Towards a Usable Attack Graph for Safety and Security*

Tim Zander, Jürgen Beyerer

We revisit a mathematical framework for estimating risk of safety and security, which describes risk in the context of safety and security problems quantitatively and integratively. We will discuss this framework in the context of other literature. We identify similar ideas and solutions that help advance the framework by adding graph structure. Further, we discuss challenges and opportunities for application of these theories.

A. Introduction

Safety and security share many commonalities. Nevertheless, measures and systems to provide and ensure safety and security are planned and implemented often independently by different experts[1]. If both aspects were treated in an integrated manner, synergies could be realized, and costs could be reduced. If we want to ensure the safety and security of such complex systems as critical infrastructures and complex socio-technical systems, many disciplines will be stakeholders: engineering, law, economics, humanities, social sciences, etc. Still, there is no common formal language that fits all approaches; meaning that there is no common formal language concerning safety and security and no common language across all involved disciplines. This paper aims to discuss quantitative mathematical approaches from the literature and enhance them a bit to serve to describe and analyse safety and security problems in a unified fashion and to plan and optimize dedicated measures and systems.

[*] This work was supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

[1] Sara Sadvandi, Nicolas Chapon, and Ludovic Piètre-Cambacédès, ‘‘Safety and Security Interdependencies in Complex Systems and SoS: Challenges and Perspectives” (Omar Hammami, Daniel Krob, and Jean-Luc Voirin eds, Springer Berlin Heidelberg 2012); Giedre Sabaliauskaite and Aditya P Mathur, “Aligning Cyber-Physical System Safety and Security” (Michel-Alexandre Cardin and others eds, Springer International Publishing 2015).

The paper "A Framework for a Uniform Quantitative Description of Risk with Respect to Safety and Security"[2] established a quantitative formulation of risk (which we refer to as UQDR from now on). Uncertainties were modelled as probabilities, which are interpreted as degrees of belief (DoB). This is due to the risks of individuals (intelligent agents) being described from their entirely subjective views. Individuals draw their decisions based on their subjective assessments of potential costs and frequencies of event occurrence with potential biases in their estimation. The three roles sources of danger D, subjects of protection S, and protectors P were used for describing different entities in the framework. Sources of danger are endowed with a DoB distribution describing the probability of occurrence and are partitioned into subsets of random causes, carelessness, and intention.

A set of flanks of vulnerability F was assigned to each subject of protection. These flanks characterize different aspects of vulnerability, including mechanical, physiological, informational, economic, reputational, psychological vulnerabilities. The flanks of vulnerability are endowed with conditional DoBs that describe to which degree an incidence or an attack will be harmful. Additionally, each flank of vulnerability was endowed with a cost function that quantifies the costs that are charged to the subject of protection. Additionally, we will introduce the notion of multi-stage attack in this paper. Where an initial attack might be successful, such as gaining non-privileged remote user access to an office system. Only a secondary attack might lead to access to the industrial network, where a production system could be damaged[3]. Hence, we introduce in this paper a directed graph structure to the flanks of vulnerability, where one broken flank opens new flanks.

There are many methods in the literature of a graph or tree view of vulnerabilities in safety and security and its algorithm for finding solutions. Among those are techniques of probabilistic risk analysis such as fault and event trees[4] and that of (cyber-)security, such as attack trees and graphs[5]

[2] Jürgen Beyerer and Jürgen Geisler, "A framework for a uniform quantitative description of risk with respect to safety and security" (2016) 1 European Journal for Security Research 135.

[3] Markus Karch and others, "CrossTest: a cross-domain physical testbed environment for cybersecurity performance evaluations" (2022).

[4] TJ Bedford and R Cooke, Probabilistic risk analysis: foundations and methods (Cambridge University Press April 2001).

[5] Mohsen Khouzani, Zhe Liu, and Pasquale Malacaria, "Scalable min-max multi-objective cyber-security optimisation over probabilistic attack graphs" (2019) 278(3)

and its automatic generation[6]. Moreover, there exists work in which combines fault and attack trees[7].

The calculated risk in UQDR was balanced against the cost of protection measures, or in the case of a rational attacker, it would balance the benefit of a specific attack against its cost. We will discuss challenges that arise from this subjective view. As individual agents will choose the cost-optimal solutions, this often leads to worse general utility, as sometimes a protection measure is only effective if enough people commit to it, and then an attack could become completely unprofitable. There is often an imbalance between producers of digital goods and their users. The first is richly rewarded for innovations that carry with them heightened security risks, and the latter bears the majority of these risks. This moral hazard leads to the necessity that certain security measures should be enforced by regulation[8].

In the UQDR framework, challenges of the determination of the cost functions were discussed. Especially the estimation of the probabilities (DoBs) of the model. We revisit this in the context of existing Bayesian approaches for safety and security. Bayesian approach for probabilistic risk assessment is a well-established approach[9] and is used in applications such

European Journal of Operational Research 894;b Tadeusz Sawik, ‘‘Selection of optimal countermeasure portfolio in it security planning’’ (2013) 55(1) Decision Support Systems 156; Mohsen Khouzani and others, ‘‘Efficient numerical frameworks for multi-objective cyber security planning’’ (2016); Teodor Sommestad, Mathias Ekstedt, and Hannes Holm, ‘‘The cyber security modeling language: a tool for assessing the vulnerability of enterprise system architectures’’ (2012) 7(3) IEEE Systems Journal 363; Nathaporn Poolsappasit, Rinku Dewri, and Indrajit Ray, ‘‘Dynamic security risk management using bayesian attack graphs’’ (2011) 9(1) IEEE Transactions on Dependable and Secure Computing 61; Lei Wang and others, ‘‘An attack graph-based probabilistic security metric’’ (2008); Hatem M Almohri and others, ‘‘Security optimization of dynamic networks with probabilistic graph modeling and linear programming’’ (2015) 13(4) IEEE Transactions on Dependable and Secure Computing 474.

[6] Alyzia-Maria Konsta and others, ‘‘Survey: Automatic generation of attack trees and attack graphs’’ (2024) 137 Computers & Security 103602 ⟨https://www.sciencedirect.com/science/article/pii/S0167404823005126⟩.

[7] E Andre and others, ‘‘Parametric Analyses of Attack-Fault Trees’’ (IEEE Computer Society June 2019) ⟨https://doi.ieeecomputersociety.org/10.1109/ACSD.2019.00008⟩; Rajesh Kumar and Mariëlle Stoelinga, ‘‘Quantitative Security and Safety Analysis with Attack-Fault Trees’’ (January 2017).

[8] Jeffrey Vagle, ‘‘Cybersecurity and Moral Hazard’’ (2020) 23 Stanford Technology Law Review ⟨https://ssrn.com/abstract=3055231⟩.

[9] Dana Kelly and Curtis Smith, Bayesian inference for probabilistic risk assessment: A practitioner’s guidebook (Springer Science & Business Media 2011).

as deep water drilling operations[10]. A combined risk estimation of safety and security for process industries with Bayesian networks was done in[11] for security alone, efficient algorithms for solving Bayesian Stackelberg games have been found.[12] Moreover, this has been applied to network security to optimally decide which initial security measures to take and which are the optimal online measures to take while receiving signals.

Challenges arise for the costs of certain security or safety measures or the cost of successful attacks or incidents. For that, the opinion of experts can be a viable tool to get valuable data to fit into a model. The optimal combination of multiple expert opinions is exceptionally useful researched field[13]. With recent advances in large language models, for some cases, it might be an expert on its own[14] and has been shown to help estimate some values[15]. We will discuss how this can be useful in the UQDR.

Often, it is also useful to directly influence the attackers to believe via some deterrence signal such as that of insider threat[16] or other[17].

[10] Jyoti Bhandari and others, ‘‘Risk analysis of deepwater drilling operations using Bayesian network'' (2015) 38 Journal of Loss Prevention in the Process Industries 11 ⟨ https://www.sciencedirect.com/science/article/pii/S0950423015300188⟩.

[11] Priscilla Grace George and VR Renjith, ‘‘Evolution of Safety and Security Risk Assessment methodologies towards the use of Bayesian Networks in Process Industries'' (2021) 149 Process Safety and Environmental Protection 758.

[12] Praveen Paruchuri and others, ‘‘Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games.'' (Lin Padgham and others eds, IFAAMAS 2008) ⟨http://dblp.uni-trier.de/db/conf/atal/aamas2008-2.html#ParuchuriPMTOK08⟩.

[13] Robert T Clemen and Robert L Winkler, ‘‘Combining Probability Distributions From Experts in Risk Analysis'' (1999) 19(2) Risk Analysis 187 ⟨https://ideas.repec.org/a/wly/riskan/v19y1999i2p187-203.html⟩.

[14] Siru Liu and others, ‘‘Assessing the Value of ChatGPT for Clinical Decision Support Optimization'' [2023] medRxiv ⟨ https://www.medrxiv.org/content/early/2023/02/23/2023.02.21.23286254⟩.

[15] Michael Haman, Milan Školník, and Michal Lošťák, "AI dietician: Unveiling the accuracy of ChatGPT's nutritional estimations" (2024) 119 Nutrition 112325 ⟨ https://www.sciencedirect.com/science/article/pii/S0899900723003532⟩.

[16] William Casey and others, ‘‘Compliance signaling games: toward modeling the deterrence of insider threats'' (2016) 22(3) Computational and Mathematical Organization Theory 318 ⟨ http://dx.doi.org/10.1007/s10588-016-9221-5⟩.

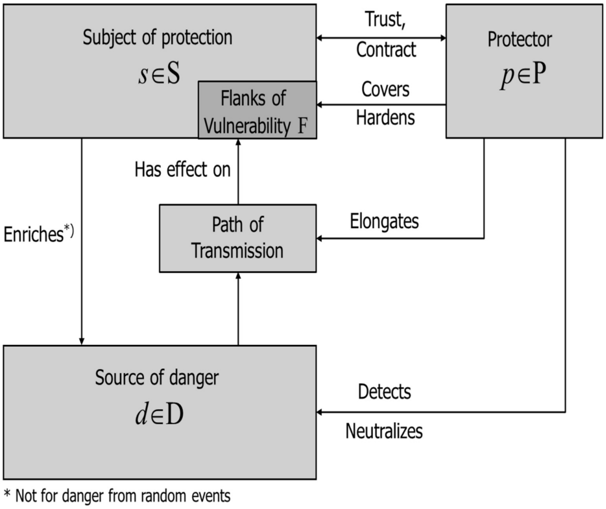
[17] NJ Ryan, ‘‘Five Kinds of Cyber Deterrence'' (2017) 31(3) Philosophy & Technology 331 ⟨http://dx.doi.org/10.1007/s13347-016-0251-1⟩.

B. Attack Graph model of safety and security

In the UQDR framework, the modelling included simple flanks of vulnerabilities that could be breached, and some damage could occur. We extend the model by introducing an attack graph where the flanks are now the edges of a graph, and the nodes are the elevated states of an attacker. These states could be increased privileges in a computer network. But also such things as stolen or forged access cards on the streets or an attacker who hid in a cabinet till the office closed. Some flanks require no elevated state but can be exploited directly, such as a distributed denial of service attack.

In the UQDR framework, the vulnerability with respect to attacks α or incidents i on flank$f \in F$ of $d \in D$ was modelled as a DoB-density with some degree of success β, if α or i hits s via$f$. Attacking system s via flank$f$ with success β incurs a cost $c(s, f, \beta) \in [0, \infty)$

Figure 1: Flow graph of the conceptual role model as introduced in UQDR[18].



[18] (Beyerer and Geisler [n 2]).

We build up on the ideas of Yunxiao Zhang and Pasquale Malacaria[19], where a Bayesian Stackelberg game on attack graphs with preventive security portfolio was defined. Some choice of security controls, such as online ones, could mitigate the probabilities of attacks.

I. Attack-graph

Precisely, we define a probabilistic attack graph similar to that existing in literature[20] where $G = (A, V, E, h, t, p, s, T, M)$ is a directed multi-graph where:

- A: is a set of attackers. (This was not defined before.)
- V: is the set of vertices (or nodes); a privileged state of an attacker in the organization.
- E: is the multi-set of directed edges. Note that there can be multiple edges between two vertices, corresponding to different atomic attacks between two attackers' privilege states. Equivalently, an edge e can be represented by the ordered triplet $e = (i, j, k)$, where i and j are the tail and head of the edge, and k is its index among all such edges that go from i to j.
- h: $E \to V$: returns the head node of an edge.
- t: $E \to V$: returns the tail node of an edge.
- p: $E \times A \to (0,1]$: defines the conditional success probability for an attacker to progress from one privileged state to another using a specific attack step. If an attacker has reached privilege state i and aims to advance to state j using attack stepe, where $j = h(e)$ and $i = t(e)$, then the likelihood of successful advancement is represented by p_e. Until then, the values for p_e are assumed to be known.
- $s \in V$: one of the vertices labelled as source, specifying the initial privilege state of an attacker.
- $T \subset V$: a subset of the vertices labelled as targets (or sink vertices). These are the privilege states or final attacks (e.g. deletion of all the data on the

[19] Yunxiao Zhang and Pasquale Malacaria, ‘‘Bayesian Stackelberg games for cyber-security decision support” (2021) 148 Decision Support Systems 113599 ⟨https://www.sciencedirect.com/science/article/pii/S0167923621001093⟩.

[20] Khouzani, Liu, and Malacaria (n 5).

computer network[21] or destruction of the machinery[22]) that constitute the potential goal of an attacker.

- $M: V \rightarrow S$: a membership function that assigns the ownership of a node to a subject. (This was not defined before.)

Note that this is indeed potentially a graph with cycles. For example, one might compromise a machine up to some user-level account. The administrator then deletes the attacker's account, which loses the attacker the privileges he has gained so far.

If the attacker successful reaches $v \in T$ similar as in UQDR incurs a cost $c(v) \in [0, \infty)$ on subject. Note that compared to the cost before, we replaced the flank with the node and got rid of the degree of success. If there is the need for such a degree of success, one can introduce multiple nodes, each representing some degree of success, and model the probability distribution of the success discretely via the conditional success probability p or, if needed, a success parameter β is added to the target nodes in T. With this, we essentially reproduce the expressibility of the original UQDR framework but can now express more complex problems with agents. It also extends the settings of the approach existing in literature[23] as now multiple agents control different parts of the security graph. This leads to a complex multiplayer game-theoretic situation.

Moreover, the owner or protector has a belief about the nodes attackers have breached (say some set $A \subset V$), and about the effectiveness of countermeasures at a certain cost (decrease of success probability p) and the costs to him when a node is breached ($c(v)$). If we give the node owner or its protector as in UQDR some security portfolio of countermeasures on some edges $E_\mathrm{r} \subset E$, then they can choose which measures to apply to harden the flanks. This portfolio $\hat{E}_\mathrm{r}$ can be represented as the set of all possible countermeasures, where each countermeasure is a tuple containing its effect on the success probability for an attacker α on an edge e:

$$\widehat{E}_\mathrm{r} = \{(e, p_r(e, a), c_r) : e \in (E_r), a \in A\}$$

[21] Oxford Analytica, ‘‘Cyberactivity in Ukraine signals Russian limits” [2022] (oxan-db) Emerald Expert Briefings.

[22] David Kushner, ‘‘The real story of stuxnet” (2013) 50(3) ieee Spectrum 48.

[23] Zhang and Malacaria (n 18).

to apply to harden the flanks. Meaning, that a countermeasur r on the edge e will reduce $p(e, a)$ to $p_r(e, a)$ but cost c_r. They can choose the best countermeasures in a two-player game situation, as with techniques introduced before by others[24]. Note that the belief about breached nodes can be incorporated into the probability p, as the detection of a privileged attacker or at least the presumption about one present through improved detection measures will influence its success probably of the following attacks. The membership function introduces the ownership of different nodes to different agents, which can be used to analyse more complex scenarios.

II. Risk in the attack graph

The DoB-risk of a member $m \in M$ can be calculated as follows. Let $V_m = \{v \in V\colon M(v) = m\}$, some belief-function of breaches $p_b\colon V \to [0,1]$ or more sophisticated some belief-function about multiple types of attackers at a node p_b: V → $[0,1]^A$. Moreover, let $\pi(v, a) \in \{0,1\}$ with $\alpha \in A$ be an indication function that attacker a has attacked and $\tilde{E} \subset \hat{E}$ multi-set of edges where the countermeasures are applied. The risk of m is the following;

$$\sum\nolimits_{v \in V} \sum\nolimits_{a \in A} c(v) \cdot p(v) \cdot \pi(v, a) + \sum\nolimits_{e \in \widetilde{E}} c_r \,.$$

If we take the approach as a multi-step game, then the unintended danger can be modelled in the form of an attacker where the $\pi(v, i)$ is always 1, meaning there is always the chance of such an event taking place. The DoB-probability $b_m\ (a, v)$ of $m \in M$ whether an attacker α has compromised node v is conditioned on the full history of all attacks of all attackers in the past. From that and his belief about the attacker function below, a belief about the next step of the attacker can be formed.

Now, on the attacker side, the attacker has certain knowledge about the attack graph. In fact, we replace the conditional success probability $p\colon E \to (0,1]$ with a belief $p_d\colon E \to [0,1]$ of the attacker d of the probability. For many attacks, such as a zero day's exploit[25], the ordinary attacker might not know about these attacks. The attacker also has a cost function for conducting an attack.

[24] Ibid.

[25] Leyla Bilge and Tudor Dumitraş, ‘‘Before we knew it: an empirical study of zero-day attacks in the real world" (2012).

In the UQDR framework, the costs of an attacker were described with $c_{\text{Effort}}(a, s, f)$ for the effort executing an attack α on s via$f$. $c_{\text{Penalty}}(s, f, \beta)$ described the penalty for being caught while conducting damage β and $g(s, f, \beta)$ was the gain of an successful attack of degree β. Now, in our new graph description, the cost of the attacker is $c_{\text{Effort}}(e)$ on $e \in E$ on the condition that the attacker has reached node h(e). Moreover, attacking and being caught has some penalty $c_{\text{Penalty}}(e)$ associated with it. The DOB-probability of being caught $Pr(\text{Penalty} \mid s, f, b) = 1 - Pr(\neg\text{Penalty} \mid s, f, b)$ becomes $Pr(\text{Penalty} \mid \text{nodes attacked till now})$. The reason the condition for nodes attacked till now is that while some notes, such as lock picking, might not leave any trace in some circumstances, other nodes, such as breaking a door to enter known at some point and countermeasures will be taken and the attacker is tried to be caught. The cost of an initial attack such as vulnerability scanning [26]might be very cheap to conduct. Moreover, such as with vulnerability scanning, the penalty cost might be even zero.[27] Additionally, there is a gain $G(a, t)$ for the attacker when they reach a node in $t \in T$.

The effort of an attacker a choosing attack path $P = (e_0, \ldots, e_l)$ in the attack graph is then described by the following formula;

$$\sum_{0 \le i \le l} c_{\text{Effort}}(e_0) \cdot \prod_{j < i} g(a, e_j),$$

meaning that the attacker only can conduct an attack if he gained access to the next node. The penalty for a path can be calculated as

$$\sum_{0 \le i \le l} Pr(\text{Penalty} \,|\, \text{nodes attacked till now}) \cdot c_{\text{Penalty}}(e_i) \cdot \prod_{j < i} g(a, e_j).$$

Finally, the gain of the attacker is

$$\sum_{0 \le i \le l,\, t(e_i) \in T} G(a, t(e_i)) \cdot \prod_{j < i} g(a, e_j).$$

Now, a rational attacker without countermeasure will attack if the sum of all these three costs is positive.

Ultimately, the game is played as follows. The node owners set up their security measures to reduce p on the edges leading to or from their nodes. Here is already a moral hazard at play, as the ones bearing the cost of the attack are the software's users further down the graph and not the software company

[26] Munawar Hafiz and Ming Fang, "Game of detections: how are security vulnerabilities discovered in the wild?" (2016) 21 Empirical Software Engineering 1920.

[27] Jamie O'Hare, Rich Macfarlane, and Owen Lo, "Identifying Vulnerabilities Using Internet-Wide Scanning Data" (January 2019).

further. Then, the attacker attacks and takes over some nodes. Again, the node owner applies countermeasures given their signal about compromised nodes, cost structure, their own cost, and so forth. So, the game is, in its most general form, a multi-leader multi-follower game with incomplete information[28]. Now additionally, the attacker also will try to improve their attacking strategy, i.e. a path or, more generally, a probability distribution of paths through the attack graph to maximize their gain. However, for many applications, it might be enough for single-leader multi-follower, multi-leader single-follower, or ordinary two-player Stackelberg games. For the most general form, it remains unclear if such a game will produce meaningful strategies to apply in the real world, even if there is a chance of finding very good ones with recent developments in reinforcement learning[29].

III. Example: Multiple Stakeholders

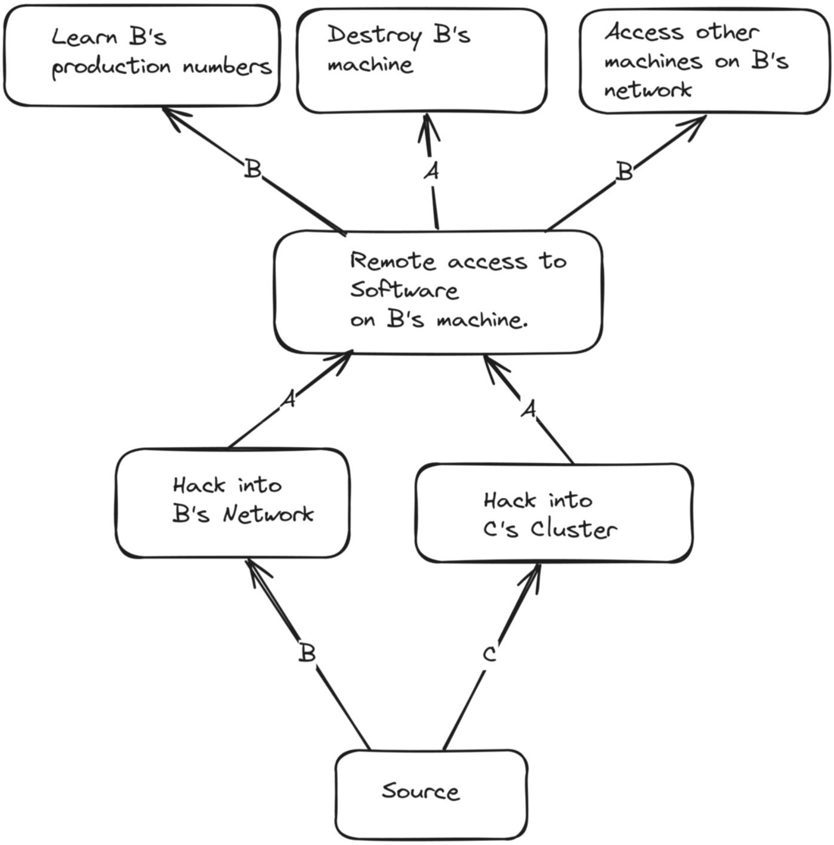
As we modelled the graph in a multi-agent way, we can now express scenarios with multiple node owners. For example, machine building company A sells machines with a certain AI functionality that needs remote access to a machine learning cluster owned by some company C. These machines come with a software vulnerability that would grant total control of the machine to an attacker who could access it over the network. Now, this machine is owned and run by Manufacturer B. Now, B has an incentive to fix this vulnerability, as a malicious person or a hack of company C could compromise the whole industrial network of C. Now, every one of these agents has their incentive, and potentially, A and C could have the incentive not to fix the security of their product, leaving B to fix the security. Which might be much harder and costly for B or might be impossible because all flanks till B's target are not in B's possession (see Figure 2). Moreover, if there are many machine owners just like B, then the average cost per machine owner might be a price everyone is

[28] Didier Aussel and Anton Svensson, ''A Short State of the Art on Multi-Leader-Follower Games'' in Stephan Dempe and Alain Zemkoho (eds), Bilevel Optimization: Advances and Next Challenges (Springer International Publishing 2020) ⟨https://doi.org/10.1007/978-3-030-52119-6_3⟩.

[29] Weichao Mao and Tamer Başar, ''Provably Efficient Reinforcement Learning in Decentralized General-Sum Markov Games'' [2022] Dynamic Games and Applications ⟨http://dx.doi.org/10.1007/s13235-021-00420-0⟩; Sailik Sengupta and Subbarao Kambhampati, ''Multi-agent Reinforcement Learning in Bayesian Stackelberg Markov Games for Adaptive Moving Target Defense'' (2020) abs/2007.10457 CoRR ⟨https://arxiv.org/abs/2007.10457⟩.

willing to pay. But then again, we are stuck on the problem of software's external effects. This means that with enough software users, the price for producing the bug fix is high, but the cost per copy is near zero[30].

Figure 2: Attack graph for the example of a machine with a software vulnerability. And the ownership of the edges, respectively, flanks denoted.


Learn B's production numbers
Destroy B's machine
Access other machines on B's network
B
A
B
Remote access to Software on B's machine.
A
A
Hack into B's Network
Hack into C's Cluster
B
C
Source

[30] Ross Anderson and Tyler Moore, ''Information Security Economics -- and Beyond'' (Alfred Menezes ed, Springer Berlin Heidelberg 2007).

In some other cases, the attacker might have a false belief in what type of attack he does. They might believe that they have gained privileged access to some computer system, which will lead to future gains. But instead, they might be trapped in a honey pot[31] or a scammer might believe he has some potential victim, but it is just some scam-baiter trying to fool the scammer[32]. All in all, the attacker has to choose its initial victim, and as outlined in the paper[33], if enough targets of the attacker are false positive, the profitability of the attacker will completely collapse. Or in terms of our attack graph, the attacker will try to estimate the success probability of a certain edge by doing such things as writing an unbelievable email or conducting a vulnerability scan. Now, the attacker has some belief about the probability of an attack being successful on edge e owned by member $m(h(e))$ and spends $c_{\mathrm{Effort}}(e)$ to do it. Now, there will be only very few edges that are worth attacking, but it completely relies on its strategy to improve the true-positive rate. Moreover, if this rate is low enough and the penalty high enough, the attack might be completely unprofitable[34] (see Figure 3). Moreover, if enough people commit to some countermeasures to some form of attacks, such as a car theft, the underlying economy such as that of car jacking might completely collapse (see Section 13.2.2 ‘‘Deterrence” of Ross Anderson’s Book Security Engineering[35]). The problem here lies in the incentive; the installation of countermeasures costs money, but normally, the risk is not big enough to make an effort or is externalized to a third party, such as an insurance company.

[31] Marcin Nawrocki and others, ‘‘A Survey on Honeypot Software and Data Analysis” (2016) abs/1608.06249 CoRR ⟨ http://arxiv.org/abs/1608.06249⟩.

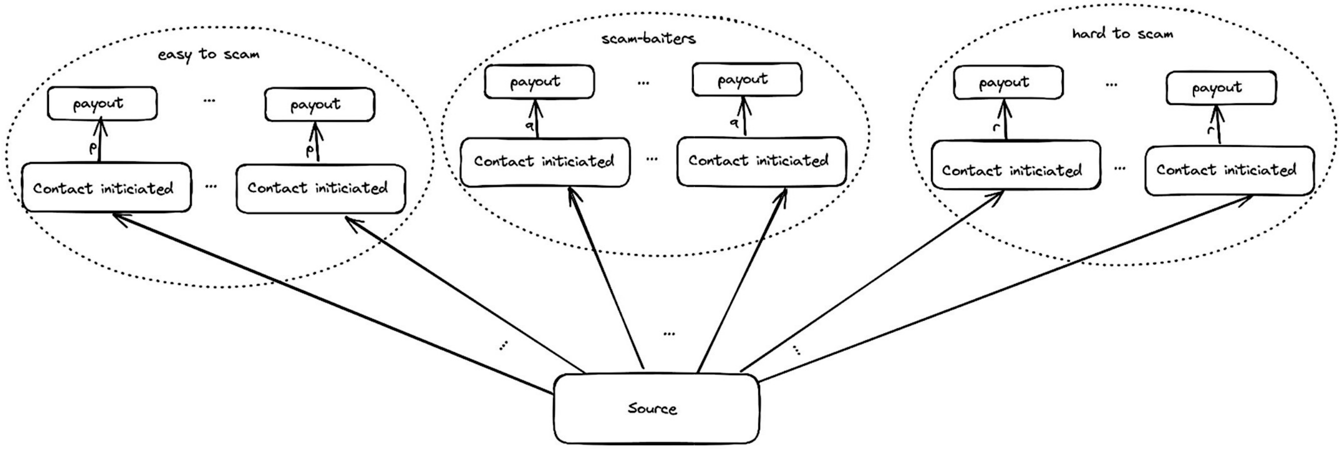
[32] Andreas Zingerle and Linda Kronman, ‘‘Humiliating Entertainment or Social Activism? Analyzing Scambaiting Strategies Against Online Advance Fee Fraud” (2013); Lauri Tuovinen and Juha Röning, ‘‘Baits and beatings: Vigilante justice in virtual communities” [2007] Proceedings of CEPE 397; Matthew Edwards, Claudia Peersman, and Awais Rashid, ‘‘Scamming the scammers: towards automatic detection of persuasion in advance fee frauds” (2017); Cormac Herley, ‘‘Why do Nigerian Scammers Say They are from Nigeria?” [2012] Proceedings of the Workshop on the Economics of Information Security.

[33] Herley (n 32).

[34] Ibid.

[35] Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems (3rd edn, Wiley 2020).

Figure 3: Attack graph of Advanced Fee Frauds of people responding to the scammer. The group to the left are people who are easily susceptible to such attacks and have a high success rate p after an initial contact is made. The group to the middle are scam-baiters, which have a very low success rate q to convert contact to money but want to mimic the left group. The groups to the right are hard to scam people with also a very low success rate r. The replies of this group can be avoided by making outrageous claims in the initial contact email. However, this also increases the number of scam-baiters, as they easily recognize these emails in their honey pot email addresses, or some of the right groups may become scam-baiters by chance out of interest.



IV. Granularity of the attack graph

Another effect we see is that there is often a specialization of certain attacks. The one conducting a distributed denial of service (DDoS) attack might not be the one that gives access to the devices involved in the first place. This increases the problem of effective measures against such problems. Increasing the punishment of the DDoS-attacker directly did little help to mitigate the problems, but forcing the administrators of the compromised servers to get rid of the access of the perpetrator did follow with a decrease of such attacks[36]. So, finding the right level in the attack graph to mitigate problems seems like the key to finding optimal utility for the common. As with the smart device, which will become the next DDoS device. Should the internet provider be forced to block any traffic from owned devices, or should the device manufacturer be held accountable, which might be non-existent anymore at the time when the device becomes a problem. For a single entity such as a company, the right security implementation might still be higher up in the attack graph as many nodes near the source s might be hard to fix for a single entity that is affected by the attacks.

We can also incorporate the safety aspect into the attack graph model. Certain attacks only become available when certain safety measures fail. For example, a power outage might cause a security camera system to shut down. So, an attacker can now sneak past the camera surveillance area without much risk. Or, because of the power outage, a remote admin might not be able to receive any info on the server they administer because they live in the countryside with a single power line reaching their house. While the server is still running, alerts of the server system fail to reach the administrator.

A general limitation of the attack graph is that it is non-suitable for doing fine-grained safety analysis as the graph will be too complex for a human to construct the graph and oversee the analysis. While there exists work that automatically constructs certain attack graphs[37], in many scenarios using other techniques may help reduce the overall complexity of a fault tree. The

[36] Ben Collier and others, ‘‘Influence, infrastructure, and recentering cybercrime policing: evaluating emerging approaches to online law enforcement through a market for cybercrime services” (2022) 32(1) Policing and Society 103.

[37] Ferda Özdemir Sönmez, Chris Hankin, and Pasquale Malacaria, ‘‘Attack Dynamics: An Automatic Attack Graph Generation Framework Based on System Topology, CAPEC, CWE, and CVE Databases” (2022) 123 Computers & Security 102938 ⟨https://www.sciencedirect.com/science/article/pii/S0167404822003303⟩.

fault tree may express a general Boolean statement[38] and to incorporate this into the graph structure, for example, any AND-statement would need to incorporate any subset of the atoms as a node in the graph. Which is, of course, the cardinality of the power set of the set of all atoms, which grows exponentially with the number of atoms.

V. Attack-fault trees

Because of the limitation just stated, one has to look at the attack graph at a subsystem-size granularity, as tracing any screw of every security camera attachment as a failure mode in the attack graph is infeasible. However, one can break down the subsystems in a fault tree. The more recent approach that one can use is one of the so-called attack-fault trees as described in research before[39], which are in these works connected to automata theory[40]. Stochastic timed automata (STA) were used in a paper[41] to do stochastic model checking.

They gave concrete examples, such as a fire safety door example, which highlighted the friction point between safety and security. A fire door might be used as an exit by the user of the building. This can already be a security risk, as intruders or insiders can steal stuff and then leave the building unnoticed through some fire exit. Also, such doors tend to be used as an exit for convenience, such as smoking, and grant re-entry by blocking the door from being closed with something. Also, people can easily use it as an entrance if enough people use it as an exit and if there is enough anonymity present. This helps an attacker to sneak in without passing by typical building access control such as a doorman. One solution could be to lock the fire door, weld it shut, or not construct any in the beginning. Which, sadly, can lead to a catastrophe in the event of a fire. The risk might still be taken by the owner to prevent immediate costs like stealing or extra safety measures[42]. A more typical solution in countries where fire safety rules tend to be enforced, apart from making the door only open from the inside, is to install alarms that either make a loud noise when the door is opened or

[38] Balbir S Dhillon and Chanan Singh, Engineering Reliability (Wiley series in systems engineering & analysis, John Wiley & Sons April 1981).

[39] Andre and others (n 7); Kumar and Stoelinga (n 7).

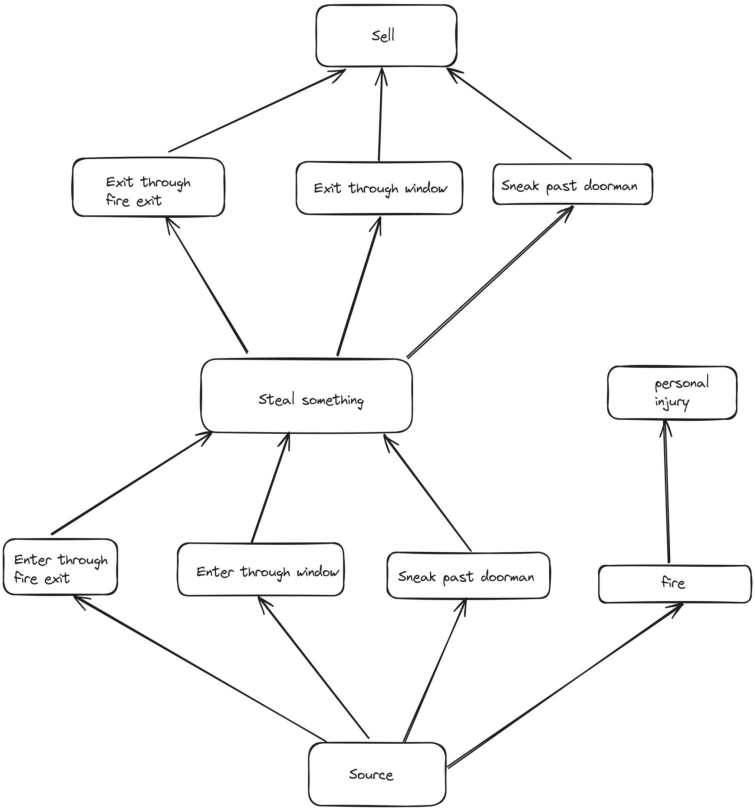
[40] Arto Salomaa, Theory of automata (Elsevier 2014).

[41] Kumar and Stoelinga (n 7).

[42] Margrethe Kobes and others, ‘‘Building safety and human behaviour in fire: A literature review” (2010) 45(1) Fire Safety Journal 1 ⟨https://www.sciencedirect.com/science/article/pii/S0379711209001167⟩.

trigger some (potentially silent) alarm in the security system. This prevents the door from being used for convenience and mitigates most attacks but keeps the fire exit usable in an emergency. However, cheap solutions tend not to be very robust against some forms of attacks, such as lock picking the latch. So, some sort of risk prevails in any case.

Figure 4: Security improvements such as having no fire door or locked fire doors such as having barred windows will greatly reduce the likelihood of the success of the four left-most attack paths. But this will also, for many buildings, greatly increase the risk on the right-most path, which ends in personal injuries.



In another paper[43] a new type of parametric timed automata (PTA) with (discrete) rational-valued weight parameters named parametric weighted timed automata (PWTA) was defined, and it was shown that attack-fault trees ‘‘equipped with an execution time and a rich cost structure that includes the cost incurred by an attacker and damage inflicted on the organization.” could be translated to these automata. The addition of time is quite a meaningful feature, as cracking some cryptography might take years, but the data it encrypts might only be valuable for a very restricted time frame. Or an automatic update might patch a vulnerability at midnight, so if the initial attack is not complete by that time, this path will be closed. Also, our cost functions can be thought of as multi-valued. So, some parameters are more constrained, and it is hard to find an extra budget. The administrator’s time budget might be limited, and another one might be out of the monetary budget. So should the administrator rather read logs and look for signs of intruders, or should they test and deploy updates to the server? There is no reason why the cost in the upper-defined attack graph can include a multi-valued cost structure.

Moreover, the paper also showed that such attack-fault trees can be translated to an acyclic-directed graph and solved with a model checker stemming from automata theory. This closes the loop, as now such graphs, can be thought of as a subgraph of the bigger attack graph. While this resulting graph might not be very accessible for a human, at least there is language describing the impact of a metal piece with poor tolerances in a small lock to the whole multiplayer system where this piece contributes to computerized reasoning. Moreover, with the rise of very capable large language models such as GPT4[44], there seem to be better chances than ever to build such attack graphs, covering the problem in great detail without the need for a great amount of skilled human labour[45].

C. Conclusion

This paper presents an enhanced mathematical framework heavily building on works of others for estimating safety and security risks within complex systems, integrating a probabilistic attack graph model with the use of

[43] Andre and others (n 7).

[44] OpenAI and others, GPT-4 Technical Report (2024).

[45] Farzad Nourmohammadzadeh Motlagh and others, Large Language Models in Cybersecurity: State-of-the-Art (2024).

the Unified Quantitative Description of Risk (UQDR) framework. This approach continues to model uncertainties as degrees of belief (DoB) and incorporates Bayesian statistical decision theory, game theory, and graph theory to provide a tool for the analysis of potential vulnerabilities and attack vectors.

The attack graph model is a directed multi-graph that outlines the stages of an attacker's progression through a system, including potential targets and the associated costs of attacks for both attackers and defenders. It includes functions for attack success probability, detection probability, mitigation controls, and the membership function assigning the ownership of nodes to subjects. This model allows for strategic planning and optimization of security measures but also highlights the problem of multiple stakeholders for optimal security. Risk is quantified by considering the costs of successful attacks and the effectiveness of security measures in this attack graph. For future work, we suggest focusing on methodologies for using Large Language Models to semi-automatically generate the structure of these attack graphs and to help estimate their parameters (e.g., costs, probabilities) from technical reports and expert interviews. This would address the key challenge of constructing these complex models manually and improve their practical applicability. The framework can express problems such as moral hazards and incentive misalignments, emphasizing the need for regulation to enforce security measures from the bottom up.

In Summary, the paper's contributions are an advanced probabilistic attack graph. With it we highlight the trade-offs between safety and security measures, the challenges of granularity, and confirm the potential for automated tools to assist in model construction. We underscore the importance of multi-stakeholder coordination and the integration of artificial intelligence to develop effective, practical security measures. Future research should further explore the practical application of this model, a wide range of data, and the effectiveness of deterrence strategies in reducing the likelihood and impact of attacks.