THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Towards Interdependent Safety Security Assessments Using Bowties

Citation for published version:
Arnaboldi, L & Aspinall, D 2022, Towards Interdependent Safety Security Assessments Using Bowties. in M Trapp, E Schoitsch, J Guiochet & F Bitsch (eds), *Computer Safety, Reliability, and Security. SAFECOMP 2022 Workshops - DECSoS, DepDevOps, SASSUR, SENSEI, USDAI, and WAISE, Proceedings.* Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13415 LNCS, Springer Science and Business Media Deutschland GmbH, pp. 211-229, Workshops on DECSoS, DepDevOps, SASSUR, SENSEI, USDAI, and WAISE, held in conjunction with the 41st International Conference on Computer Safety, Reliability, and Security, SAFECOMP 2022, Munich, Germany, 6/09/22. https://doi.org/10.1007/978-3-031-14862-0_16

Digital Object Identifier (DOI):
10.1007/978-3-031-14862-0_16

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Computer Safety, Reliability, and Security. SAFECOMP 2022 Workshops - DECSoS, DepDevOps, SASSUR, SENSEI, USDAI, and WAISE, Proceedings

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 04. Apr. 2023

Towards Interdependent Safety Security Assessments using Bowties

Luca Arnaboldi and David Aspinall

School of Informatics, University of Edinburgh, Edinburgh, UK
{luca.arnaboldi,david.aspinall}@ed.ac.uk

arXiv:2208.03484v1 [cs.CR] 6 Aug 2022

Abstract. We present a way to combine security and safety assessments using Bowtie Diagrams. Bowties model both the *causes* leading up to a central failure event and *consequences* which arise from that event, as well as *barriers* which impede events. Bowties have previously been used separately for security and safety assessments, but we suggest that a unified treatment in a single model can elegantly capture safety-security interdependencies of several kinds. We showcase our approach with the example of the October 2021 Facebook DNS shutdown, examining the chains of events and the interplay between the security and safety barriers which caused the outage.

Keywords: Safety and Security · Bowtie Diagrams · Risk Analysis.

1 Introduction

Structured graphical reasoning methods help the systematic discovery, assessment and mitigation of risks. A variety of methods have been used for security and for safety. While safety usually considers unintentional events and security deals with intentional malicious events, both safety and security assessments may consider causes and consequences of critical events. An early method is cause/consequence diagrams [23], commonly referred to as the "bowtie method" due to its graphical representation. A bowtie is formed of a tree of causes and a tree of consequences, fanning out to the left and right of a central critical (or *top*) event. Nodes in the tree represent causal or following events and *barriers* represent ways to prevent or mitigate events. Bowties have been used both for safety [27] and security [4] separately, and an initial investigation was even performed to inform a safety bowtie by an attack tree [1] in terms of risk quantification. They have many desirable properties, such as ease of visualisation [4] and the ability to quantify and calculate likelihoods of events, given a mathematical definition for a diagram's meaning.

In this paper we introduce a way to combine safety and security assessments together in a single bowtie, to enable reasoning about security and safety together. In particular we account for four different types of interplay between event types: safety and security can act independently, with conditional dependency, in concert, or in opposition [20]. An ideal structured reasoning approach should capture each of these.

2021 Facebook DNS outage. We use Facebook's recent global DNS outage as a running example, which nicely demonstrates some safety-security interplay [15,16]. On October 4th 2021, there was a complete outage of all Facebook apps and services (including WhatsApp, messenger, Instagram, etc.), which lasted approximately 6 hours. An initial small change to the DNS configuration triggered a complicated chain of events that made resolving a seemingly small issue overly complex and time consuming, and affected millions of people. This happened directly as an outcome of a security response disabling the ability for the safety engineers to patch the server remotely, forcing them to physically access the servers, which exacerbated the disruption.

A bowtie describing the sequence of events is presented in Fig. 1. In outline it shows that the DNS server-down event (the top event) could have a security cause or a safety cause. For example, a safety cause could be that a faulty update to the server configuration goes wrong; a barrier to stop this would be to check the patch doesn't cause the server to fail. A security cause might be that an attacker manages to access the DNS server and make a malicious update. If the DNS server does go down, we would like to find a path to a rapid fix to make the server operational again. But two fixes, a safety fix and a security fix, are *antagonistic* and act in opposition; only one of them should be undertaken. In the Facebook DNS incident, both fixes were implemented, disallowing remote access to the servers and delaying the response.

Obviously this bowtie is not a complete safety-security assessment (and was constructed after-the-fact). A more complete model could cover the many bad consequences that can occur from a central DNS outage and

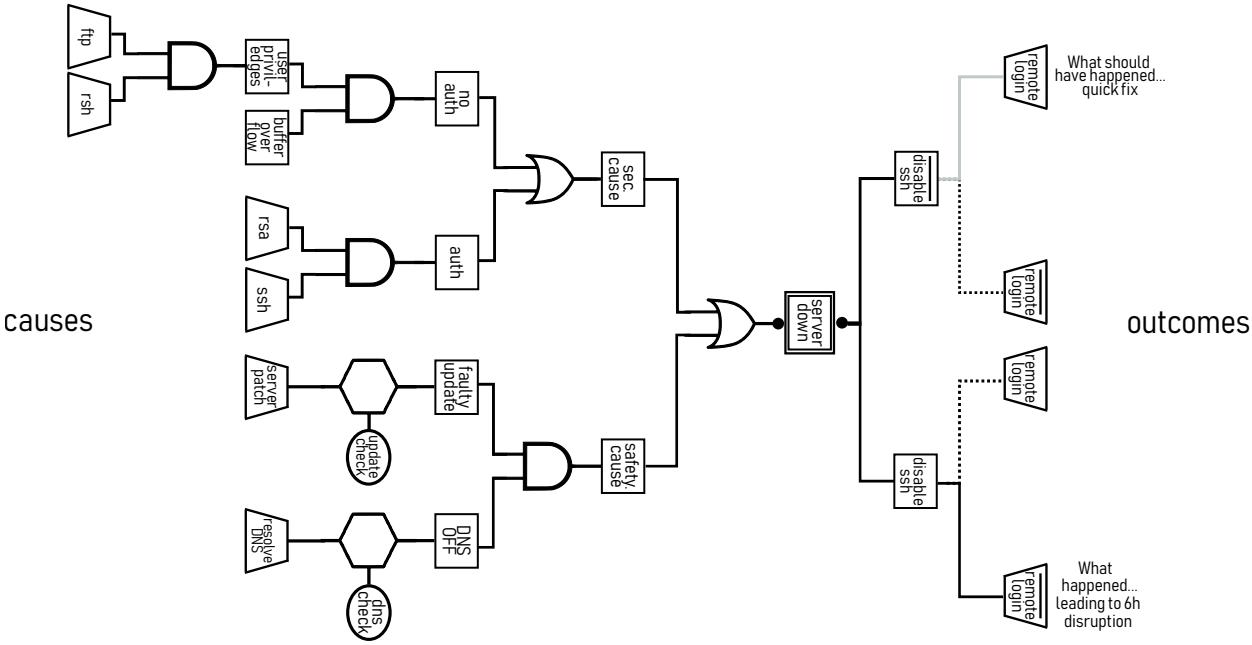

Fig. 1. High level bowtie diagram of the Facebook DNS outage (Tab. 2 expands the labels). The negative outcome is in black, whilst the grey path highlights the path that would avoid the negative outcome. Showing the antagonism between the security and safety responses.

ways those could be mitigated, as well as further ways such outages could occur. But this example serves to demonstrate how we can model some safety-security interdependences and we will give mathematical definitions which formalise this, returning to the example and variations of it in Sec. 4.

Outline. The rest of the paper is structured as follows: Sec. 2 contains background on diagrammatic representation of bowties, we note that further background is added as needed to the respective sections, Sec. 3 the formalisations of safety-security bowties, Sec. 4 the interdependence analysis and the formalisation of its operations, and in Sec. 5 we provide examples of related work, related assessment techniques, and discuss next steps for this work.

Contributions. The contributions of this work are the following:

1. A demonstration of security and safety interplay using bowtie diagrams and the Facebook DNS outage. Fig. 1 highlights the need to treat safety and security together, before a critical event and afterwards.
2. Expanding on recent work on combining and formalising safety-security trees as *disruption trees* [29], we add barriers (safety) and mitigations (security), modelling both response forms in the same formalism.
3. We also add a corresponding mechanism for outcomes, to give a new definition for safety-security bowtie diagrams. This provides a solid underlying basis for the graphical notation, paving the way for further theoretical analysis and discussion, and ultimately towards trustworthy tools to reason about and manipulate such diagrams.

This paper is a first step and we want to investigate more complex case studies as well as studying properties of the definitions, see Sec. 5 for more discussion of next steps.

2 Diagrammatic Representations

Our bowtie diagrams use Boolean logic gate notation to show relationships between events tracking the passage of failures from “inputs” to “outputs”. Boxes containing labels describe intermediate events.

Table 1. Symbols and Relationships in a fault tree

Event Types	
	Intermediate event - an identity function labelling outgoing edges.
	Prevention - restrictive conditions, must fail to progress
	LEAF - initiation event (BEs in Fault Trees) eventual outcome (in Event Trees)
Relationship Types	
	INHIBIT - output occurs if the LH occurs and prevention does not
	OR - output occurs if at least one of the inputs occurs
	AND - output occurs if both of the inputs occur

Faults and attacks in the same diagram. Security analysis traditionally uses attack trees [26] whilst Safety analysis makes use of fault trees [30]. One shows the steps which an attacker can take to achieve a goal whilst the other shows the events leading to a fault. Stoelinga, however, has recently argued that fault trees and attack trees can be treated uniformly in a single structure [29]. Tab. 1 shows the symbols we use, following the Fault Tree Handbook notation [30].

Barriers and defences treated uniformly. We further propose to also treat barriers and defences in the same logic. To capture actions or conditions that impede fault conditions arising or defences in an attack tree, we use an `INHIBIT` gate drawn as a hexagon. The orthogonal side input is the inhibiting condition which prevents the flow of the fault or prevents the attack.

Safety Security Bowties. Now that we have a unified tree notation for attack trees and fault trees, we can use event tree tree notation for consequences as usual. This stays true to the essence of bowties as previously defined [25]. The focal point of the assessment is the critical risk event (or "top event", following fault tree parlance). To the left of the event are potential causes of the risk and ways to reduce them; this is the *prevention assessment*. On the right on the other hand we observe the outcomes of the risk event known as the *consequence assessment*.

An example unlabelled bowtie diagram is shown in Fig. 2. These diagrams are a Boolean network which are static and capture only logical dependencies: a bowtie itself is not quantitative, but it is a qualitative model that may be analysed quantitatively by adding additional annotations representing risk assessments. This has been done by various authors for fault and attack trees, for example [10, 24]. For simplicity in the rest of this paper, we consider the basic logical interpretation only.

3 Formalisation of Safety-Security Bowties

Like typical bowties a safety-security bowtie will be composed of a *prevention assessment* and a *consequence assessment*. For the prevention assessment we use a *disruption prevention tree* which captures safety and security assessments, as well as attack and fault mitigations, which correspond to barriers in traditional bowties [25]. For the consequence assessment we use an event tree to show the consequences of the risk event and potential outcomes. The graphical representation from the figures includes intermediate events between gates, for enhanced understanding, but these are not part of the formalisation. We first showcase a general structure for trees we make use of for all subsequent formalisations in Sec. 3.1. We note that although, we are mostly concerned with trees, we adopt Directed Acyclic Graphs (DAGs) to allow input and outputs to be shared between gates.

3.1 Structure trees

Our models are based on generalised n-ary trees with additional structure given by a node type labeling.

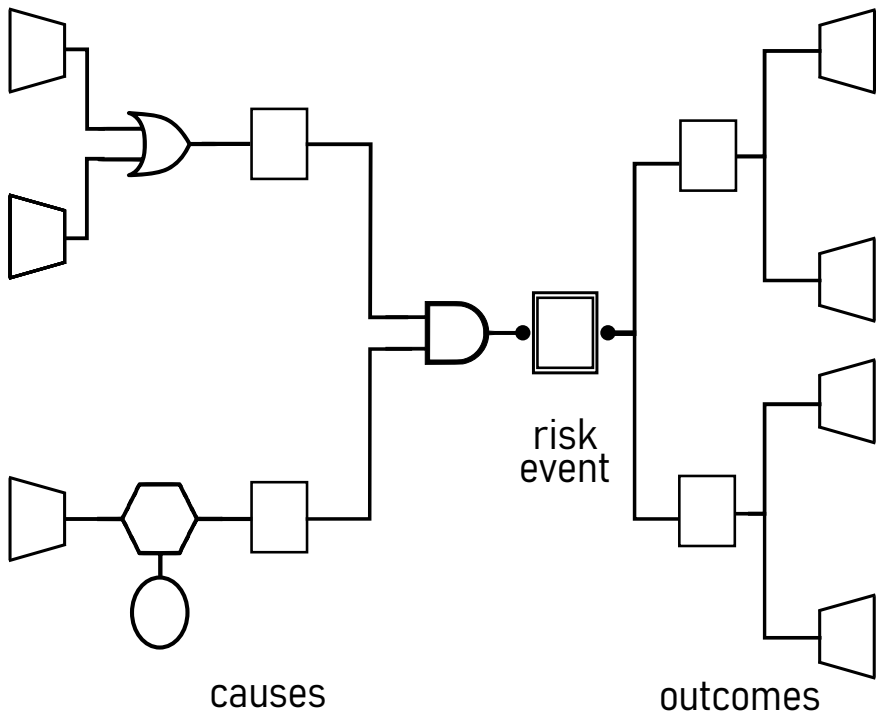

Fig. 2. Generic representation of Bowtie Diagram with causes on the left and outcomes on the right.

Definition 1 (Structure Tree). *A Structure Tree is a tuple* $T = \langle N, \mathbb{T}, t, ch \rangle$ *where:*

- N *is a finite set of nodes and* $\mathbb{T}$ *is a set of node types;*
- $t : N \to \mathbb{T}$ *gives the type of each node;*
- $ch : N \to N^*$ *gives the sequence of children of a node.*

Moreover, T *satisfies the following constraints:*

- (N, E) *is a connected DAG where* $E = \{ (v, u) \in N^2 \mid u \in ch(v) \}$
- T *has a unique root, denoted by* R_T : $\exists! R_T \in N. \forall v \in N. R_T \notin ch(v)$ *(where* $\exists!$ *stands for unique existence).*

Given a type $\texttt{LEAF} \in \mathbb{T}$, we write $\texttt{LEAF}_T = \{ v \in N \mid t(v) = \texttt{LEAF} \}$ for the nodes labelled with that type, and similarly for other types.

3.2 Disruption Trees

The notion of Disruption Trees (DTs) is introduced by Stoelinga as the uniform representation of static safety and security risk models [5, 29]. Although extensions exists to each of these assessments which do result in key differences, in the simplest setting with only `OR` and `AND` nodes, they coincide.[1] Following Def. 1 we can interpret Stoelinga's DTs as:

Definition 2 (Disruption Tree [29]). *A Disruption Tree is a structure tree according to Def. 1 with* $\mathbb{T} = \{ \texttt{LEAF}, \texttt{AND}, \texttt{OR} \}$ *and such that* $t(v) = \texttt{LEAF}$ *iff* $v \in \mathit{LEAF}_T$.

Semantics pin down the mathematical meaning of models. The basic semantics of a disruption tree is given by its *structure function* over Booleans $\mathbb{B} = \{1, 0\}$, which defines a function from leaf events to the

[1] There are subtleties in different ways to treat `OR` gates with quantified safety or security risks, which will not be considered here.

root in the obvious way. Structure Trees are more lax compared to models we usually want. Besides allowing useful DAG structure they include redundancies like nodes with repeated children. This is alright for `AND` and `OR` but would be confusing later on with gates that have some inputs (or outputs) being negated (or mutually exclusive). We call a structure tree (and later variants) *proper* if every child of a node is distinct, i.e., $|\{\, ch(v) \mid v \in N \,\}| = \mathit{length}(ch(v))$. From now on we will restrict to models built on proper structure trees.

3.3 Prevention Assessments

We turn a Disruption Tree into a Disruption Prevention Tree by adding a new gate type for `INHIBIT`.

Definition 3 (Disruption Prevention Tree). *A Disruption Prevention Tree (DPT) is a Disruption Tree with an additional binary node type* `INHIBIT`*, i.e.,* $\mathbb{T} = \{\, \texttt{LEAF}, \texttt{AND}, \texttt{OR}, \texttt{INHIBIT} \,\}$ *and* $|\, ch(v)| = 2$ *whenever* $t(v) = \texttt{INHIBIT}$*. The second child of an* `INHIBIT` *node corresponds to the inhibiting (prevention) condition.*

There is a similarity between Disruption Prevention Trees and Attack-Defence Trees. Formalisations of Attack-Defence Trees (see [18] for example) label nodes to represent their role in the model as a proponent (mitigating an attack or protecting against a fault event) or an opponent (actors causing damage leading to the root fault). It seems possible to recover this labelling from the root of DPT flipping the role when consider the prevention input of `INHIBIT` gates.

Structure of a DPT The semantics of a DPT allows a proponent to block an opponent. These kind of barriers are called "avoid barriers" in fault trees [6,13] and "defences" in attack trees [18], but function in the same manner. Following from the DT semantics [29], we generalise the structure function to interpret the additional gate.

Definition 4 (Structure function for DPTs). *The structure function* $f_T : N \times 2^{\mathit{LEAF}_T} \to \mathbb{B}$ *of a DPT* T *and* $A \subseteq \mathit{LEAF}$ *is defined by:*

$$f_T(v, A) = \begin{cases} \mathbf{1} & \begin{cases} \text{if } t(v) = \texttt{OR} & \text{and } \exists u \in ch(v).\ f_T(u, A) = \mathbf{1} \\ \text{if } t(v) = \texttt{AND} & \text{and } \forall u \in ch(v).\ f_T(u, A) = \mathbf{1} \\ \text{if } t(v) = \texttt{INHIBIT} & \text{and } f_T(u_1, A) = \mathbf{1}, f_T(u_2, A) = \mathbf{0} \text{ where } ch(v) = [u_1, u_2] \\ \text{if } t(v) = \texttt{LEAF} & \text{and } v \in A \end{cases} \\ \mathbf{0} & \mathit{otherwise} \end{cases}$$

The interpretation for the overall tree is $f_T(A) = f_T(R_T, A)$*.*

We will sometimes use syntactic notation to construct DPTs using `AND` (written as $\cap$), `OR` ($\cup$) and `INHIBIT` ($\bigcirc$) as term constructors and label names to stand for nodes in the graph. This can be made formal by defining interpretation functions on terms (for example, see [17]).

3.4 Consequence Assessments

A consequence assessment constitutes the right hand side of a safety-security bowtie and investigates potential ways to mitigate the central risk event or negative outcomes. As usually done with bowties, we make use of event trees to investigate each possible outcome of the central risk event. For the RHS of the bowtie, the leaf nodes in the DCT are potential final consequence events and each `CHOOSE` node represents a splitting point where a sub-event can have different outcomes (typically for a binary choice point, something happens or does not happen). This is captured by a *consequence function* C which selects the outcome event at each `CHOOSE` node, tracing a path through to a final outcome. For the Boolean interpretation only a single choice can happen; for more general interpretations we might assign a possibility (Y/N/M) against each branch, probabilities or probability distributions, etc.

Definition 5 (Disruption Consequence Tree). *A Disruption Consequence Tree (DCT) is a structure tree with gate types* $\mathbb{T} = \{\, \texttt{LEAF}, \texttt{CHOOSE} \,\}$*.*

Definition 6 (Structure function for DCTs). *The structure function* $f_T : N \times (\texttt{CHOOSE}_T \to \mathbb{N}) \to N$ *of a DCT* T *is defined by:*

$$f_T(v, C) = \begin{cases} v & t(v) = \texttt{LEAF} \\ f_T(u_n, C) & \text{where } C(v) = n \text{ and } ch(v) = [u_1, \ldots, u_n] \end{cases}$$

where C *is a consequence function which selects an outcome for each* CHOOSE *node, i.e.,* $1 \leq C(v) \leq length(ch(v))$ *for* $v \in \texttt{CHOOSE}_T$*. The interpretation for the overall tree is* $f_T(C) = f_T(R_T, C)$*.*

3.5 Bowties

A bowtie is just a pair of two structured trees. Intuitively, the top event corresponds to the two roots linked together, a realised disruption.

Definition 7 (Disruption Bowtie). *A Disruption Bowtie (DB) is a pair* $\langle T_P, T_c \rangle$ *of a disruption prevention tree and a disruption consequence tree.*

4 Interdependence Analysis

When assessing the interactions between security and safety we categorize them following the existing literature [9, 20], they are:

1. *conditional dependency*, meaning the safety of the system depends on its security, and conversely,
2. *mutual reinforcement*, fulfilment of safety requirements or safety measures contributes to security, and conversely,
3. *antagonistic*, when considered jointly, safety and security requirements or measures lead to conflicting situations, and
4. *independence*, when the interactions are mutually exclusive or not in interference.

The interactions, obviously, have a great difference on the result of the assessment and in our case study we see dramatic differences in the same exact events inter-playing in different ways. We showcase in the scenarios the need for specific operations and describe their semantics in this section. We omit the right hand side of the bowtie for those interdependencies which do not strictly require it (*independence* and *conditional*). Whilst we find that both *reinforcing* and *antagonistic* require the recovery side of the bowtie. We note that each interdependency condition leads to trees joined in the usual fashion when combining to form a new tree.

4.1 Running Example: Safety Security Interplay

As main point of reference we use the reported chain of events by Facebook engineering [15, 16] as well as the external assessment conducted by CloudFlare [21]. We conducted a structured risk assessment of the Facebook DNS shutdown using security safety bowties focusing our analysis on how safety and security measures interacted to lead to the specific events. Our methodology of analysis was the following: i) we downloaded the analysis of the events released by the Facebook engineering team as well as an external examination by cloud hosting provider CloudFlare which witnessed the events externally. ii) we systematically labelled each brief to extract events, its causes and outcomes and the recovery/preventions that were in place (and failed in this case). For the sake of case study we choose to adapt an attack tree that reached the same risk event of the server outage, for this we borrow an existing example from [17]. For the sake of diagrammatic explainability edges are labelled by an identity function, which creates the intermediate events, a full table containing all edge labels is presented in Tab. 2.

The triggering of the initial Facebook blackout was caused by two independent events, firstly an engineer sent an update to the backbone routers which was faulty, this was not caught by the safety checker and was allowed to go through. An internal safety procedure was: that if the internal DNS resolver couldn't find its own address it removed the DNS from the internet, meaning they were invisible to the outside world. In the case of an attack we consider a generic server, offering ftp, ssh, and rsh services, as seen in previous

Table 2. Explanations of edge and event labels.

Attack Label	Explanation	Safety Label	Explanation
Server Down	Once the attacker has gained access to the server he chooses to deny others access.	**Server Down**	Due to erros with DNS configuration and a faulty update with a failed check the server goes offline
auth	Breaching the authentication mechanism	**faulty update**	Due to unchecked issues in patch the faulty update when't through
no auth	Bypassing the authentication mechanism	**DNS OFF**	Due to safety procedure DNS hidden from internet
user privileges	Gain illegitimate privileges	**update check**	The safety checker for updates failed to spot the fault
buffer overflow	Exploit vulnerabilities to overflow	**dns check**	Internal safety procedure was that if the internal DNS resolver couldn't find its own address it removed the DNS from the internet
rsa	RSAREF2 library buffer overflow vulnerability	**server patch**	An engineer send an update to the backbone routers which was faulty
ssh	ssh deamon overflow vulnerability	**resolve dns**	DNS is unable to resolve its own addresss
rsh	Remotely executing shell commands	**remote login**	Remotely patch the server to remove failure
ftp	Exploit ftp vulnerability to upload files		
server disrupt	Once the attacker has gained access to the server he alters the DNS settings		
disable ssh	Remove ssh access to stop vulnerability		

work [17]. The attack tree in Fig. 1 shows how an attacker can disrupt a server in two ways: either without providing any user credentials (no-auth) or by breaching the authentication mechanism (auth). In the first case, the attacker must first gain user privileges and then perform a local buffer overflow attack. To gain user privileges, the attacker must exploit an FTP vulnerability to anonymously upload a list of trusted hosts as well as remotely executing shell commands using RSH. The second way is to abuse a buffer overflow in both the ssh daemon (SSH) and the RSAREF2 library (RSA) used for authentication.

Using the described notation we formalise the security and safety scenarios as DPT_S and DPT_A for safety and security respectively:

$$DPT_S = ((\texttt{server patch} \hexagon \texttt{update check}) \cap (\texttt{resolve DNS} \hexagon \texttt{dns check}))$$

$$DPT_A = ((\texttt{ftp} \cap \texttt{rsh}) \cap \texttt{buffer overflow}) \cup (\texttt{rsa} \cap \texttt{ssh})$$

4.2 Independence

The Independence relationship can simply be defined as an `OR` gate, whereby either the security scenario *or* the safety scenario lead to the eventual risk condition, an initial investigation of this has has been done before [28], and can be visualised in Fig. 3 for our case study.

For quantitative analysis there are, however, certain complications. Since now both a safety assessment is present as well as a security assessment, there needs to be a choice as to how the OR gate quantification is performed. Whilst this topic in itself warrants a deeper discussion orthogonal to the contributions of this work, a naive approach is to employ the approach from Steiner and Liggesmeyer and calculate a composition of risk-probability pairs leading to the `OR` gate outcome [28]. In the case study we simple see that the server outage may be caused by an active attacker (attack tree), or a failure of the update process and DNS misconfiguration (safety tree), or both.

Independence is modelled as two DPTs joining roots with an `OR` gate.

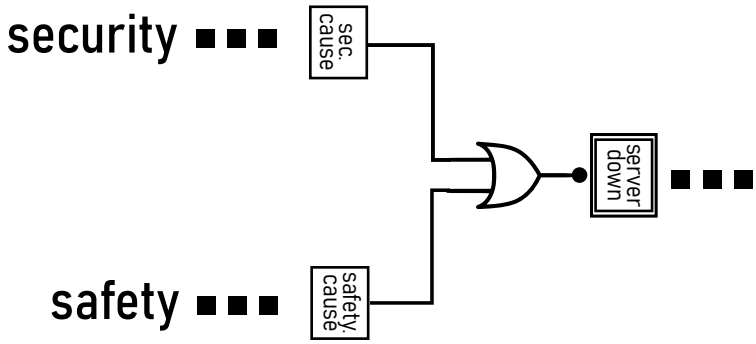

Fig. 3. Independence - when the two events do not interact to cause the final outcome. This is a sub-tree of the tree in Fig 1, and as such details are omitted

Definition 8. *An independent joining of a security and a safety assessment (assuming both are modelled as disruption defence trees) is defined as: given two disruption defence trees DPT_A and DPT_F a new tree DPT_{AF} is formed as:*

– $DPT_{AF} := \langle N_A \uplus N_E \uplus \{R_{T_{AF}}\},\ \mathbb{T},\ t_a \uplus t_f \uplus \{R_{T_{AF}} \mapsto \mathtt{OR}\},\ ch_A \uplus ch_F \uplus \{R_{T_{AF}} \mapsto [R_{T_A}, R_{T_F}]\}\rangle$

So for the example in Fig. 3, the independent joining would look like this:

$$FB_{IND} = (((\mathtt{ftp} \cap \mathtt{rsa}) \cap \mathtt{buffer\ overflow}) \cup (\mathtt{rsa} \cap \mathtt{ssh})) \cup ((\mathtt{server\ patch} \bigcirc \mathtt{update\ check}) \cap (\mathtt{resolve\ DNS} \bigcirc \mathtt{dns\ check}))$$

4.3 Conditional Dependency

Given two disruption prevention trees a conditional dependence means that the vertex of one of the trees corresponds to a child node or intermediate event of the other. A conditional joining extends the tree to include all the children of the vertex being merged in, with the old vertex simply becoming an intermediate event to the other tree. The assessment now needs to include all the new nodes that lead to the risk event. In the example we can easily presume that the failure to resolve the DNS could be due to an attacker having gained access to the server (following the attack path from the attack tree) and changing the configuration. So by merging the attack tree to the fault tree on the resolve DNS node we get a conditional dependence, as seen in Fig. 4.

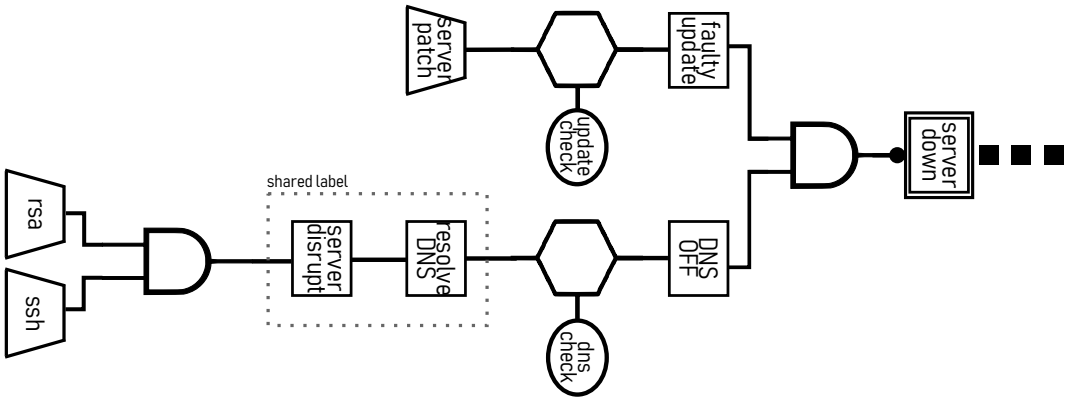

Fig. 4. Conditional - when the safety outcome is conditional on the security or vice versa. Scenario modelled by security outcomes (simplified) causing fault.

A conditional dependence may be viewed as the expansion of a previously unknown external event. Whilst previously we had a `LEAF` node that represented an input to an assessment, we now expand this node into a fully fledged disruption tree. This notion of external event is seen in previous literature [30], and this can be seen as an enhancement of knowledge of what lead to the input events. Semantically it means that a node at the bottom of a disruption tree simply gains as children all the nodes in another disruption tree, and the previous child leaf node is substituted.

Definition 9. *A conditional joining of a security and a safety assessment (assuming both are modelled as a disruption defence tree) is defined as: given a joining of two disruption defence trees DPT_A to DPT_F and a condition node n_c (of type* `INHIBIT`*) a new tree DPT_{AF} is formed as:*

- $DPT_{AF} := \langle N_A \uplus N_F \uplus \{n_c \mapsto R_{T_A}\}, \mathbb{T}, t_A \uplus t_F \uplus \{t_{AF}(n_c) := t_A(R_{T_A})\}, ch_A \uplus ch_F \rangle$

So for the example in Fig. 4, the conditional joining would look like this:

$$FB_{COND} = ((\texttt{server patch} \bigcirc \texttt{update check}) \cap ((\texttt{rsa} \cap \texttt{ssh}) \bigcirc \texttt{dns check}))$$

4.4 Reinforcing Dependency

The reinforcement dependency takes place when a DCT from one bowtie contains a response/outcome that leads to the resolution of the initial scenario of a separate DPT. This means that due to the response to an incident another incident is no longer plausible. If we take the same security response as previously, disabling ssh to stop the buffer overflow, we see that the patching of the server event which led to the faulty update cannot take place anymore, as per Fig. 5. This once again requires the right hand side of a bowtie. In this scenario the response outcome of one bowtie is chaining to the cause event of another tree. Concretely, the response `disable ssh` negates the `server patch` event of the fault tree. This is modelled through a `INHIBIT` gate. Simply this is evaluated as the action no longer being possible (as the response is not negated it results to 0 in `INHIBIT` semantics).

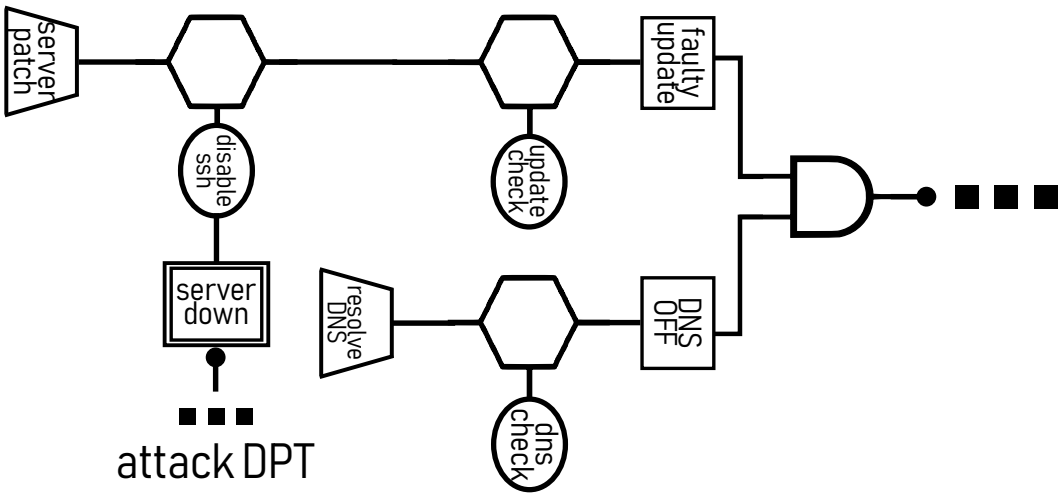

Fig. 5. Reinforcing - when a security or safety measure helps the other incident. This is modelled by the outcome of security (simplified) in this case, stopping the safety incident. Attack tree omitted as it is sub-tree of Fig. 1

Reinforcing dependence assumes that as an outcome of fixing a fault or mitigating an attack, another fault or attack is no longer possible. The approach taken in this scenario is to take the DCT on the right hand side of the bowtie, joining into the reinforced DPT, extending it, and creating a new DPT. This leads to the final risk event of the reinforced DPT being the new risk event of focus for the whole assessment. We make the assumption that if an event does not lead to any other disruptions in the DPT being reinforced it is not

influencing the disruption and can be pruned. (to preserver the tree structure and avoid back propagating branches). We infer that if an event is present in both trees there is an implied relationship, extensions could instead assign further labels to construct these relationships.

Definition 10. *A reinforcing joining of a disruption consequence assessment (implicitly joining any left most side DPT if a bowtie) DCT and any disruption defence tree DPT takes the form of a joining of the single reinforcing branch $V_C = f_T(R_T \in DCT, C)$ into a node n_r of type* `INHIBIT`*:*

– $DPT_{AF} := \langle N \uplus V_C, \mathbb{T}, t_P \uplus t_C, ch_P \uplus ch_C \uplus \{ch_{PC} : n_r \mapsto [u1, \overline{V_C}]\}$

We note that an inherent relationship between prevention and disruptions is assumed. This is not automatically inferred through formalisations. In this case a prevention, child two of the `INHIBIT` gate, is replaced by the branch of the DPT that reinforces the first child. We see that the previous child two $u2$ of the `INHIBIT` node is replaced by the branch V_C. One can observe that under this formalism some knowledge about the previous events are lost, as they are pruned in the joining. For clarity, an example joining between DCT_i and DPT_j:

$$DPT_i = (\texttt{X} \bigcirc \texttt{Y}) \text{ and } DCT_j = ([j_i \cup \overline{j_i}, j_n \cup \overline{j_n}])$$

if j_n is reinforcing to `X`, their reinforcing joining DPT_{ij}, joins the path $V_j = \{j_i, j_n\}$ to DPT_i

$$DPT_{ij} = (\texttt{X} \bigcirc \overline{V_j})$$

4.5 Antagonistic Dependency

When security and/or safety measures directly antagonise each other it means that a measure that attempts to make the system more secure has an adverse effect on its safety and conversely. We see that this antagonism is almost exclusively present in an outcome of an incident taking place (i.e. after the warning has gone off), and not visible in the bare DPT. For this specific scenario we need the use of the right hand side of a bowtie and event trees to represent the dependency. Going back to our case study, once the server has been taken down, engineers from the safety domain and the security domain investigate the incident. The first team (safety), sees that the incident is due to a faulty DNS configuration and patches the DNS remotely through SSH. The second team (security), sees that the incident is due to a vulnerability in the ssh library and therefore disables ssh (perhaps temporarily whilst the issue is being fixed by local engineers, as was the case for Facebook). As one can see, if the security response is put in place, there is no way the safety response can be put in place, this means that the fault or incident cannot be fixed if the security incident is fixed. We note this is exactly what took place during the Facebook outage, a security response disallowed the safety engineers to remote in and patch the fault, leading to physical access being required. In practice this means that the system can either be made safe or it can be made secure, but not both. This unlike the previous relationships is modelled through the consequence assessment and through the usage of event trees.

Definition 11. *An antagonistic joining of a security and a safety consequence is defined as: given a joining of two disruption consequence assessments DCT_A and DCT_F, and antagonistic event e, a new tree DCT_{AF} is formed as:*

– $DCT_{AF} := \langle N_a \uplus N_s, \mathbb{T}, t_a \uplus t_s, ch_s \uplus ch_a \uplus \{ch_{sa} : e \mapsto \texttt{CHOOSE}\} \uplus \{ch_{sa} : \overline{e} \mapsto \texttt{CHOOSE}\}\rangle$,

finally the Facebook DNS Disruption as a Bowtie can be defined in the following steps, given the DPT_S and DPT_A from Sec. 4.1 they can be *independently* joined following Def. 8 to form:

$$\begin{aligned} FB_{DPT} =& (((\texttt{ftp} \cap \texttt{rsh}) \cap \texttt{buffer overflow}) \cup (\texttt{rsa} \cap \texttt{ssh})) \cup \\ & ((\texttt{server patch} \bigcirc \texttt{update check}) \cap (\texttt{resolve DNS} \bigcirc \texttt{dns check})) \end{aligned}$$

and then given the two possible consequence trees, $[DCT_A, DCT_S]$:

$$DCT_S = (\texttt{remote login}) \text{ and } DCT_A = (\texttt{disable ssh})$$

we can antagonistically join them following Def. 11 to form the FB case study situation of conflict:

$$\begin{aligned} FB_{DCT} =& (\texttt{remote login} \cap \overline{\texttt{disable ssh}}) \cup (\overline{\texttt{remote login}} \cap \texttt{disable ssh}) \\ & \text{or more simply } (\texttt{remote login} \oplus \texttt{disable ssh}). \end{aligned}$$

Putting the two together you can obtain the bowtie $FB_{Bowtie} = [FB_{DPT} \rightarrow [[\texttt{server outage}]] \rightarrow FB_{DCT}]$

5 Conclusions

This work presented the first formalisation of safety security interdependence covering the full spectrum of dependencies through the use of bowtie diagrams. We notice some interesting outcomes: 1) for some of the dependencies dynamic gates are necessary i.e. `INHIBIT` gates (not typically used). 2) some dependencies make more sense on the outcome of the tree, are beyond the attack/fault tree, and need to be reflected in the recovery phase (showcasing the suitability of bowties as a way to explore interdependence). We also provide the first formalisation of DPTs alongside its semantics. Finally, this work is the first to provide formalisations of the safety security interdependence operations under a unified assessment.

5.1 Related work

This work is by no means the first exploration of combining security and safety [1, 12, 14, 19, 20]. Existing work in the area also highlights that this is by no means an easy feat as there are several differences as an outcome of developing in completely different contexts [12]. Gould and Bieder [12] explore the initial development of the two fields and the contexts in which they coexist. Safety is originally developed as means to avoid systems failures guided by the increased understanding of hazards and inevitability of accidents. Conversely, security developed initially as an outcome of malicious parties wishing to harm systems, however has developed into a critical component of any system analysis. The authors cite the increase of emphasis on the reduction of risks associated with mitigating security threats as a key factor as to why security and safety should be integrated more. However security and safety fields diverge vastly on technology usages and quantification of risks. The two fields however are strongly interconnected and as such their integration is essential. The authors propose the need for new policies and shared practice as without the other neither field can be complete and neither safe nor secure [12]. This recent work highlights the need to unify the assessment techniques, although they do not provide intuition as to how to do so. Literature has already considered how these two fields may be considered under the same methodology [14], however, often this is done rather relying on expert analysis and without underlying formal foundations. However, there have also been some specific assessment techniques proposed to join the two assessments.

5.2 Assessment Techniques

An emerging field that has a deep integration between these two topics is that of industrial control systems (ICS) [20]. Industrial systems have always been focused on safety, however with the new integration of internet infrastructure and the IoT the avenues of attack greatly increase [2, 3], leading to the need for better security evaluations to be in place. Work by Kriaa et al. [20], surveys a list of techniques used in this field for both safety and security. In this work the authors discuss the possibility for a safety case to be broken by security threats and discuss means to address this in the system assessment. The paper also discusses current standards for safety and security in the context of ICS, how they may integrate and assessments to decide which category needs to be considered. Using a different formalism of BMDPs previous work has similarly used the case study of a pipeline to observe security and safety interactions [19]. Their work was one of the first to evaluate the different interdependency types. In more recent work [11] address this same topic in the context of safety and security assurance. In this work the authors conduct an hazard analysis and risk assessment (HARA) a common safety assurance methodology alongside a threat analysis and risk assessment (TARA), which can be used to assess security risks. Combining these two techniques they are able to conduct a unified safety security assessment for their case study of an autonomous vehicle adhering to the safety standard ISO26262 and security standard ISO21434 for autonomous vehicles. Although we note that the unification of these techniques is not done formally under a single construct but more as an iterative manual process.

Having established the need for rigourous security assessment recent work experimented with the usage of bowties for this purpose [4]. In their work the authors apply the usage of bowties to security analysis in the maritime industry. They showcase the flexibility of the approach in quantified various threats and calculating the impact of the attacks. The paper proposes a formula for quantifying threat and consequence impact. The authors also ran a user study showed that people in the maritime industry thought Bowties for security were a very useful tool, showcasing the applicability of bowties to industry case studies. The

first approach combining safety and security bowties [1], combines an attack tree with a safety bowtie, and whilst this proves to be a great first step, falls short of completely uniting the various interdependencies into a single assessment.

The literature showcases that there is already a wide use of these techniques across a wide range of fields and a formal representation allows to set the foundations for further expansions and reasoning. Formal descriptions allow to have a more rigorous way to analyse systems, they allow to establish what you can and can't do using the approach as well as reason about properties of a formalism. Previous work has applied this in the context of safety cases [7] and showcased it as a valuable way to create and assess system evaluations. Even more so, security researchers have applied formal representations to attack trees [17, 22], showcasing its usefulness to reason about systems and performing structured analysis [10]. In more recent work [8] the authors formalise bowties in and causality sequences for usage of safety and avionics, their formalisation is similar to the one used in this work, although they omit the analysis of interdependency. As such they do not discuss joining operations and do not assess security and safety under the same construct, which lies at the foundation of our security and safety assessments. Our work begins to apply the same formal rigour to safety and security assessments using bowties to unify the assessment in a structured and formal manner, both drawing from clear semantic understanding as well as a clear visual representation.

5.3 Next steps

In order to fully generalise this approach a larger case study should be adapted, in particular further analysis on how the two types of trees, event and disruption, can be chained to combine the analysis of several consequent risk events is desirable. We also see as desirable the investigation of further gate types to fully encapsulate the the full spectrum of analysis currently seen across the literature. The formal definition of joining operations we have provided for security and safety assessments presents a unique potential for automating the process of combining safety security analysis, future work into this area would have tremendous impact to more rigorously examine scenarios such as the Facebook DNS incident from both assessment angles. However there is also a need for further investigation into properties of these provided join operations. We also see the usefulness of hierarchicalisation as means to combine sequences of events into a single more readable structure. This has been explored in the past successfully for assurance cases [7] and as in this approach several assessments are joined together under the same formalism, an enhancement of this approach could be even more effective and desirable. We note that further to the formalisation of the structures and their semantics some distinctions exist in how the probabilities of risk are calculated, however this is orthogonal to the work presented here and we reserve this investigation to future work.

Acknowledgements. We're grateful to Ewen Denney for suggesting to us to investigate bowtie diagrams for safety-security assessments, as well as comments on an early draft. This work was funded by the AISEC grant under EPSRC number EP/T027037/1

References

1. Abdo, H., Kaouk, M., Flaus, J.M., Masse, F.: A safety/security risk analysis approach of industrial control systems: A cyber bowtie–combining new version of attack tree with bowtie analysis. Computers & security **72**, 175–195 (2018)
2. Arnaboldi, L., Czekster, R.M., Morisset, C., Metere, R.: Modelling load-changing attacks in cyber-physical systems. Electronic Notes in Theoretical Computer Science **353**, 39–60 (2020)
3. Arnaboldi, L., Morisset, C.: Quantitative analysis of dos attacks and client puzzles in iot systems. In: International Workshop on Security and Trust Management. pp. 224–233. Springer (2017)
4. Bernsmed, K., Frøystad, C., Meland, P.H., Nesheim, D.A., Rødseth, Ø.J.: Visualizing cyber security risks with bow-tie diagrams. In: International Workshop on Graphical Models for Security. pp. 38–56. Springer (2017)
5. Budde, C.E., Kolb, C., Stoelinga, M.: Attack trees vs. fault trees: two sides of the same coin from different currencies. In: International Conference on Quantitative Evaluation of Systems. pp. 457–467. Springer (2021)
6. De Dianous, V., Fievez, C.: Aramis project: A more explicit demonstration of risk control through the use of bow–tie diagrams and the evaluation of safety barrier performance. Journal of Hazardous Materials **130**(3), 220–233 (2006)

7. Denney, E., Pai, G., Whiteside, I.: Formal foundations for hierarchical safety cases. In: 2015 IEEE 16th International Symposium on High Assurance Systems Engineering. pp. 52–59. IEEE (2015)
8. Denney, E., Pai, G., Whiteside, I.: The role of safety architectures in aviation safety cases. Reliability Engineering & System Safety **191**, 106502 (2019)
9. Eames, D.P., Moffett, J.: The integration of safety and security requirements. In: International Conference on Computer Safety, Reliability, and Security. pp. 468–480. Springer (1999)
10. Fila, B., Wideł, W.: Exploiting attack–defense trees to find an optimal set of countermeasures. In: 2020 IEEE 33rd Computer Security Foundations Symposium (CSF). pp. 395–410. IEEE (2020)
11. Gallina, B., Montecchi, L., de Oliveira, A.L., Bressan, L.P.: Multiconcern dependability-centered assurance via qualitative and quantitative coanalysis. IEEE Software (2022)
12. Gould, K.P., Bieder, C.: Safety and security: The challenges of bringing them together. In: The Coupling of Safety and Security, pp. 1–8. Springer, Cham (2020)
13. Guldenmund, F., Hale, A., Goossens, L., Betten, J., Duijm, N.J.: The development of an audit technique to assess the quality of safety barrier management. Journal of hazardous materials **130**(3), 234–241 (2006)
14. Haider, Z., Gallina, B., Carlsson, A., Mazzini, S., Puri, S.: Concertofla-based multi-concern assurance for space systems. ADA USER **40**(1), 35 (2019)
15. Janardhan, S.: Update about the october 4th outage (Oct 2021), `https://engineering.fb.com/2021/10/04/networking-traffic/outage/`
16. Janardhan, S., Janardhan, S.: More details about the october 4 outage (Oct 2021), `https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/`
17. Jhawar, R., Kordy, B., Mauw, S., Radomirović, S., Trujillo-Rasua, R.: Attack trees with sequential conjunction. In: IFIP International Information Security and Privacy Conference. pp. 339–353. Springer (2015)
18. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of attack–defense trees. In: International Workshop on Formal Aspects in Security and Trust. pp. 80–95. Springer (2010)
19. Kriaa, S., Bouissou, M., Colin, F., Halgand, Y., Pietre-Cambacedes, L.: Safety and security interactions modeling using the bdmp formalism: case study of a pipeline. In: International Conference on Computer Safety, Reliability, and Security. pp. 326–341. Springer (2014)
20. Kriaa, S., Pietre-Cambacedes, L., Bouissou, M., Halgand, Y.: A survey of approaches combining safety and security for industrial control systems. Reliability engineering & system safety **139**, 156–178 (2015)
21. Martinho, C.: Understanding how Facebook disappeared from the Internet (Oct 2021), `https://blog.cloudflare.com/october-2021-facebook-outage/`
22. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: International Conference on Information Security and Cryptology. pp. 186–198. Springer (2005)
23. Nielsen, D.S.: The cause/consequence diagram method as a basis for quantitative accident analysis. Risø National Laboratory (1971)
24. Ren, H., Chen, X., Chen, Y.: Fault tree analysis for composite structural damage. In: Reliability Based Aircraft Maintenance Optimization Applications, pp. 115–131. Academic (2017)
25. de Ruijter, A., Guldenmund, F.: The bowtie method: A review. Safety science **88**, 211–218 (2016)
26. Schneier, B.: Attack trees. Dr. Dobb's journal **24**(12), 21–29 (1999)
27. Shahriar, A., Sadiq, R., Tesfamariam, S.: Risk analysis for oil & gas pipelines: A sustainability assessment approach using fuzzy based bow-tie analysis. Journal of loss prevention in the process Industries **25**(3), 505–523 (2012)
28. Steiner, M., Liggesmeyer, P.: Combination of safety and security analysis-finding security problems that threaten the safety of a system. In: DECS : ERCIM/EWICS Workshop on Dependable Embedded and Cyber-physical Systems (2013)
29. Stoelinga, M., Kolb, C., Nicoletti, S.M., Budde, C.E., Hahn, E.M.: The marriage between safety and cybersecurity: Still practicing. In: International Symposium on Model Checking Software. pp. 3–21. Springer (2021)
30. Vesely, W.E., Goldberg, F.F., Roberts, N.H., Haasl, D.F.: Fault tree handbook. Tech. Rep. NUREG-0492, Nuclear Regulatory Commission Washington DC (1981)