

Each resource on the modified attack tree shall be associated with five parameters as listed above and can be represented with a simple record, as shown with an illustrative example in Equation (1) below.

$$'WebApp':\{'depth':3,'impact':500,'likelihood':0.3,'asset\_value' \\ 500000,'countermeasure':1'\} \quad (1)$$

This approach takes into account the unique characteristics of different assets and their dependencies in an attack tree. The impact of an asset is measured by its potential damage or loss, while the likelihood of an attack is based on its probability of occurrence. The depth of an asset in the attack tree reflects its position in the attack chain and its dependence on other assets. We can then calculate the risk score of each asset by combining these factors as per Equation (2).

$$Calculate\_risk\_score = impact\_score * likelihood\_score * \\ depth\_score * cost\_score * countermeasures\_score * asset\_value \quad (2)$$

We have also estimated the total risk score of an attack tree with multiple assets based on the five parameters listed above. To achieve this, we aggregated the individual risk scores of each asset into a single overall risk score for the attack tree. The steps are listed below.

- Assign a potential impact score to each asset based on its value to the organization.
- Estimate the likelihood of detection for each asset based on its depth in the attack tree and other relevant factors.
- Use the potential impact score and likelihood of detection to calculate the risk score for each asset.
- Aggregate the individual risk scores for each asset to obtain an overall risk score for the attack tree. Presently, we have used a weighted average of the individual risk scores, where the weights are based on the relative importance of each asset to the organization.

We calculate the risk score for each asset using a loop over the assets dictionary and store the result in the asset_risk_scores dictionary. Then, we calculate the overall risk score for each attack step by looping over the attack_steps. For a differential risk assessment of on-premise and cloud-based assets, each asset has a "type" property indicating whether it is an on-premise or cloud-based asset. For cloud-based assets, the risk score is multiplied by 0.5, as shown in Equation (3), to account for the reduced control that organizations typically have over cloud-based assets. This value may also be assigned by the analyst (Figure 6).

$$Asset\_risk\_scores = calculate\_risk\_score * 0.5 \quad (3)$$

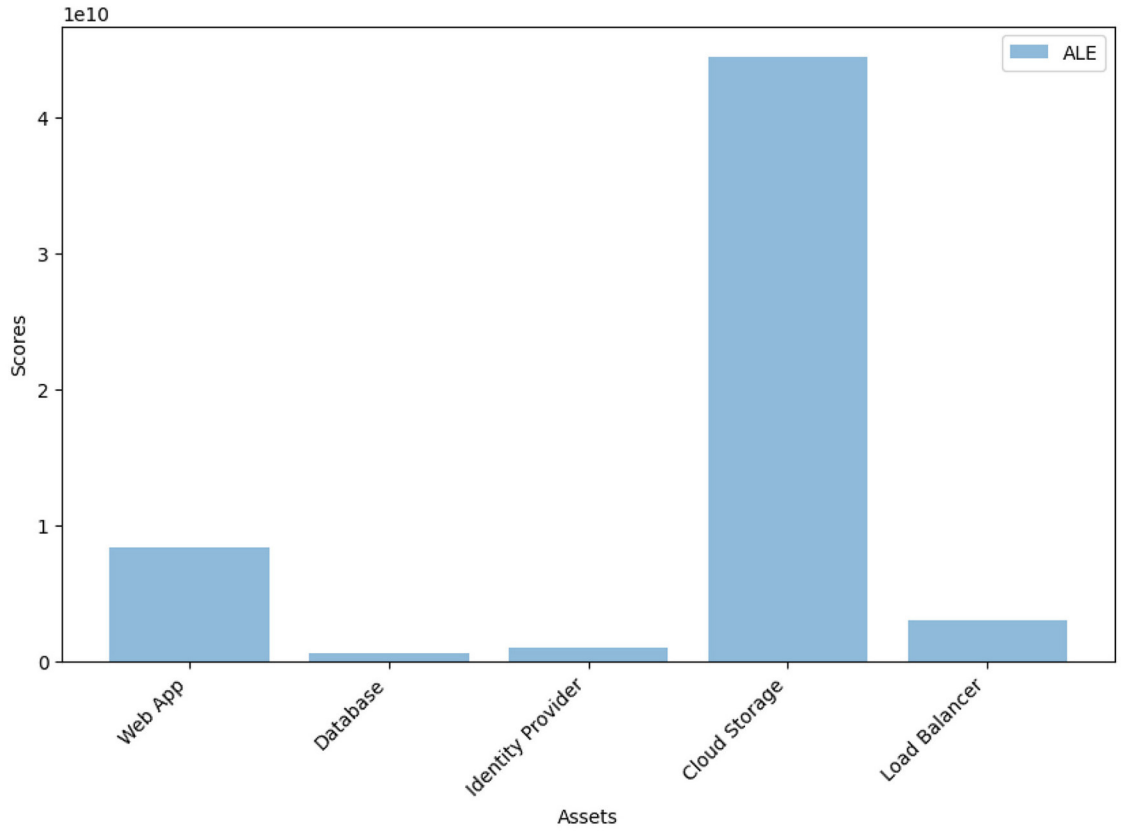

FIGURE 6
Individual ALE based on impact, likelihood, and depth.

4.1 From theory to practice: a real-world application

In the illustrative example, we have defined six assets—web server, database server, application server, load balancer, firewall, and VPN server—and three attack steps—phishing/malware attack, SQL injection/DdoS attack, and data exfiltration—along with their dependencies.

The following scenarios are discussed in this attack tree.

1. Phishing users and deploying malware

- In the first scenario, the attacker does reconnaissance (active or passive and gathers the data) of the user and the organization;
- Next, the attacker creates a specific attack using phishing based on the reconnaissance results (for example, by either embedding the malware in the file or attaching a malicious URL and sending that mail to the user);
- Once the user receives the file, the malware executable will be launched and will be installed on the user's machine.
- The malware will then try to elevate its privileges to become a local admin.
- Using local admin privileges, the malware will try to attack the active directory (AD) server to gain domain admin.
- With those admin rights, it does a lateral movement toward the intended machine where the data resides, which is of use to the attacker.
- The attacker is exfiltrating the data.

2. Attacking web application/applicative DDOS attack on the internet-facing application

After completing the reconnaissance, the attacker/red teamer may choose to perform an attack on the internet-facing application, which may result in either loss of data or exfiltration of data.

- There are two kinds of basic attacks possible at this point: volumetric attack and applicative attack.
- A volumetric attack is a type of distributed denial of service attack that aims to overwhelm a target's network bandwidth, essentially clogging the pipes that connect the targeted system to the internet. The attacker may choose to use multiple techniques such as HTTP flood, SYN Flood, or DNS amplification.
- Applicative DDoS attacks are Layer 7 attacks, which focus on overwhelming specific resources or functionalities within the application layer of a targeted service or server. These attacks include SQL injection, Brute Force, and XSS.

- The attacker may try to reach the application server behind the application, a vulnerability in the application.
- Once the attacker gains access to the application server, they try to reach the database server by creating or modifying transactions between the application server and the database.
- Once access to the database has been obtained, the attacker may either delete, alter, or exfiltrate the data.
- The attacker may also create a backdoor for maintaining their persistence and keep moving laterally from there to achieve their goals.

3. Cloud services attack

There are three attacks mentioned in this attack tree.

- In the first and most common kind, the ransomware initially compromises a victim's local device and then spreads to the cloud when their data syncs with a cloud storage service. This can be done by first phishing the user and then compromising the user's machine.
- In the second form of ransom cloud attack, criminals get direct access to an organization's cloud systems through phishing and then encrypt or extract their contents.
- The third kind of attack directly targets a particular cloud provider to gain access to its customers' data. "Attackers are putting a target on cloud providers because they know that if they can infect the provider's infrastructure, they can then encrypt huge amounts of customer data through a single infection".

4.2 The algorithm for the methodology

The algorithm for the methodology that resulted in the complete attack tree presented in Figure 4 is described below.

To address the dependencies, we have used an implementation that calculates the likelihood of an attack step being successfully executed as the minimum likelihood of the assets affected by the attack step. We have assumed the parameter values for the purpose of demonstrating the risk assessment in FAIR-modified attack trees. The values assumed are for the depth of resources in the network, countermeasure effectiveness, impact, likelihood of attack, and tangible value of the assets. The actual values of parameters shall be organization-specific but can be easily accommodated in our approach. These assumed values are based on some real-world application links to some of those that can be found during the research.

5 Results

We evaluate our proposed approach on a sample attack tree consisting of five assets and three attack scenarios. Our results show that the proposed approach leads to quantitative risk assessments, capturing the unique characteristics of each asset and its dependencies. As a callback to the gaps and limitations in particular, the known risk assessment systems, such as DREAD

```
FAIR RISK ASSESSMENT (impact, likelihood,
depth, cost, asset_value, countermeasures)
Function calculate_sle(impact, asset_value)
    {return impact * asset_value}
Function calculate_aro(likelihood)
{return 1.0/likelihood}
Function calculate_risk_score
{
# Normalize all parameters in the range of 0
to 1 impact_
    score = impact
    likelihood_score = (1.0 - likelihood)
* 10.0
    depth_score = depth/10.0
    cost_score = cost/100000.0
# Normalize all parameters
    countermeasures_score =
    sum([countermeasures[cm] for cm
    in countermeasures])/5.0
    return (impact_score * likelihood_score *
    depth_score * cost_score *
    countermeasures_score *
asset_value)
}
# Let assets be a dictionary of assets and
their properties. (Illustrative examples are
shown in the below code snippet)
# provides assets with a value to perform
the calculation, these values may be
different in the real environment
Let assets = {
    'Web App': {'type': 'on-premise',
    'depth': 3, 'impact': 500, 'likelihood':
    0.3, 'asset_value': 500000},
    'Database': {'type': 'on-premise',
    'depth': 2, 'impact': 200, 'likelihood':
    0.7, 'asset_value': 2000000},
    'Identity Server': {'type': 'cloud',
    'depth': 2, 'impact': 1000,
    'likelihood': 0.1,
     'asset_value': 1000000},
    'Cloud Storage': {'type': 'cloud',
    'depth': 4, 'impact': 8000,
     'likelihood': 0.9,
     'asset_value': 5000000},
    'Load Balancer': {'type': 'cloud',
    'depth': 4, 'impact': 1500,
     'likelihood':0.5,
     'asset_value': 1000000},
    .
    .
    .
    }
# Let attack_steps be a dictionary of attack
steps and their dependencies.
Let attack_steps = {
```

Algorithm 1. Continued

```
    'Step 1': {'depends_on': [], 'affects':
    ['Web App', 'Database',
    'Identity Server]},
    'Step 2': {'depends_on': ['Step 1'],
    'affects': ['Cloud Storage']},
    'Step 3': {'depends_on': ['Step 1'],
    'affects': ['Load Balancer']},
    }
# Let countermeasures be a dictionary of
countermeasures and their effectiveness.
Sample values are assumed here.
Let countermeasures = {
    'Firewall': 0.9,
    'Intrusion Detection System': 0.8, 'Data
    Encryption': 0.7,
    'Access Control': 0.6,
    'Security Information and Event
    Management': 0.5
    }
# Let combined_risk_scores and ales be
empty dictionaries.
Let cost = 200000
# Assumed cost of an attack (This shall
be variable)
# calculating Single Loss Expectancy(sle),
Annualized Rate of Occurrence(aro)
for asset, properties in assets:
    impact = properties['impact']
    likelihood = properties['likelihood']
    depth = properties['depth']
    asset_value = properties['asset_value']
    countermeasures = countermeasures
    sle = calculate_sle(impact, asset_value)
    aro = calculate_aro(likelihood)
    risk_score = calculate_risk_score
    (impact, likelihood, depth, cost,
    asset_value, countermeasures)
    ale = sle * aro
    combined_risk_scores[asset] = risk_score
    ales[asset] = ale
# Adjust likelihood score based on
attack steps
for step_name, step_properties
in attack_steps.items():
    if asset in step_properties['affects']:
# If an asset is affected by this step,
reduce the likelihood based on the minimum
likelihood of the affected assets
    affected_assets =
    step_properties['affects']
    min_likelihood =
    min([assets[a]['likelihood'] for a in
    affected_assets]) likelihood
    *= min_likelihood
    elif step_name in
    step_properties['depends_on']:
# If an asset is not affected but depends on
this step, reduce the likelihood based on
the complement of the minimum likelihood of
the affected assets
    affected_assets =
    step_properties['affects']
    min_likelihood =
    min([assets[a]['likelihood'] for a in
    affected_assets]) likelihood *= (1
    - min_likelihood)
#checking If the asset Is based on
onpremisis data center or cloud
    asset_risk_scores = {}
    for asset, properties in
    assets.items():
    if properties['type'] == 'on-premise':
    asset_risk_scores[asset] =
    calculate_risk_score() elif
    properties['type'] == 'cloud':
    asset_risk_scores[asset] =
    calculate_risk_score() * 0.5
# Calculate overall risk score for each
attack step step_risk_scores = {}
    for step, properties in
    attack_steps.items():
    step_risk_score = 0.0
    for asset in properties['affects']:
    step_risk_score +=
    asset_risk_scores[asset]
    step_risk_scores[step]
    = step_risk_score
# Calculate the aggregated risk score for
the entire attack tree
    aggregated_risk_score
    =sum(step_risk_scores.values())
}
```

Algorithm 1. Continued

Algorithm 1. Algorithm for the methodology.

(Microsoft Press, 2003), TRIKE,[1] or VAST (ThreatModeler, 2023), are limited to assessing the risk of individual vulnerabilities and do not capture the dependencies among different assets, which has been addressed in the FAIR-modified attack tree approach presented in this study.

We are particularly interested in a comparative analysis of quantifiable values such as risk scores of the assets, vis-a-vis the annual loss expectancy. For example, Figure 2 shows the individual asset risk scores and the annual loss expectancy based on the FAIR approach-based calculation combined with the attack tree methodology.

1 Trike. octotrike.org. Available online at: http://www.octotrike.org/ (accessed April 03, 2023).

5.1 Based on the assessment

- An organization may choose to reorganize the network to adjust the depth of some critical resources in the network.
- They may optimize annual loss expectancy by changing the effectiveness of countermeasures by restructuring the network security costs.
- Additional countermeasures may be commissioned for resources with higher ALE and risk scores.
- The organization may choose a more secure on-premise deployment of certain critical resources instead of hosting them on the cloud.
- Optimization of countermeasures or resource depth may also be carried out based on the identified attack steps.
- Additional optimization may be done by increasing the parameters in the overall risk calculation.

Optimizing overall organizational risk can be a tricky process, and this is where using the FAIR-modified attach trees helps in quantifying the different parameters associated with the risk assessment can help the analysts to simulate different scenarios, and choose the best optimization plan within the organizational budget.

A preliminary threat modeling application (Rana et al., 2023b) based on the above work is made available online by the authors. The streamlined application is a work in progress and allows the users to submit their own attack trees for risk assessment. The application uses computer vision to read differential risk assessments based on color-coded nodes to assess the overall risk associated with the submitted attack tree.

6 Conclusion and future work

In this study, we propose a novel approach for assessing risk in attack trees based on the FAIR approach. Our results demonstrate the effectiveness of this approach in capturing the unique characteristics of different assets and their dependencies, leading to quantifiable and more accurate risk assessments. It also aids in enhancing the process of risk management and improving its understanding. This study assigns numerical values to factors that impact risk, including impact, likelihood, depth, cost, asset value, and countermeasures, thereby making the calculation more precise and measurable and providing a comprehensive analysis. This process also ensures consistency each time an organization attempts to make a risk assessment and facilitates better decision-making.

Another advantage of the FAIR approach for organizations is that it enhances communication and reporting of risk with the attack tree, providing a straightforward approach to all relevant stakeholders and C-level executives. Therefore, its ability to offer a standardized framework, quantitative assessments, and assistance in decision-making makes it a valuable tool for identifying, prioritizing, and mitigating security risks effectively within complex attack scenarios within an organization.

Future work includes the extension of the proposed approach to larger and more complex attack trees and the integration of other risk assessment methods. It would also be feasible to suggest optimal countermeasures based on asset risk profiles combined with the organizational security budget. The proposed FAIR-modified attack trees have been implemented with a very simplistic assumption of higher risk in the cloud environment. The approach may be extended with a redefinition of asset properties to better reflect the cloud-based environment. For example, properties could include factors such as data sensitivity, accessibility, and compliance requirements. In addition, attack steps and their dependencies may also need to be revised to take into account the unique threats and vulnerabilities of a cloud-based network. For instance, the attack steps could include things like unauthorized access to cloud resources, data exfiltration, and denial of service attacks.

It may also be possible that the risk assessment model used in the program may need to be adjusted to account for the dynamic and distributed nature of cloud-based networks. This could involve using machine learning algorithms to identify anomalous behavior and potential security incidents in real time, as well as incorporating threat intelligence feeds and security information and event management (SIEM) solutions.

Finally, the program may need to be updated to reflect the multi-cloud and hybrid cloud environments that are becoming increasingly common in enterprise settings. This could involve adding support for multiple cloud providers and incorporating additional security controls and monitoring tools to ensure consistent security across all cloud environments.

In conclusion, the proposed FAIR-modified attack trees approach offers a practical and effective way to assess risk in attack trees, providing a more accurate and comprehensive understanding of the overall risk of an attack. The proposed approach complements existing risk assessment methods and can be used for quantitative risk analysis and the optimization of organizational security goals.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

AR: Writing—original draft. SG: Writing—review & editing. BG: Writing—review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Acknowledgments

This paper and the research behind it would not have been possible without the exceptional support of my supervisors, SG

and BG for their enthusiasm, knowledge, and exacting attention to detail have been an inspiration and kept my work on track.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdulla, P. A., Cederberg, J., and Kaati, L. (2010). "Analyzing the security in the GSM radio network using attack jungles," in *Leveraging Applications of Formal Methods, Verification, and Validation, Vol. 6415*, eds T. Margaria, and B. Steffen (Berlin, Heidelberg: Springer Berlin Heidelberg), 60–74. Available online at: http://link.springer.com/10.1007/978-3-642-16558-0_8 (accessed October 18, 2010).

Archived docs (2009). *The STRIDE Threat Model*. Available online at: https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20) (accessed April 03, 2023).

Audinot, M., and Pinchinat, S. (2016). "On the soundness of attack trees," in *Graphical Models for Security, Vol. 9987*, eds B. Kordy, M. Ekstedt, and D. S. Kim (Cham: Springer International Publishing), 25–38. Available online at: http://link.springer.com/10.1007/978-3-319-46263-9_2

Audinot, M., Pinchinat, S., and Kordy, B. (2017). Is my attack tree correct? Extended version. *arXiv*. doi: 10.48550/arXiv.1706.08507

Bistarelli, S., Dall'Aglio, M., and Peretti, P. (2007). "Strategic games on defense trees," in *Formal Aspects in Security and Trust, Vol. 4691*, eds T, Dimitrakos, F. Martinelli, P. Y. A. Ryan, and S. Schneider (Berlin, Heidelberg: Springer Berlin Heidelberg), 1–15. Available online at: http://link.springer.com/10.1007/978-3-540-75227-1_1.

Bistarelli, S., Peretti, P., and Trubitsyna, I. (2008). "Analyzing security scenarios using defence trees and answer set programming," in *Electronic Notes in Theoretical Computer Science, Vol. 197*, 121–129. Available online at: https://linkinghub.elsevier.com/retrieve/pii/S1571066108000601

Bolivar, H., Jaimes Parada, H. D., and Roa, O. (2019). "Modeling Cloud Computing security scenarios through attack trees," in *2019 Congreso Internacional de Innovación y Tendencias en Ingenieria (CONIITI)* (Bogota), 1–6. Available online at: https://ieeexplore.ieee.org/document/8960763/

Bryans, J., Liew, L. S., Nguyen, H. N., Sabaliauskaite, G., Shaikh, S., and Zhou, F. (2020). "A template-based method for the generation of attack trees," in *Information Security Theory and Practice, vol. 12024*, eds M. Laurent, and T. Giannetsos (Cham: Springer International Publishing), 155–165. Available online at: http://link.springer.com/10.1007/978-3-030-41702-4_10

Edge, K. S., Dalton, G. C., Raines, R. A., and Mills, R. F. (2006). "Using attack and protection trees to analyze threats and defenses to homeland security," in *MILCOM 2006 - 2006 IEEE Military Communications Conference*, 1–7. Available online at: http://ieeexplore.ieee.org/document/4086696/

Fila, B., and Widel, W. (2020). "Exploiting attack–defense trees to find an optimal set of countermeasures," in *2020 IEEE 33rd Computer Security Foundations Symposium (CSF)* (Boston, MA), 395–410. Available online at: https://ieeexplore.ieee.org/document/9155095/

Fraile, M., Ford, M., Gadyatskaya, O., Kumar, R., Stoelinga, M., and Trujillo-Rasua, R. (2016). "Using attack-defense trees to analyze threats and countermeasures in an ATM: a case study," in *The Practice of Enterprise Modeling, Vol. 267*, eds J. Horkoff, M. A. Jeusfeld, and A. Persson (Cham: Springer International Publishing), 326–334. Available online at: http://link.springer.com/10.1007/978-3-319-48393-1_24

Freund, J., and Jones, J. (2015). Measuring and managing information risk: a FAIR approach. Amsterdam: Butterworth-Heinemann. Available online at: https://dl.acm.org/doi/pdf/10.5555/2769769

Gupta, S., Gupta, B., and Rana, A. (2023). "A comparative cost analysis of organizational network security test lab setup on cloud versus dedicated virtual machine," in *Smart Trends in Computing and Communications, Vol. 396*, eds Y. D. Zhang, T. Senjyu, C. So-In, and A. Joshi (Singapore: Springer Nature Singapore), 623–632. Available online at: https://link.springer.com/10.1007/978-981-16-9967-2_58

Hyder, B., and Govindarasu, M. (2022). "A novel methodology for cybersecurity investment optimization in smart grids using attack-defense trees and game theory," in *2022 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT)* (New Orleans, LA), 1–5. Available online at: https://ieeexplore.ieee.org/document/9817467/

Ingoldsby, T. R. (2013). *Attack Tree-based Threat Risk Analysis* (PhD thesis). Available online at: https://www.amenaza.com/downloads/docs/AttackTreeThreatRiskAnalysis.pdf

Jürgenson, A., and Willemson, J. (2010). "Serial model for attack tree computations," in *Information, Security and Cryptology – ICISC 2009, Vol. 5984*, eds D. Lee, and S. Hong (Berlin, Heidelberg: Springer Berlin Heidelberg), 118–128. Available online at: http://link.springer.com/10.1007/978-3-642-14423-3_9

Kaiser, B., Liggesmeyer, P., and Mackel, O. (2004). "A new component concept for fault trees," in Eighth *Australian Workshop on Safety Critical Systems and Software (SCS 2003), Vol. 33* (Canberra, ACT), 37–46. Available online at: https://crpit.scem.westernsydney.edu.au/confpapers/CRPITV33Kaiser.pdf

Kordy, B., Kordy, P., Mauw, S., and Schweitzer, P. (2013a). "ADTool: security analysis with attack–defense trees," in *Quantitative Evaluation of Systems, Vol. 8054*, eds K. Joshi, M. Siegle, M. Stoelinga, and P. R, D'Argenio (Berlin, Heidelberg: Springer Berlin Heidelberg), 173–176. Available online at: http://link.springer.com/10.1007/978-3-642-40196-1_15

Kordy, B., Mauw, S., Radomirovic, S., and Schweitzer, P. (2014). Attack-defense trees. *J. Logic Comp.* 24, 55–87. doi: 10.1093/logcom/exs029

Kordy, B., Mauw, S., and Schweitzer, P. (2013b). "Quantitative questions on attack–defense trees," in *Information Security and Cryptology – ICISC 2012, vol. 7839*, eds Kwon, T. M.-K. Lee, and D. Kwon (Berlin, Heidelberg: Springer Berlin Heidelberg), 49–64. Available online at: http://link.springer.com/10.1007/978-3-642-37682-5_5

Kumar, R., Ruijters, E., and Stoelinga, M. (2015). "Quantitative attack tree analysis via priced timed automata," in Formal Modeling and Analysis of Timed Systems, vol. 9268, eds S. Sankaranarayanan, and E. Vicario (Cham: Springer International Publishing), 156–171. Available online at: https://link.springer.com/10.1007/978-3-319-22975-1_11

Kumar, R., and Stoelinga, M. (2017). "Quantitative security and safety analysis with attack-fault trees," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)* (Singapore), 25–32. Available online at: http://ieeexplore.ieee.org/document/7911867/

Lallie, H. S., Debattista, K., and Bal, J. (2002). A review of attack graph and attack tree visual syntax in cyber security. *Comp. Sci. Rev.* 35:100219. doi: 10.1016/j.cosrev.2019.100219

Mauw, S., and Oostdijk, M. (2006). "Foundations of attack trees," in *Information Security and Cryptology - ICISC 2005, Vol. 3935*, eds D. H. Won and S. Kim (Berlin, Heidelberg: Springer Berlin Heidelberg), 186–198. Available online at: http://link.springer.com/10.1007/11734727_17

Microsoft Press (2003). *Improving Web Application Security: Threats and Countermeasures*. Lisbon.

MITRE (2023). *News & Insights*. Available online at: https://www.mitre.org/news-insights (accessed April 03, 2023).

Morana, M. M., and Uceda Vélez, T. (2015). *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. Hoboken, NJ: Wiley.

Pieters, W., Hadziosmanovic, D., Lenin, A., Montoya, L., and Willemson, J. (2014). "TREsPASS: plug-and-play attacker profiles for security risk analysis," *2014 IEEE Symposium on Security and Privacy* (San Jose, CA), 4–19.

Pièrre-Cambacédès, L., and Bouissou, M. (2010). "Beyond attack trees: dynamic security modeling with Boolean Logic Driven Markov Processes (BDMP)," *2010 European Dependable Computing Conference* (Valencia), 199–208.

Poolsappasit, N., Dewri, R., and Ray, I. (2012). Dynamic security risk management using Bayesian Attack Graphs. *IEEE Trans. Dependable Secure Comput.* 9, 61–74. 10.1109/TDSC.2011.34

Rana, A., Gupta, S., and Gupta, B. (2023a). "A dual attack tree approach to assist command and control server analysis of the red teaming activity," in *Advances in Cybersecurity, Cybercrimes, and Smart Emerging Technologies, Vol. 4*, eds A. A. Abd El-Latif, Y. Maleh, W. Mazurczyk, M. ELAffendi, and M. I. Alkanhal (Cham: Springer International Publishing), 55–68. Available online at: https://link.springer.com/10.1007/978-3-031-21101-0_5

Rana, A., van Dijk, V., Gupta, S., and Gupta, B. (2023b). *Threat Modeling Risk Calculator*. Available online at: https://securityscientist-threat-model-dxf-home-sdtk0e.streamlit.app/

Roy, A., Kim, D. S., and Trivedi, K. S. (2012). Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees: ACT: unifying constructs of attack and defense trees. *Sec. Comm. Netw.* 5, 929–943. doi: 10.1002/sec.299.

Schneier, B. (2015). "Attack trees," in *Secrets and Lies* (Indianapolis, IN: Wiley Publishing, Inc.), 318–333. Available online at: https://onlinelibrary.wiley.com/doi/10.1002/9781119183631.ch21

Sheyner, O., Haines, J., Jha, S., Lippmann, R., and Wing, J. M. (2002). "Automated generation and analysis of attack graphs," in *Proceedings 2002 IEEE Symposium on Security and Privacy* (Berkeley, CA), 273–284. Available online at: http://ieeexplore.ieee.org/document/1004377/

ThreatModeler. (2023). *Threat Modeling Methodologies: What is VAST?* Available online at: https://threatmodeler.com

Weiss, J. D. (1991). "A system security engineering process," in *Proceedings of the 14th National Computer Security Conference, Vol. 2*, 572–581. Available online at: https://csrc.nist.gov/csrc/media/publications/conference-paper/1991/10/01/proceedings-14th-national-computer-security-conference-1991/documents/1991-14th-ncsc-proceedings-vol-2.pdf

Whitley, J. N., Phan, R. C.-W., Wang, J., and Parish, D. J. (2011). Attribution of attack trees. *Comp. Elect. Eng.* 37, 624–628. doi: 10.1016/j.compeleceng.2011.04.010

Yager, R. (2006). OWA trees and their role in security modeling using attack trees. *Inf. Sci.* 176, 2933–2959. doi: 10.1016/j.ins.2005.08.004.

Zonouz, A., Khurana, H., Sanders, W. H., and Yardley, T. M. (2009). "RRE: a game-theoretic intrusion response and recovery engine", in *Proceedings of the 2009 IEEE/IFIP International Conference on Dependable Systems and Networks* (Lisbon) 439–448.