IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 March 2024, accepted 16 May 2024, date of publication 22 May 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3404264

APPLIED RESEARCH

Automated Knowledge-Based Cybersecurity Risk Assessment of Cyber-Physical Systems

STEPHEN C. PHILLIPS, STEVE TAYLOR, MICHAEL BONIFACE, STEFANO MODAFFERI, AND MIKE SURRIDGE

IT Innovation Centre, University of Southampton, SO17 1BJ Southampton, U.K.

Corresponding author: Stephen C. Phillips (S.C.Phillips@southampton.ac.uk)

This work was supported in part by European Union's Horizon Europe Research and Innovation Program through the Synthetic Generation of Hematological Data Over Federated Computing Frameworks (SYNTHEMA) Project under Grant 101095530, and in part by the U.K. Research and Innovation (UKRI) and UKRI Trustworthy Autonomous Systems Hub under Grant EP/V00784X/1.

ABSTRACT This paper describes a simulation-based approach for automated risk assessment of complex cyber-physical systems to support implementers of ISO 27005. The approach is based on systematic cause-and-effect modelling of threats, their causes and effects, and the ways in which the effects of one threat can lead to other threats. In this way, the approach deals with inter-dependencies within the target system, automatically finding attack paths and secondary effect cascades, which generally are very complex and the source of many challenges when implementing ISO 27005. The approach uses a knowledgebase describing classes of system assets and their possible relationships, along with the associated threats, causes and effects in a generic context. A target system can then be modelled in terms of related assets, describing the intended system structure and purpose (in the absence of any deviations). The knowledgebase is then used to identify which threats are relevant and create a cause-and-effect simulation of those threats. This allows threat likelihoods and risk levels to be found based on input concerning trust assumptions and the presence of controls in the system. The approach has been implemented by the open source Spyderisk project and validated by modelling a published case study of an attack on a steel mill. Given reasonable assumptions about security controls in place, the shortest, highest likelihood attack path found coincides with the published analysis. The case study demonstrates the strengths of the approach: transparency, reproducibility, and performance.

INDEX TERMS Computer security, cyber-physical systems, information security, risk analysis, systems modeling, threat assessment.

I. INTRODUCTION

Increasing cyber-physical convergence and inter-connection between formally disparate systems, including critical infrastructures, is increasing the range of threats and potential consequential harms faced by designers, operators and those in society who depend on their safe and reliable operation. Risk assessment is the widely accepted way of mitigating threats to systems. Making decisions based on risk is a good approach and can be applied throughout a system's lifecycle to ensure resources are efficiently deployed to address the highest risk consequences first. For example, if a new security vulnerability is registered in the Common Vulnerabilities and Exposures (CVE) database [1] it is crucial to understand the impact. How likely is it that an attacker will be able to exploit the vulnerability given the context of the software process in the wider system? What would be the impact be if they did? The complex inter-asset dependencies in today's networked information systems makes such analysis difficult to achieve, and information security only offers a partial view of the overall risk within a complex cyber-physical system. Other risk factors and harms related to privacy, trustworthiness, safety, and regulatory compliance must also be considered. The increasingly complex relationships that exist between assets within a system, threats, vulnerabilities, and harms means it is now not practical to document everything

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

about a system using the traditional approach of human experts, spreadsheets and taxonomies of threat catalogues. The efficiency of risk analysis and consideration of coverage of the wide range of attack paths becomes a major concern. As such, there is now a need to model and curate systems knowledge in a machine-readable format, and reuse of that knowledge in computational simulation and inference processes that can model complexity in an automated, repeatable and reliable way.

In this paper we describe a modelling approach for automated risk assessment of complex cyber-physical systems as implemented by the open source Spyderisk project [2]. We describe a practical application in information security demonstrating how the approach supports system designers and operators in risk assessment processes defined in ISO 27005 [3] and Information Security Management System defined in ISO 27001 [4]. The approach aims to foster trust and security in complex systems by supporting both security by-design and risk assessment during operations in ways that explain how harm can arise before it happens.

We present a new ontology for describing models of cyber-physical systems to be analysed along with the related threats, consequences, and controls. The ontology is designed to support systematic cause-effect simulation using semantic reasoning that can find the threats to the system, their consequences and associated risk levels. The key advantages of the approach are that the risk assessment is consistent and complete (within the limits of the model); that cause and effect is followed through the system so that chains of attack steps and secondary effects are automatically considered; and that the risk levels are automatically calculated based on a process requiring minimal input.

The scope of any risk assessment is limited by what is known by those undertaking the analysis. This can be knowledge provided by experts or knowledge encoded into taxonomies or more complex data structures such as ontologies. Our knowledgebase has been developed for over 10 years of case studies and published evidence. This includes threats to information systems of natural or human origin, and which could be accidental or deliberate, along with mechanisms for inferring complex features (such as network paths and data flows in the case of information systems). In the paper, we also describe a software implementation of our approach known as Spyderisk. With a minimal amount of input from the user, Spyderisk can compute the threats to a system, ordered by risk level, where the risk level combines the business impact of a consequence as well as the computed likelihood.

The structure of the paper is as follows. In Section II we discuss related work and in Section III we describe the basic principles of the approach. We then detail the core ontology in Section IV, and processes of risk assessment in Section V. Section VI then provides validation describing a "network" knowledgebase applied to a documented attack on a German steel mill, along with how the approach and knowledgebase aligns with ISO 27005 process and terms. Finally, we offer conclusions and future work in Section VII.

II. RELATED WORK

Modelling and analysing threats and risk, particularly in information systems, has a rich history and a scattered development that has brought to the definition of several standards and many different approaches and terminology to address risk assessment. At European level the European Union Agency for Cybersecurity (ENISA) provides an up-to-date information and policy on Cybersecurity and risk assessment methods and tools [5]. A level of customisation is commonly needed in different domains, and historically risk management methods have been developed for a specific domain (e.g., manufacturing, health).

A. INFORMATION SECURITY RISK ASSESSMENT METHODS

Risk assessment methods can cover the entire lifecycle but more often focus on specific phases of the cycle. There is a wide variety of required inputs, and different types (and quality and detail level) of outputs, with some methods better supported by tools than others. The required knowledge and the involvement of all the stakeholders vary across the methods and they differ in completeness (a detailed comparison of 11 methods is presented in [6]).

General approaches such as ISO 27005, NIST SP 800-30 [7], and OCTAVE [8] have a very strong theoretical power, but their implementation often leaves the risk analyst to manually consider the possible threats, consequences and risks, which imposes too much effort on the analyst. Mnemonics such as CIA and methodologies such as STRIDE [9] and LINDDUN [10] help the analyst consider a variety of information security or privacy threats, but alone they cannot assess risk. Other quantitative approaches like FAIR [11] and CORAS [12] require extensive knowledge and effort to identify all the inputs to the system, while FRAAP [13] is more qualitative and prioritises a fast result, becoming suitable only for relatively small projects.

ISO 27005 defines a standard way of managing information security risks, refers to ISO 27000 [14] for underlying nomenclature, and supports Information Security Management System defined in ISO 27001. In the asset-based approach of ISO 27005 one identifies primary assets (e.g., key data and business processes) and supporting assets (e.g., computers and networks). One must then identify threats affecting any of those assets, estimate their likelihood and impact (taking security controls into account), and from this estimation, determine the risk level for each threat. If the risk levels are too high, one should then add more security controls to reduce the residual risk to an acceptable level.

NIST provide a broad Cybersecurity Framework [15] which includes the NIST SP 800-30 Guide for Conducting Risk Assessments [7] of federal information systems

and organisations. The guide permits a wide range of analysis approaches (threat-oriented, asset/impact-oriented or vulnerability-oriented) and analysis techniques (e.g., graph-based) and is in this sense less prescriptive than ISO 27005.

Of these methods, ISO 27005 is the only international standard, is considered the most complete [6], and its strongly asset-based approach is also well-suited to a computerised process. Spyderisk therefore follows the ISO 27005 method for risk analysis but models more than just information security risks. Spyderisk shares a common language with the security compliance sector, while allowing potential harms and controls to be explored beyond those envisaged by the standards.

B. ONTOLOGIES

Any risk assessment method requires the definition of the meaning of terms and some use ontologies for this. Several information security risk assessment domain ontologies [16], [17], [18], [19] have been created and are useful to provide a common language for such risk assessment processes. A comparative study of ISO 27000 series ontologies is presented in [20].

While other ontologies often take a theoretical approach to formalise terms used in the field, we have defined a minimum set of generic risk assessment concepts for practically supporting automation (see Section IV). Furthermore, the Spyderisk knowledgebase which describes specific asset types, relations, threats and controls for information security (see Section VI-B) is also described using an ontology, including a type hierarchy, and describing many generic threat types and controls. The D3FEND ontology [21] similarly takes a practical approach in using an ontology to model known defensive techniques. The main feature of the Spyderisk ontology is that it is designed to support a cause-and-effect approach to risk modelling.

C. CONTEXT AND THREAT PROPAGATION

Many risk assessment methods (and their software implementations) focus on individual assets, and do not consider their context or interconnection in system and their components where vulnerabilities in one asset may propagate to others. ISO 27005 states ''dependencies between assets should be documented and risk propagation assessed'' and suggests using asset dependency graphs as a tool. However, the detail of determining the threats and their likelihood is left to the risk analyst.

Spyderisk analyses the interdependencies of assets and automatically takes account of how the consequence of one threat can increase the likelihood of another. Several other works address the threat propagation issue [22], [23], but they focus on the propagation without including some other features included in Spyderisk such as the automated identification of threats through a knowledgebase.

D. SOFTWARE SUPPORT

Software solutions to help in the analysis process do exist [24] with open source tools available [25] and spreadsheets commonly being used [26]. Graphical tools to assist with a manual analysis include CORAS [12], Microsoft Threat Modeling Tool (TMT) [27], and OWASP Threat Dragon [28]. However, leaving the user to identify all the threats and estimate risk levels inevitably leads to a subjective and incomplete risk treatment plan which can also be also time consuming to create and therefore only updated periodically (often annually). The periodic and highly manual nature of such risk assessments means that the assessment quickly gets out of synch with changes in the IT infrastructure, business processes and the external environment of known vulnerabilities. Software tools that address more automated risk assessment, similar to Spyderisk, include IriusRisk [29] and ThreatModeler [30], both proprietary and closed source. Both use threat libraries simply linked to asset types to identify threats in the risk analyst's model of a system. ThreatModeler uses process-flow diagrams whereas IriusRisk uses a variation of a data-flow diagram. IriusRisk has a complex risk calculation method and both products integrate with external systems. The methods and tools mentioned above focus primarily on individual assets, and do not sufficiently consider their context or interconnection in systems and their components where vulnerabilities in one asset may propagate to others. Consequently, there is a need for new ways to support risk identification, analysis, evaluation, and treatment for an ever-growing and complex range of interconnected risk factors, assets, threats and harms.

Spyderisk has a rich web-based interface and a client API supporting the specification and analysis of the system model. As described in the following sections, it uses some advanced modelling to automate much of the risk analysis process. It, and the associated knowledgebase, is also available free and open source and open to users and contributors.

E. AUTOMATED RISK ASSESSMENT

The ThreMA approach [31] (extended in [32]) has some similarity to Spyderisk in that it includes a formal vocabulary for modelling ICT infrastructure along with a threat catalogue and a reasoning process, but uses the Protégé ontology tool rather than a specialised client interface and scalable multi-user service. APSIA [33] considers both cyber and privacy risks but has only six inter-asset relation types, limiting the scope and specificity of the threat analysis. AMBIENT [34] incorporates data from many sources and combines a cyber-security risk assessment (using CORAS models) with a separate privacy risk assessment (using the same limited inter-asset dependency model as APSIA). The following approaches offer partial solutions for specific types of ICT systems. The AutSEC method [35] takes as input the data flow diagram (DFD) manually created in the Microsoft TMT, and in [36] an ontology is used to build a representation of the data-flow using Docker Compose files as the basis of

the analysis. Both methods then use pattern matching on the DFD to find problems, and so are limited to finding threats relating to data-flows rather than the broad scope of cyber-physical systems. Microsoft TMT is also used in [37] but with a threat catalogue limited to IoT system analysis. A form of attack graph, without risk assessment or consideration of controls, is automatically generated from a formal system model description in [38] and [39].

III. GENERAL APPROACH

We have developed our approach based on case studies and evidence. Firstly, we tackled cyber-physical risk amplification resulting from systems connectivity as exemplified by the airline industry through increased information sharing between airport service operators and air traffic control under the EUROCONTROL Airport Collaborative Decision-Making initiative [40]. The challenge with optimisation based on better information is that efficiency comes from reducing spare capacity at the cost of greater interdependency. If data used to make decisions becomes inaccurate, unavailable or (worst of all) unauthentic, the impact will be very serious. Our approach used semantic models of threats and countermeasures to derive a Bayesian network representing the threats and risks in a specific air traffic control zone to support decisions on how best to respond to disruption in physical/cyber space [41]. We then generalised the approach to model trust relationships in complex multiple stakeholder ICT systems such as 5G networks. Here trust was held to exist when one stakeholder is exposed to threats whose countermeasures depend on other stakeholders [42]. The same semantic modelling approach was applied to modelling compliance requirements. Modelling non-compliance as a threat allowed system-level compliance to be analysed via machine reasoning, including compliance with GPDR in multi-stakeholder health care scenarios [43]. Finally, we 'democratized' the use of risk analysis by providing support for standardised risk assessment methods, with a specific focus on ISO 27005. This work uses semantic knowledge models to capture expertise in cybersecurity threats and countermeasures and support the key steps in the ISO 27005 process: identifying threats, determining threat likelihood and impact, evaluating the consequent risks, and deciding on the appropriate risk response, which may include implementing security control measures to reduce risks to an acceptable level. Much of the research focuses on human perceptions of cyber security [44] and the need for tools that fit user knowledge and expectations, yet expand their situational awareness [45]. As part of this, the Bayesian approach from [41] was dropped because it requires too many causation probability parameters. Instead, a simple, parsimonious approach was adopted, using fuzzy sets mapped to a set of linguistic terms (likelihood levels), which is consistent with the ISO 27005 risk analysis procedure.

Our approach addresses the challenge of complex risk assessment of inter-connected assets by breaking the risk assessment problem down into small pieces: finding localised threats to individual assets based on their connectivity to small numbers of other assets. The likelihood of a threat is determined by attributes describing the expected behaviour of the connected assets and by the presence (or not) of control strategies. The consequence of a threat changes the expected behaviour of connected assets and thus influences the likelihood of other threats, creating a complex web of threat-consequence-threat filaments throughout the system representing multi-step attacks and secondary effect cascades.

The approach is implemented in a core ontology defining the concepts, and in software encoding the various steps of the algorithm required for threat discovery and risk level calculation. The Spyderisk concepts and analysis methodology are not tied to any domain: they are generic to risk analysis. The approach requires an expert ''Knowledgebase'' created by a domain modeller (a role responsible for encoding system knowledge using an ontology), which describes the asset types, relations and controls available to be used, and the potential threats to the assets. The risk analyst creates a ''System Model'' of the intended (correctly functioning) system for which a risk assessment is required, using the assets, relations, and controls from the Knowledgebase, and defines the business impact of any undesirable behaviours for the primary assets. They then trigger the various algorithms in the software to find the threats and compute the risk levels. The software allows the risk analyst to explore the threat graph and presents control options that can be selected to reduce the likelihood of threats, with additional required controls forming a risk treatment plan. A feature key to the realism of the approach is that the Knowledgebase encodes not only which controls reduce each threat's likelihood, but which threats arise if particular controls are present, capturing side effects or threats that exploit the controls. Thus, adding controls may reduce some risks but introduce new ones.

IV. SPYDERISK CORE ONTOLOGY

The core ontology underpinning our approach defines how System Models are described along with the classes required for evaluating *Threats*. The classes used to define our System Model are described in Fig. 1. Each of these classes and further supporting classes are described in detail in the following sections, with the broad principles summarised here.

Structural classes:

- The system structure is described in a *System Model* using *Assets* and directed *Relations*, each one linking from and to an *Asset*.

Configuration classes:

- The *Trustworthiness Attributes*, *Consequences* and *Controls* further describe and configure each *Asset* type.
- *Consequences* are (generally) things that are undesirable. A *Consequence* has a defined *impact level*, its *likelihood* is determined from the *Threat(s)* that cause it, and its *risk level* is then computed.

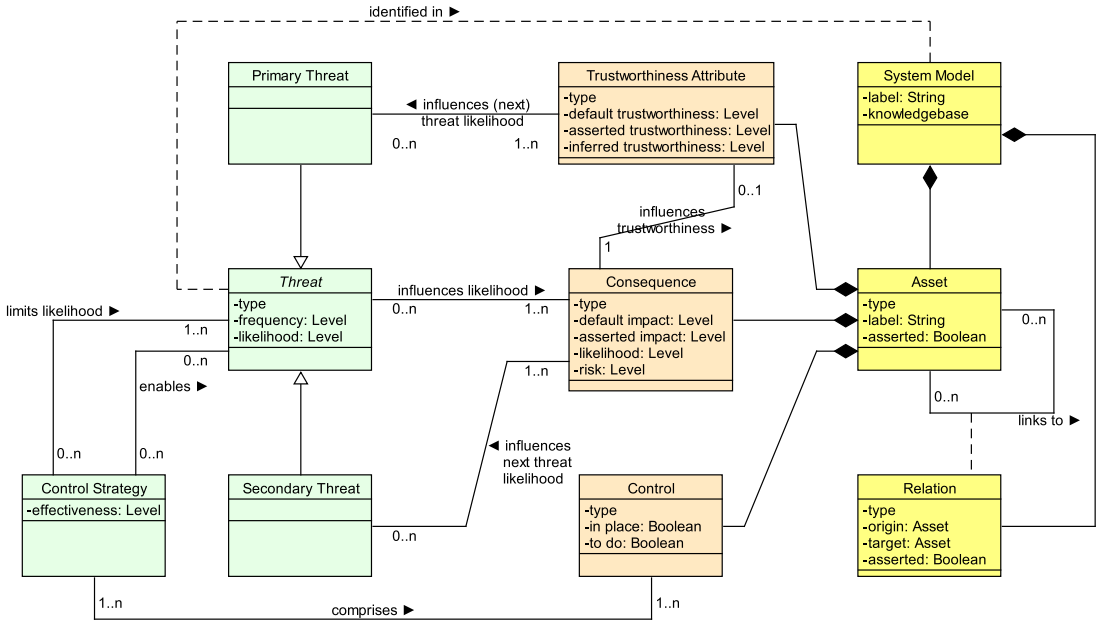

FIGURE 1. System Model classes. The yellow boxes are structural classes; brown boxes represent classes which record Asset configuration; green boxes show classes relating to threats.

- *Trustworthiness Attributes* model the expected behaviour of an *Asset*, are (generally) desirable properties and are closely related to the *Consequences*: each *Trustworthiness Attribute* is undermined by a *Consequence*.
- *Controls* are located at *Assets* and may help to reduce the *likelihood* of some *Threats* affecting the *Asset*.

Threat classes:

- A *Threat* is present in a *System Model* if a defined pattern of *Assets* and *Relations* is found. Each *Threat* has a cause and effect.
- A *Primary Threat* is a deliberate or accidental event made more *likely* by low *Trustworthiness Attribute levels* on the *Assets* it relates to.
- A *Secondary Threat* is the inevitable consequence of another *Threat* and is made more *likely* by a high *likelihood* level of one or more *Consequences.*
- *Control Strategies* combine one or more *Controls* to limit the *likelihood* of associated *Threats*. *Control Strategies* can also ''enable'' *Threats* that would otherwise be ignored, so a *Control Strategy* can both reduce the *likelihood* of one *Threat* and cause another problem by enabling a different *Threat*.

A. STRUCTURAL CLASSES

Assets and their *Relations* form the structural description of the system being modelled. Each *Asset* has zero or more *Relations* and each *Relation* links from and to an *Asset*. Some system *Assets* and their *Relations* are defined (''Asserted'') by the risk analyst, and some are added (''Inferred'') by the software. Generally, the *Assets* in a System Model are those things with value to the organisation, plus those necessary to provide sufficient information for the risk assessment.

Inferred *Assets* and *Relations* are added by an automatic analysis of the *System Model* which adds in complex aspects such as network paths and data flows but also adds in some simpler *Assets* and *Relations* that the risk analyst may have forgotten to add.

B. ASSET CONFIGURATION CLASSES

1) TRUSTWORTHINESS ATTRIBUTE

Each *Asset* type has zero or more linked *Trustworthiness Attribute* classes (with default, asserted, and inferred *trustworthiness levels*), each describing a facet of the *Asset's* behaviour and reflecting the propensity of systems or actors to fulfil expectations of correct or desirable behaviour considering the objectives of the system under evaluation. *Trustworthiness Attributes* influence the *likelihood* of related *Primary Threats*: high levels of *Trustworthiness Attributes* result in the *likelihood* of related *Primary Threats* to be low: if something is trustworthy, then you expect it not to fail.

The default *Trustworthiness Attribute* levels are determined by the domain modeller and describe the expected external environment that the *System Model* exists in. If the risk analyst feels that a default level is wrong, they can set the asserted level of any attribute which will then be used instead of the default. The attribute levels describe the effect of agents who affected an *Asset* before it entered the system (but who are not modelled in the *System Model*) and agents who are not included in the *System Model* but are still able to access an *Asset* because it is shared with other systems. The initial

attribute levels determine the entry points for attacking the system.

In addition, the inferred *Trustworthiness Attribute* levels are used to track how the effects of *Threats* propagate through the inter-connected *Assets* of the *System Model* during the risk level calculation, resulting in a consistent internal trustworthiness model.

2) CONSEQUENCE

Our definition of a *Consequence* follows ISO 27000 which defines it as the ''outcome of an event affecting objectives'' where ''objectives can relate to different disciplines'' (such as information security). The same standard defines the ''event'' affecting objectives as an ''occurrence or change of a particular set of circumstances'' which relate to our *Threat* entity (though we model the *likelihood* of the occurrence of the *Threat,* not the occurrence itself).

It is the *Consequence* that encapsulates the risk level: each *Consequence* has an impact level which is combined with its computed *likelihood* to calculate a risk level. Each *Consequence* has a default impact level which can be overridden by the risk analyst if necessary. *Threats* then take their risk level from the highest risk level *Consequence* that they ultimately cause (following back the chains of cause and effect).

A key point is that the risk analyst does not need to consider what impact levels to set on most *Assets*, as chains of cause and effect and the consequent impacts are automatically taken account of by the risk calculation - they only need to set impact levels on primary assets, those that are the most important to them. For example, if a data asset hosted in a server is required to be highly available, the modeller should indicate that the impact level for loss of availability is ''high'' but does not need to change any impact levels for the server, as the knock-on effects are automatically dealt with.

Most *Consequences* influence a paired *Trustworthiness Attribute* on the same *Asset*, with a high *likelihood Consequence* resulting in a low *Trustworthiness Attribute*.

3) CONTROL

An *Asset* can have one or more *Controls* located at it. The ISO 27000 ''control'' term means ''measure that is modifying risk''. If we interpret ''modifying risk'' to mean ''reducing risk level'', the ISO 27000 ''control'' is more akin to our *Control Strategy* which comprises one or more *Controls* (see below) as it is the *Control Strategy* which relates to a *Threat* and therefore to risk level.

The risk analyst can assert the presence of (or intention to add) *Controls* which may help reduce the *likelihood* of *Threats* to the *Asset*. They may want to assert which controls on each asset in the System Model is ''in place'' and then later, when reviewing the results of the risk calculation, may also indicate that an additional control is ''to do'', thereby letting the risk level calculation take it into account, but tracking that it represents a change in the risk treatment plan.

C. THREAT CLASSES

1) THREAT

Our definition of Threat follows ISO 27000 which defines a ''threat'' as a ''potential cause of an unwanted incident, which can result in harm to a system or organization''. We consider threats of natural and human origin, which can be accidental or deliberate. A threat can arise from within or from outside the organization.

A *Threat* has the potential to cause harm to an *Asset* in the system, which we model by the *Threat* increasing the *likelihood* of one or more *Consequences*. There are two sub-classes of *Threat*: *Primary Threat* and *Secondary Threat*. Deliberate or accidental actions are generally modelled as *Primary Threats* and they are made more likely by low level *Trustworthiness Attributes* on *Assets* involved in the *Threat*. *Secondary Threats* model the unavoidable knock-on effect of the *Consequence* of another *Threat*.

Each *Threat* has a fixed *frequency* attribute which models all the external factors contributing to the *Threat likelihood*, such as the difficulty of the attack in terms of complexity or the resources required.

2) CONTROL STRATEGY

Each *Threat* has zero or more *Control Strategy* classes which represent available options to limit the *likelihood* of a *Threat*. Each *Control Strategy* class comprises one or more *Control* classes located at *Assets* (see above) all of which must be ''in place'' or ''to do'' for the *Control Strategy* to be considered present. A *Control Strategy* has an ''effectiveness'' attribute which, if the *Control Strategy* is present, places a ceiling on the *Threat's likelihood* level.

A *Control Strategy* may also ''enable'' zero or more *Threats*: that is, be a pre-requisite. Where a Threat is ''enabled'' by such *Control Strategies*, at least one must be present for the *Threat* to have a non-negligible *likelihood*. This models the situation where a *Control Strategy* can make some things better but create side-effects or enable new types of attacks.

D. DISCUSSION

Any knowledgebase represents a trade-off between the desire for fidelity (the ability to provide precise and complete descriptions of cybersecurity issues in a system), and utility (the ability to deduce the presence of cybersecurity issues from a minimal set of starting points). For example, among the first applications of semantics to cybersecurity were ontologies developed by SBA [16], which provided comprehensive coverage of the German IT Grundschutz Manual [46].

Detailed description has never been our goal. The approach used is parsimonious compared to most other efforts to classify and describe cybersecurity concepts. Instead, the focus has been to support a machine reasoning procedure (using an automatically generated cause-and-effect simulation) that can provide as much information as possible based on a small

set of assertions about the system to be analysed. Risk analysts should not need to assert that threats are present in the system – that should be determined automatically, along with possible security measures to counteract them, and this is the purpose of our knowledgebase approach. Risk analysts still need to specify which security measures are in place or should be added and provide some input on trust assumptions about humans, software, or external influences, but only after the machine reasoning procedure has determined which inputs are needed to evaluate the risks.

Some otherwise surprising gaps in the core ontology exist for this reason, e.g., the absence of ''vulnerability'' as a first-class concept. In most comparable ontologies, threats are defined in terms of the exploitation of vulnerabilities, but this means one cannot deduce the existence of a threat unless one has asserted the presence of vulnerabilities. Many vulnerabilities are modelled in our approach as the absence of *Controls*. For example, an ''insufficient security training'' vulnerability would be modelled as the absence of the ''security training'' *Control*. Other vulnerabilities, such as technical vulnerabilities in software or other factors relating to the system's environment, are modelled by reducing specific *Trustworthiness Attributes*. The ability to exploit a vulnerability is often dependent on the attacker having the necessary access and this is modelled through our cause-and-effect chaining.

Our approach also includes rules that insert inferred *Assets* and check model consistency. These ensure that risk analysts do not need to explicitly define every detail (see Section V-B). If a cause-and-effect simulation cannot be constructed for risk analysis due to gaps, ambiguities, or inconsistencies, their presence is flagged to the risk analyst.

System boundaries are defined implicitly, being delineated by the presence of ''shared'' assets in the *System Model* that have users outside the modelled system. These assets are where trustworthiness levels can be adjusted to indicate exposure to external attackers. *System Models* may also include *Assets* belonging to different stakeholders, e.g., a retailer's online store, and the customer devices used to access it, or health care providers from different countries and a patient from one country being treated in the other. This allows detection of when data is flowing across an organisational boundary, or when data protection requirements apply to data no longer under the control of the data subject. The cause-and-effect simulation, once constructed, allows attack paths to be found that may cross organisational boundaries, making it possible to model attacks via partner organisations or customers, and understand what security measures should be recommended where these must be implemented by different organisations or by the users.

Our approach has no explicit classes of threat actors. *Threats* are expressed a potential cause of problems, independent of whether anyone has the opportunity, motive, and skills to cause them. Insider attackers can be asserted by lowering the trustworthiness levels of user roles from default levels that assume users are benevolent but not necessarily astute. The presence of external attackers can be asserted by reducing the trustworthiness levels of *Assets* directly accessible to people outside of the control of the system owner, the default levels generally being based on worst-case assumptions. One can also modify software asset trustworthiness levels to express the presence of vulnerabilities, setting the level based on the difficulty of exploitation relative to the skills of the anticipated attackers. Non-malicious sources of risk can also be captured using trustworthiness attributes, e.g., functional bugs in software, or user errors.

Trustworthiness levels (including the risk analyst's trust assumptions) and security controls are the main inputs to a risk analysis once the cause-and-effect simulator has been generated. They become the focus for ''what if'' experiments, using the simulator to find risk levels, and discover the effect on those risk levels of different trust assumptions or changes in security controls.

V. RISK ASSESSMENT

Risk assessment in our approach follows a specific process, which is repeated when system elements and factors change:

1) The *System Model* is augmented using *Construction Patterns* to infer additional *Assets* and *Relations* that enable the following step, threat discovery.
2) *Threats* are discovered via comparison of the System Model's topology with the specifications for each *Threat*.
3) *Threat likelihood* is determined, and from this, the *likelihood* of the *Consequences* resulting from them are determined.
4) Risk levels are set via the combination of the *impact* level (set by the risk analyst) and the *Consequence likelihood* level calculated previously.

We now elaborate novel features of the risk assessment approach.

A. MATCHING PATTERNS

A *Matching Pattern* describes a set of connected *Assets* to be looked for in the *System Model*: particular *Asset* types connected by specific *Relation* types. To provide the discrimination necessary both to construct inferred *Assets* and to identify *Threats*, *Matching Patterns* are necessarily quite complex. Each *Matching Pattern* must have one or more ''root'' nodes that will be matched in the System Model exactly 1 time. In addition, each pattern can contain:

- *Nodes* that match *Assets* in the *System Model* 1..n times (''mandatory'').
- *Nodes* that match *Assets* in the *System Model* 0..n times (''optional'').
- *Nodes* that match *Assets* in the *System Model* 0 times (''prohibited'').
- *Links* that must be present in the *System Model* (''mandatory'').
- *Links* that must not be present in the *System Model* (''prohibited'').

- *Links* describing that *Assets* in the *System Model* matched by two *Nodes* in the *Matching Pattern* must be distinct.

The *Construction Pattern* and *Threat* classes then use the *Matching Pattern* classes, as described below.

B. CONSTRUCTION PATTERNS

The purpose of the *Construction Patterns* is two-fold: firstly, to make it simpler for the risk analyst to construct the *System Model* by adding elements (*Assets* and/or *Relations*) that the analyst might easily forget but which can be inferred, and secondly to add elements that the risk analyst should not be expected to define but which are essential for *Threat* discovery.

Each *Construction Pattern* is linked to a *Matching Pattern* and defines the inferred *Assets* and *Relations* that will be added to the *System Model* in each location that the *Matching Pattern* is found.

Each *Construction Pattern* has a numeric ordering and a Boolean flag indicating whether it should iterate. The ordering is important, as *Construction Patterns* later in the sequence may match on the presence of inferred *Asset* types which are added by patterns executed earlier in the sequence. The ability to also repeat a *Construction Pattern* until it no longer matches (because of prohibited nodes or links in the pattern) allows complex paths through a *System Model* to be constructed.

C. THREATS

Once the *Construction Patterns* have been executed, the now expanded System Model is analysed for the presence of *Threats*. Recalling the *Threat* definition, that they are the ''potential cause of an unwanted incident, which can result in harm to a system or organization'', each *Threat* uses a *Matching Pattern* to detect parts of the System Model where *Assets* and *Relations* are such that the unwanted incident (the *Threat*) can arise.

Each *Threat* then includes a specification of how the *Nodes* in the *Matching Pattern* relate to the *Threat* causes (*Asset Trustworthiness Attributes* for *Primary Threats* and *Consequences* for *Secondary Threats*), and *Consequences*. The *Threat* model also specifies any *Control Strategies* that block the *Threat* or are necessary for the *Threat* to occur. *Control Strategies* specify *Controls* making up the strategy in terms of specific *Nodes* in the same *Matching Pattern*.

When the *Matching Pattern* matches a part of the *System Model*, a corresponding *Threat* is added to the model. This may occur in in multiple locations and thereby add many *Threats* of the same type, involving different system *Assets*.

The *Threats* then link together (via *Consequences*) to describe cause-effect chains across the System Model.

D. RISK LEVEL CALCULATION

Once *Threats* relevant to the System Model have been discovered, our approach is to use an iterative process to compute the *likelihood* of *Threats* and their *Consequences*. We then combine the *likelihood* levels and *impact* levels (defaults or specified by the risk analyst) to give *risk* levels for the *Consequences*. This is done via a simple look-up table as described in Table 1 which is consistent with ISO 27005 Appendix A.1.1.2.3. The table currently used is asymmetric, so that low likelihood but high impact risks are rated higher than high likelihood but low impact risks. This asymmetry is a domain modeller preference, motivated in part by the sense (based on anecdotal evidence) that if an event is rare, the ability to handle it may be overestimated.

The algorithm proceeds through three distinct phases:

Initialisation

1) The *likelihoods* of all *Threats* and *Consequences* are set to the lowest level.
2) The inferred levels of all *Trustworthiness Attributes* are set to their default (generally ''Very High'') level unless the risk analyst has chosen to assert that the level should be lower.

Likelihood and Trustworthiness Level Calculation

3) The *likelihood* of each *Primary Threat* may be increased according to the levels of the *Trustworthiness Attributes* that influence it, combined with its *frequency* and with a ceiling given by the *effectiveness* of any relevant *Control Strategies* that are fully present.
4) The *likelihood* of each *Secondary Threat* may be increased according to the *likelihood* levels of the *Consequences* that influence it, combined with its *frequency,* and tempered by *Control Strategy effectiveness*.
5) The *likelihood* of each *Consequence* is set to the most likely causal *Threat*.
6) Where a *Consequence* is linked to a *Trustworthiness Attribute*, the *level* of the *Trustworthiness Attribute* may be reduced as the *Consequence's likelihood* increases.
7) The process iterates back to step (3) until no further changes are made. Note that the algorithm must converge as *likelihoods* are only ever increased and *trustworthiness levels* only ever decreased and both scales are discrete and finite.

Risk Level Calculation

8) The ''direct'' *risk level* for each *Consequence* is calculated from its inferred *likelihood* level and asserted (or default) *impact* level using a lookup table (such as Table 1).
9) The ''system'' *risk level* for each *Threat* is set to be the highest *risk level* that it causes at a *Consequence* either directly or indirectly.

VI. VALIDATION

In this section we validate our approach against a representative cyber-physical system, describing a ''Network'' Knowledgebase to model risks in a well-known documented attack on a German steel mill, along with how the approach aligns with ISO 27005 process and terms.

TABLE 1. Mapping from likelihood and impact levels to risk level.

Specified Impact	Calculated Likelihood: Negligible	Very Low	Low	Medium	High	Very High
Negligible	Very Low	Very Low	Very Low	Very Low	Very Low	Very Low
Very Low	Very Low	Very Low	Very Low	Very Low	Low	Low
Low	Very Low	Very Low	Very Low	Low	Low	Medium
Medium	Very Low	Very Low	Low	Medium	High	High
High	Very Low	Low	Medium	High	Very High	Very High
Very High	Very Low	Low	Medium	High	Very High	Very High

The validation uses the Spyderisk software, available on GitHub [2] under the Apache 2.0 licence, which implements the risk assessment approach described in this paper and supports the ISO 27005 process. There are various Spyderisk sub-projects, including:

- the "Spyderisk System Modeller": a multi-user web service with a rich graphical client (see Section VI-A);
- the "Network" Knowledgebase, describing socio-technical information systems (see Section VI-B);
- tools for maintaining Knowledgebases and generating documentation;
- user documentation; and
- deployment scripts.

In the following sub-sections we briefly describe the Spyderisk software and the Network Knowledgebase, then discuss how their combination relates to ISO 27005 and show how they can be used to reproduce and analyse an attack on a steel mill seen in Germany in 2014.

A. SOFTWARE IMPLEMENTATION

The Spyderisk software is implemented as a multi-user Java web service [47]. The service exposes various RESTful API endpoints, both for use by the integrated graphical client (see Fig. 2) and by other non-graphical clients. The service is multi-user, with private user accounts, and authentication performed by a Keycloak service [48]. The graphical client is web-based and provides each risk analyst with a dashboard (listing their models) and with a model editing and exploration interface. Risk analysts can build and edit models using drag-and-drop, run the risk assessment, explore the attack paths, and see the effect of adding and removing *Controls*. Models can be shared between risk analysts as read-only or with write access and the software has a built-in help system.

The service is generic, not tied to networked information system analysis, and must be configured with one or more Knowledgebases to become functional. User documentation for the Spyderisk System Modeller is available [49].

B. NETWORK KNOWLEDGEBASE

The Network Knowledgebase [50] is used to model networked cyber-physical systems. In this paper we have used version 6a3-1-4. In combination with the Spyderisk System Modeller, it provides a thorough risk assessment of information systems across a wide range of assets, threats and controls and so demonstrates use of the core ontology. Documentation for the Network Knowledgebase is available [51] and we summarise its scope in the following sub-sections, giving some examples of the various Core Ontology classes previously described.

1) STRUCTURAL CLASSES

The asserted *Asset* class hierarchy is summarised in Fig. 3. There are also various inferred *Asset* classes, including some that are only used transiently as part of *Construction Pattern* sequences.

Generally, a new *Asset* class is added only because it is necessary to identify a type of *Threat*, provide a location for a type of *Control*, or provide a variation on the trustworthiness properties which influence *Threat* likelihoods. For example, various types of Process are included in the Network Knowledgebase, with Application Process as the generic one. Other process types are included to identity the risk of different *Threats*, such as SQL injection attacks specific to Databases, or web application threats involving Web Browsers.

In total there are 205 *Asset* classes with 64 of them being assertable. There are 179 *Relationship* types with 43 being assertable.

The asserted *Relationships* for some *Asset* classes are shown in Table 2. As relations are directed, the table is asymmetric. *Relationships* can be defined between any two classes in the *Asset* hierarchy and are inherited by the sub-classes.

2) ASSET CONFIGURATION CLASSES

Trustworthiness Attributes and Consequences

Each *Asset* type in the Network Knowledgebase has its own set of *Trustworthiness Attributes*. The "Adult" *Asset* type for instance has six *Trustworthiness Attributes*:

- Astuteness: the ability to avoid insecure behaviour and detect attempted deception by malicious agents.
- Availability: the likelihood that the adult will be available to carry out their role in the system.
- Benevolence: how free of unprovoked malicious motives the adult is.
- Competence: the ability to perform their system role correctly even if presented with incorrect inputs.

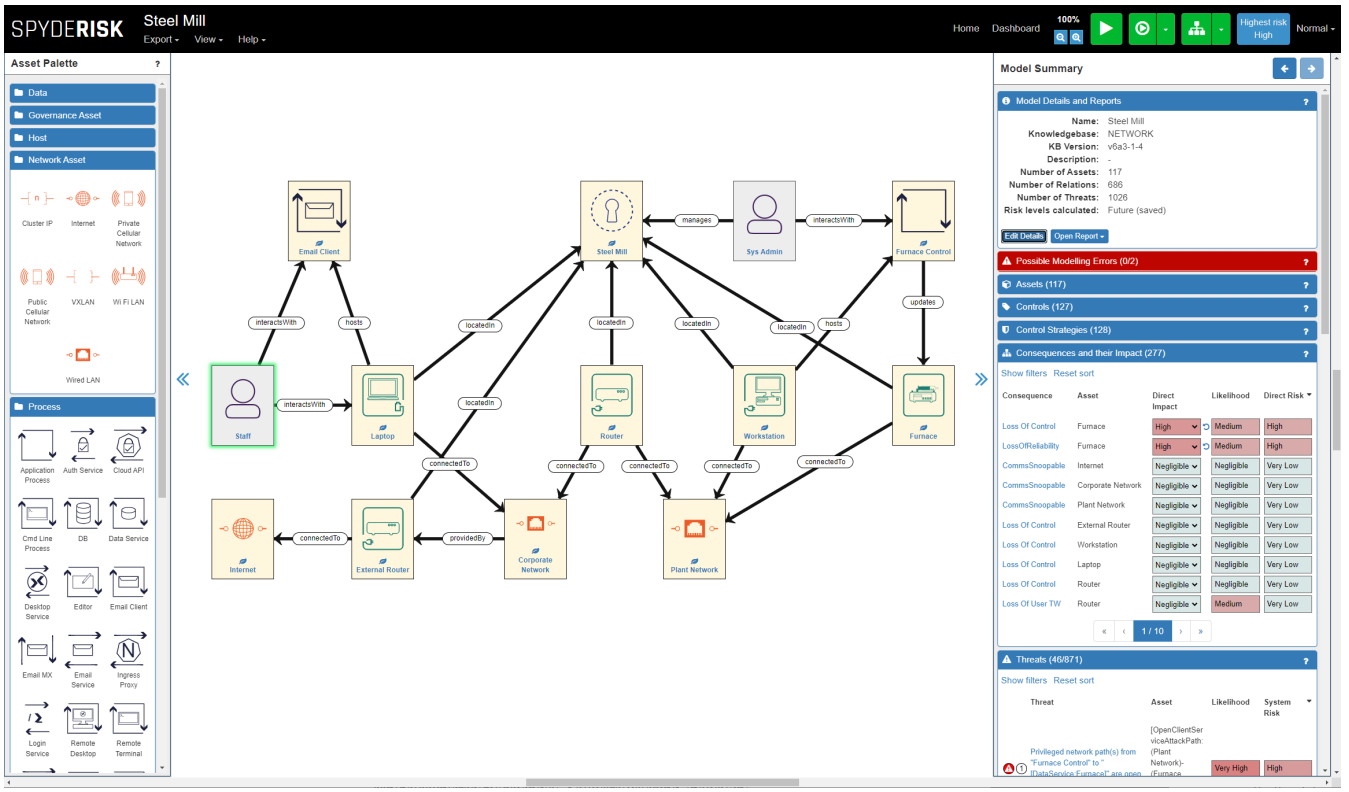

FIGURE 2. Screenshot of the spyderisk system modeller web interface.

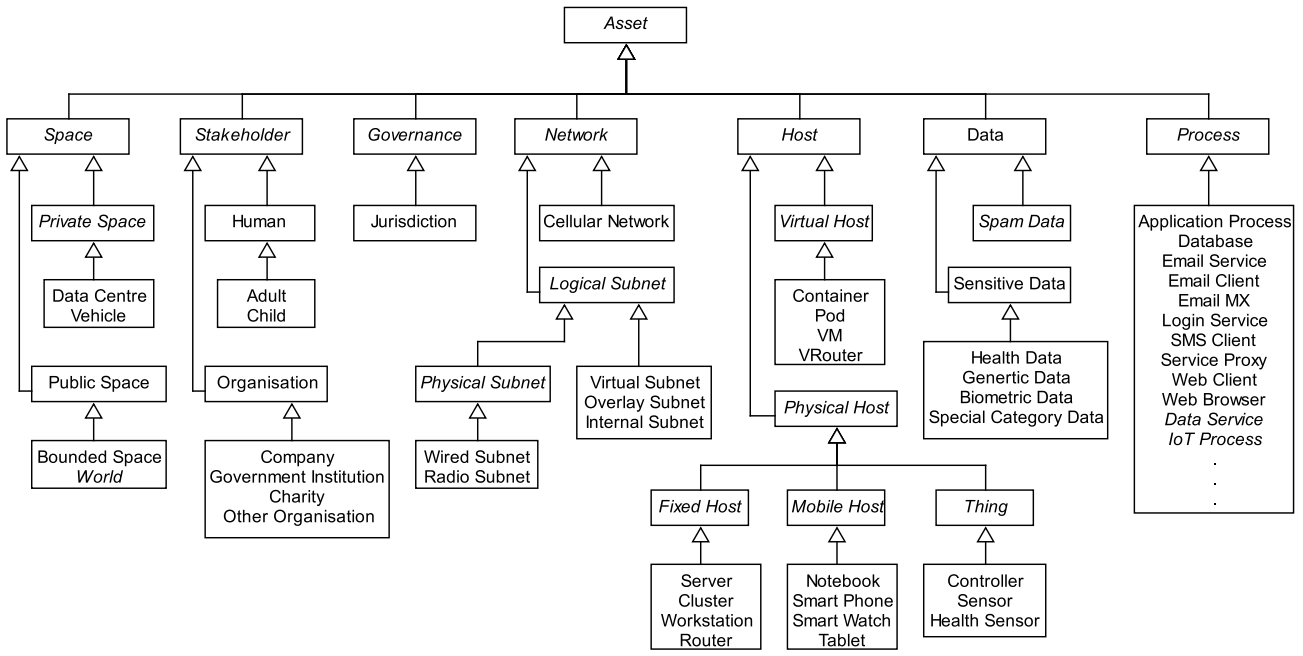

FIGURE 3. The assertable asset class hierarchy for the "Network" Knowledgebase. Classes in italics are not assertable. Boxes with multiple lines of text represent multiple separate classes all with the same inheritance. Not all classes nor all the hierarchy is shown, in particular there are more sub-classes of process.

- Reliability: the propensity to avoid unprovoked unintentional errors.
- Timeliness: the likelihood that the adult will have up to date inputs on which to base their actions.

TABLE 2. Asserted relationships for some network knowledgebase asset classes.

Source		**Notebook**	**Application Process**
Target	**Notebook**	Pairs via Bluetooth Pairs via USB	Controls
	Application Process	Hosts	Controls Uses Uses for AuthN
	Data	Stores	Amends Appends Creates Reads Receives Serves Updates
	Wired LAN	Connected To	Controls

The concepts of availability, benevolence and reliability/competence are commonly found in academic work on human factors that contribute to trust [52]. Our additions of astuteness and timeliness reflect the human's role in a socio-technical system subject to cyber-attack. ''Competence'' is often found in other works, and astuteness is a particular facet of this attribute relating to cyber-security. ''Timeliness'' is more a reflection on the impact the rest of the system has on the human's work. In relation to *Threats,* a *Threat* representing a phishing attack involving an adult *Asset* in the system would be made more likely if the adult's ''Astuteness'' *Trustworthiness Attribute* had a low level for example.

The ''Data'' *Asset* type *Trustworthiness Attributes* include the standard ''CIA triad'' from Information Security of ''Confidentiality'', ''Integrity'' and ''Availability'', but three more are added: ''Authenticity'', ''Health'' and ''Timeliness''. ''Health'' relates to whether the Data *Asset* is free from self-propagating malware. The Authenticity attribute relates to whether the Data *Asset* is what it claims to be, i.e., it is neither forged nor altered in a way designed to induce false behaviour in other assets consuming the data. If Authenticity is lost, then loss of Integrity automatically follows.

The close link in the Core Ontology between *Consequences* and *Trustworthiness Attributes*, with every *Trustworthiness Attribute* being undermined by a *Consequence*, means that many *Consequences* are phrased just as ''Loss of'' the Trustworthiness Attribute. For example: ''Loss of Astuteness'' or ''Loss of Availability'' in the case of the Adult.

''Application Process'' *Assets* have the following *Consequences*:

- Loss of user trustworthiness: untrusted, potentially malicious agents gained user rights in some system context.
- Loss of availability: the asset cannot (or will not) carry out its function within the system, failing to interact with other assets as expected.
- Loss of intrinsic trustworthiness: deterioration in the quality and/or integrity of software engineering used to implement the asset, such that it will contain more functional software bugs that cause errors or crashes without external provocation.
- Loss of reliability: the process is liable to make errors with an unacceptable frequency or extent. Caused by internal failings (e.g., software bugs), by using forged, corrupt, or inaccurate information as input, or by a dependency on some other asset that is not reliable.
- Loss of timeliness: represents a state in which a process has outdated or (temporarily) unavailable inputs.
- Malware infection: insertion into the asset of malicious, self-propagating software.
- Overloaded: the asset is being used or requested more than allowed or expected.
- Theft of control: untrusted, potentially malicious agents gained control of a device or process running on it after the device has been removed by theft from the system.

Controls

There are numerous *Controls* in the Knowledgebase which support the *Control Strategies* found below. *Controls* on the ''Host'' *Asset* include that it has a secure BIOS, up to date patched software, or anti-malware installed. *Controls* on the ''Application Process'' *Asset* class include the presence of logging or X509 service verification. *Controls* on the ''Adult'' *Asset* class include that they have received security training, hold a physical ID, or have been through a screening process.

3) LIKELIHOOD, IMPACT AND RISK SCALES

The various ''levels'' previously referred to are scales defined in the Knowledgebase for likelihood and trustworthiness, the impact of consequences, and risk levels. The Network Knowledgebase uses five or six points on each scale, but this is not mandatory. However, each point on a scale needs to be an order of magnitude different to the next for the likelihood calculation to work.

Likelihood and trustworthiness are negatively associated providing two inverse views of the same scale. Providing both as relative concepts is helpful as it makes thinking and describing some aspects of the model easier, for example if something is *very trustworthy* in some way then your expectation is that there is a *very low likelihood* of it behaving in an adverse way. The Network Knowledgebase scales are shown in Table 3.

To be able to compute a risk level, the impact and likelihood of an event must be known. The Network Knowledgebase scale for impact is shown in Table 4.

The risk levels in the Network Knowledgebase follow a five-point scale, as described in Table 5.

4) CONSTRUCTION PATTERNS

The Network Knowledgebase contains 423 construction patterns. The three most important categories of information that are inferred through Construction Patterns are:

- network assets (interfaces, routes, paths);
- client-service communications; and
- data lifecycle (data flows, stored copies).

TABLE 3. Likelihood and trustworthiness levels in the network knowledgebase.

Likelihood Level	Meaning	Trustworthiness Level
Very high	Expected within minutes	Very low
High	Expected within hours to days	Low
Medium	Expected every year or so	Medium
Low	Possible but not inevitable within the lifetime of a typical system	High
Very low	Rare within the lifetime of a typical system	Very high
Negligible	The possibility can be ignored	Safe

TABLE 4. Impact levels in the network knowledgebase.

Impact Level	Meaning
Very high	The Threat or Consequence is fatal to key business objectives, and must be prevented at all costs
High	The Threat or Consequence causes a serious loss of business functionality that will be fatal if not stopped very quickly
Medium	The Threat or Consequence causes a serious but non-fatal loss of business functionality or efficiency
Low	The Threat or Consequence causes a moderate loss of business functionality or efficiency
Very low	The Threat or Consequence causes a small loss of business functionality or efficiency
Negligible	The Threat or Consequence has no significant direct impact on business functionality or efficiency

By analysing the relations between network elements, hosts and routers (including a model of Network Address Translation - NAT), the connectivity across the system model is established, encompassing mobile hotspots, cellular networks, host pairing (via Bluetooth or USB), Wi-Fi and wired networks, the hosts, and their network interfaces.

Links between clients and services are often key to the operation of an information system and must be analysed so that the software can identify any *Threats* to them, such as snooping, spoofing, or disrupting an element of the network path that the communication goes over. The *Construction Patterns* infer all client-service pairs and link them to the network paths and interfaces already inferred.

Understanding where Data or copies of Data are in the system, how Data is changed (where and by what) and through which Network Paths data moves is key to many of the *Threats* that need to be identified and mitigated. The fine-grained relationship types between Process and Data Assets and their links to Hosts (see for example, Table 2) supports a complex analysis of the data lifecycle. The inferred model includes data in memory (when used by a process), data in transit between processes (and what path it takes), storage of data on hosts and copies of data. The inferred data-flows are also linked to the client-service relationships already inferred.

The ''context'' of *Assets* is also added to the inferred model, including the physical locations of devices, and the networks to which they have access. Assets associated with portable devices will have multiple location contexts, and modelling the context allows attack paths to be formed that are consistent, e.g., a compromise in one location does not allow direct access in another, but data accessed in one location may be exposed in another.

The Network Knowledgebase also includes a model of a Data Centre which brings in a router, cluster of physical hosts and a wired LAN. Furthermore, if a Kubernetes Pod is included as ''managed by'' the Data Centre then additional inferred assets are added representing Kubernetes virtual infrastructure such as additional subnets, proxies, and Kubernetes Nodes.

These *Construction Patterns* play a key role in creating the rich connectivity across System Models and supporting the long-ranging cause and effect chains modelled by the *Threats*.

5) THREATS

The Network Knowledgebase includes a library of 595 generic localised *Threats* which are chained together during the risk calculation. The library covers the full spectrum of threats, such as natural threats, accidental threats (e.g., hardware failures or software bugs) and human threats including malicious attackers and people making mistakes. The *Threats* in the library are generic, in that they do not relate to specific CVEs [1] as found in some risk-assessment software. For instance, the high-profile ''Log4shell'' CVEs [53] would be taken into account by generic *Threats*. The trustworthiness attributes used to describe vulnerabilities are based on a mapping from Common Vulnerability Scoring System (CVSS) metrics [54], and the corresponding *Threats* cover the means of access (e.g., remote/local, authenticated/anonymous access), and the result of successful exploitation (e.g., privilege escalation or denial of service).

The scope of the *Threats* that the Network Knowledgebase covers is:

Access and Control Privileges: representing situations where an untrustworthy agent with certain privileges can gain access to further privileges, related to resource access and control.

Insider Attacks: representing situations where a legitimate user or organisational stakeholder performs malicious actions. In most cases, this is modelled by reducing the trustworthiness of processes and devices they are operating, managing or using.

Exploiting Vulnerable Software: representing situations where an attacker can cause execution of vulnerable code and thereby gain temporary use of privileges.

Other Malicious Attacks: representing situations where a malicious attacker exploits a weakness other than a software vulnerability.

Exploitation of Stolen Devices: theft is a physical threat that leaves an attacker in possession of a device. These threats cover actions such an attacker can then take, bearing in mind that: they control the device itself, but after disconnecting it from the rest of the system. Reconnection to the system

may be possible in specific contexts, not necessarily those in which the stolen device is normally used; they can only gain partial access to system functions and privileges.

Non-Malicious Threats: representing the effect of accidents and unintentional errors that could cause problems without provocation by malicious attackers.

Compliance Threats: representing breaches of regulations (e.g. GDPR), best practice guidelines, etc.

Potential Modelling Errors:representing situations which often occur when the risk analyst has made an error. The potential modelling errors are highlighted separately to normal *Threats* and can be reviewed and in some cases dismissed by asserting controls.

6) CONTROL STRATEGIES

The *Control Strategies* in the Network Knowledgebase include:

Organisational measures: staff screening, training, policies.

Physical Security: Controlling physical access to spaces. E.g., physical locks & keys, chip & PIN, biometrics, ID checks.

Service Security: access control and privilege restriction mechanisms. E.g., TLS, AuthN, passwords, strong password, OTP, SMS codes, X.509, etc.

Software Security: software testing, pen testing, patching, device certification.

Data Security: encryption of data flows or stored copies; replicated storage. E.g., encryption, keys, replication, data access control, DB access control.

Network Security: network access control (encryption, network AuthN) and routing restrictions. E.g., radio subnet encryption, network AuthN via X.509, PSK, SIM, Bluetooth SSP, EAP-TLS, EAP-PSK, etc, blocked segments and interfaces, bandwidth management, DoS filter, etc.

Client Security: spam filtering, passwords.

Device Security: controlling direct access to devices; preventing alteration of software on devices. E.g., login password checks, chip & PIN, biometrics, anti-malware, secure host configuration, secure BIOS, remote wiping.

Resource Management: elastic hosting, process prioritisation.

User Intervention: representing user actions or depending on user actions. E.g., disabling vulnerable devices to protect them from attacks.

As described above, it is the *Control Strategy* that links *Controls* on *Assets* to *Threats*, defining what combination of *Controls* are required and how *effective* they are in reducing the *Threat likelihood*.

For example: a client verifying the identity of a service using an X.509 certificate is represented by the ''X509'' control at an Application Process (such as a web server) combined with the ''X509 service verifier'' control at a client (such as a web browser), which together make up the ''Service AuthN X509'' *Control Strategy* and give a Safe effectiveness against spoofing attacks, but one of those controls alone would have no effect. Given that there could be multiple client types using the same service, it is important to indicate separately which controls are in place where.

The *effectiveness* of the majority of *Control Strategies* is set to Safe, that is, they are completely effective against the *Threats* they address. The *effectiveness* is set to less than ''Safe'' in the cases where, considering temporal sampling, the *Control Strategy* will not always work. For instance, the ''Patching at host'' Control Strategy, representing a systematic procedure for regular security patching of software used on a host has a High effectiveness because there is inevitably a delay between software being found to be vulnerable and a patch being created and applied.

Where a *Threat* should only be considered if a *Control Strategy* is present, we say it is ''enabled'' by the *Control Strategy*. For example: the ''client password access'' *Control Strategy* which describes a password held by a client and verified at a service, enables the *Threats* describing password sniffing or credential stuffing attacks which would otherwise be ignored.

C. RELATION TO ISO 27005

The Network Knowledgebase assists in risk assessment of networked information systems. ISO 27005 builds on the risk management process specified in ISO 31000 [55] and can be used to support risk assessments of information systems, as suggested in ISO 27001. In this section we go through the steps in the ISO 27005 process and show how they relate to the concepts used in our approach, a summary of which can be found in Fig. 4.

Context Establishment is a pre-requisite for the use of our approach and puts in place the various criteria, the scope, and boundaries of the risk management process (i.e., what should be included in the system model) and gathers information about the environment that the organisation operates in.

Risk Identification describes an asset-based and an event-based approach. We follow the asset-based approach with the Spyderisk risk analyst needing to identify the assets (and their relations), the existing controls and some aspects relating to vulnerabilities (specifically, where the *Trustworthiness Attributes*of *Assets* differs from the default). Other aspects of vulnerability identification and all the processes of identifying *Threats* and *Consequences* are automated.

ISO 27005 describes an asset as ''anything that has value to the organization and therefore requires protection''. Assets that have ''value'' to the organisation are described as ''primary assets'' (commonly data assets but also business processes): these relate directly to ''Data'' Assets in the Network Knowledgebase and to ''Application Processes'' which support the primary business processes. To provide sufficient information for the risk assessment many supporting assets (e.g., ICT and networking infrastructure) and their inter-relationships also need to be specified as part of the System Model and are therefore defined in the Knowledgebase. ISO 27005 A.2.2 says that ''it is important to identify the

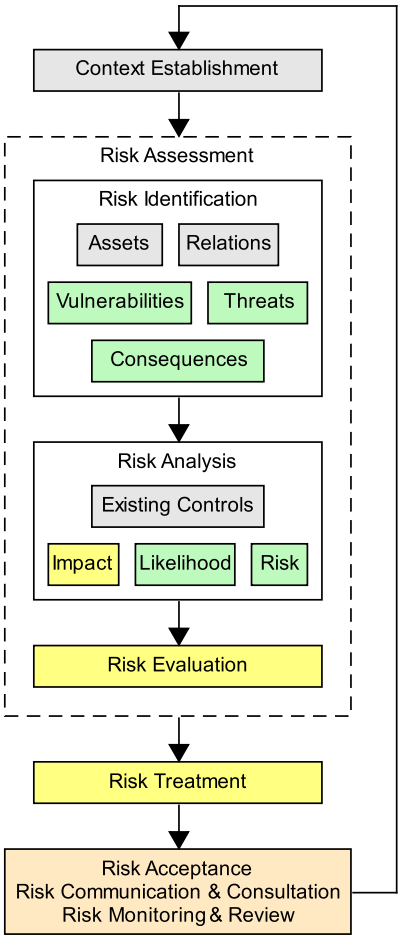

FIGURE 4. **The information security risk management process described in ISO 27005. Spyderisk automates the green boxes, supports decisions required in the yellow boxes and assists with the three processes coloured beige through system documentation.**

relationships between the assets, and to understand their value to the organization'' and suggests asset dependency graphs ''are useful tools to represent such dependencies and ensure that all dependencies have been considered''. Spyderisk's rich representation of *Relations* between *Assets* and automated dependency analysis supports this methodology.

Not much guidance is provided in ISO 27005 regarding how to identify the threats to the system. It proposes identifying the assets and their vulnerabilities and considering potential threats, with a brief table of threats provided in the Appendix. It states that ''If all valid combinations of assets, threats and vulnerabilities can be enumerated within the scope of the ISMS, then, in theory, all the risks would be identified.'' Clearly, without any automation a complete analysis would be difficult.

Our model does not include anything called a ''vulnerability'', but vulnerabilities are certainly captured. Many vulnerabilities in catalogues (such as those in ISO 27005 Annex A) are modelled as the absence of *Controls* and others through low *Trustworthiness Attributes*.

Our Network Knowledgebase contains a catalogue of generic threats along with the information necessary to identify when and where each type of threat is present in the System Model. The threats present in the system are identified automatically which reduces the time required for risk analysis while also improving repeatability, consistency and completeness of analysis.

Every threat in our catalogue describes its direct *Consequences*, so these are automatically identified and applied to the System Model. Furthermore, through the iterative procedure already described the *Consequences* of one threat can increase the *likelihood* of another *Threat* and so can have far-reaching consequences via the *Relations* between the *Assets*.

Risk Analysis comprises the assessment of the business impact or (''consequences'') of any identified ''risk scenarios'', the assessment of the likelihood and the determination of the risk level.

We take the approach of the risk analyst focussing on the primary *Assets* in the system. For these assets, they are expected to specify the business *impact* of any adverse *Consequences* for that asset. For example, if a company was hosting an important public data set, the *Consequence* of ''loss of confidentiality'' would be negligible (as it is intended to be public), but the *Consequences* of ''loss of availability'' or ''loss of integrity'' may have a significant impact on the business. In general, there is no need to specify the impact level of *Consequences* at the supporting assets because the threat propagation technique will take account of any knock-on effects on the primary assets.

ISO 27005's says that the likelihood calculation should consider ''existing controls and how effectively they reduce known weaknesses''. We describe controls in the Core Ontology with the *Control* and *Control Strategy* classes. In our ontology, the *effectiveness* attribute (which limits a *Threat likelihood*) is defined on the *Control Strategy* class, not on the *Control* itself.

The calculation of the *likelihoods* of *Threats* and *Consequences* is automated, and the combination of the *Consequence impact* levels (either explicit or default) with the calculated *likelihoods* provides automated *risk levels* of *Threats* and *Consequences* as described previously in Section V-D.

Risk Evaluation takes the *Threats* with their risk levels and determines which can be accepted as they are, and which need to be treated (in a prioritised list). The decision about which *Threats* need to be treated is done by the risk analyst and in general they would examine the *Consequences* with the highest *risk levels* (which are clearly shown in the software interface) as a starting point.

Risk Treatment is the process of taking the risk assessment from the previous steps and determining what to do (if anything) about the identified *Threats*. The options in ISO 27005 are to modify, retain, avoid, or share each *Threat*. The Spyderisk software provides an interactive and contextualised help system that guides the risk analyst by proposing

and applying *Control Strategies* to reduce the risk levels of *Threats* (''modifying'' them) with *Controls* added at this point being marked as ''*to do*''. Finally, the System Model structure may be redesigned to avoid *Threats* if appropriate, e.g., if no risk reduction controls are available. Once risk levels are low enough, the remaining risks (and Threats causing them) are assumed to be accepted.

The Spyderisk software can produce reports from the data held in the model, including a ''risk treatment plan'', which is the formal output of this step, based on the *Controls* marked as ''*to do*''.

Risk Acceptance, **Risk Communication and Consultation**, and **Risk Monitoring and Review** are out of scope for Spyderisk but supported by the model and reporting. The detailed System Model provides a documented record of the risk analysis and can support communication between stakeholders, and its existence makes subsequent analysis much simpler, including any refinements following monitoring and review.

D. ILLUSTRATION AND VALIDATION CASE STUDY

We illustrate and validate the approach described using a documented attack on a German steel mill [56]. The example is representative of a cyber-physical system as it considers humans and their interaction with an information system supporting the operational of a physical steel mill. The report on the attack is not able to explain every step taken but there is sufficient detail to construct a model representing a similar system and show that our approach (a) discovers the same likely attack vector and (b) can demonstrate how the risk level reduces with additional controls.

The report on the attack [56] provides the following summary (where we have highlighted key points relating to risk modelling in bold): ''The initial capability used to infiltrate the facility's **corporate network** was a **phishing email**. The BSI's report described this attack vector as 'an advanced social engineering' attack which multiple attackers used to gain access to the network. The adversaries then worked their way into the **production (ICS) networks**. From previous analysis of spear-phishing related incidents with ICS facilities it is highly likely that the email contained a document such as a PDF that when opened executed **malicious code** on the computer. This malicious code would have then **opened up a network connection for the attacker(s)** unbeknownst to the facility's personnel. No information has been presented on how the adversary moved into the production network but analysis of similar case-studies would indicate **probable traversal through trusted zones and connections between the corporate and plant network**.''

From this description, the system model structure shown in Fig. 5 was created. The model includes two networks: a ''Corporate Network'', with office computers such a ''Laptop'' connected to it, and a ''Plant Network'' to which the ''Furnace'' and ''Furnace Control'' software are connected.

The attack describes an email containing an attachment that is opened by a staff member on a computer connected to the Corporate Network, so a person or persons (''Staff'') using a ''Laptop'' connected to the Corporate Network and interacting with an ''Email Client'' have been included. A connection from the Corporate Network to the Plant Network has been added in the form of a ''Router''. As the report says, many industrial control system (ICS) networks were built as separate islanded systems but over time, business requirements have required them to be connected to corporate systems. The corporate network is also linked to the ''Internet'' via an ''External Router'' as this is clearly not an isolated system.

The steel blast furnace is represented in the model as an IoT controller (''Furnace''): something that can affect aspects of the physical environment in ways defined by data sent to it. We assume that there is an application process (''Furnace Control'') which updates the data controlling the Furnace and that the application process is hosted on a computer (''Workstation''), with the Workstation and Furnace both being connected to the plant network to enable inter-communication.

An element not mentioned in the report but assumed to be present, is the ''Steel Mill'' secure space where the physical hardware is all located. Without such a secure space being added to the model, the analysis would assume that the Furnace and all other hardware were unsecured in a public space which is clearly not realistic. A systems administrator (''Sys Admin'') is included in the model as someone who manages the Steel Mill space and interacts with the Furnace Control. In practice, this is likely to be two separate roles but splitting them makes no difference to the risk analysis.

As this is a model of the intended functioning system, no attacker or attack scenario is explicitly included in the model. The analysis will test all the possible attack combinations given the threats included in the Knowledgebase. *Assets* in the system model may represent a class of assets, so the single ''Staff'', ''Laptop'' and ''Email Client'' *Assets* can represent any size of workforce.

1) ASSET CONFIGURATION

Asset attributes were configured where necessary. Most *Trustworthiness Attributes* were left at their default values apart from the ''Astuteness'' *Trustworthiness Attribute.* This attribute indicates the general cyber-security competence of people, such as whether they choose a strong password or are likely to detect a phishing attack. The ''Astuteness'' of the ''Sys Admin'' was set to High, but that of the ''Staff'' was left at the default of Medium. The only other *Trustworthiness Attribute* of note is the ''Network User Trustworthiness'' of the ''Internet'' *Asset* which has the default value Very Low, signifying the likelihood of external attacks on the rest of the system by Internet users outside the system.

The *Impact level* assumptions were then configured. The Furnace is the business's primary asset and so we have chosen to set the impact of some *Consequences* at the Furnace to

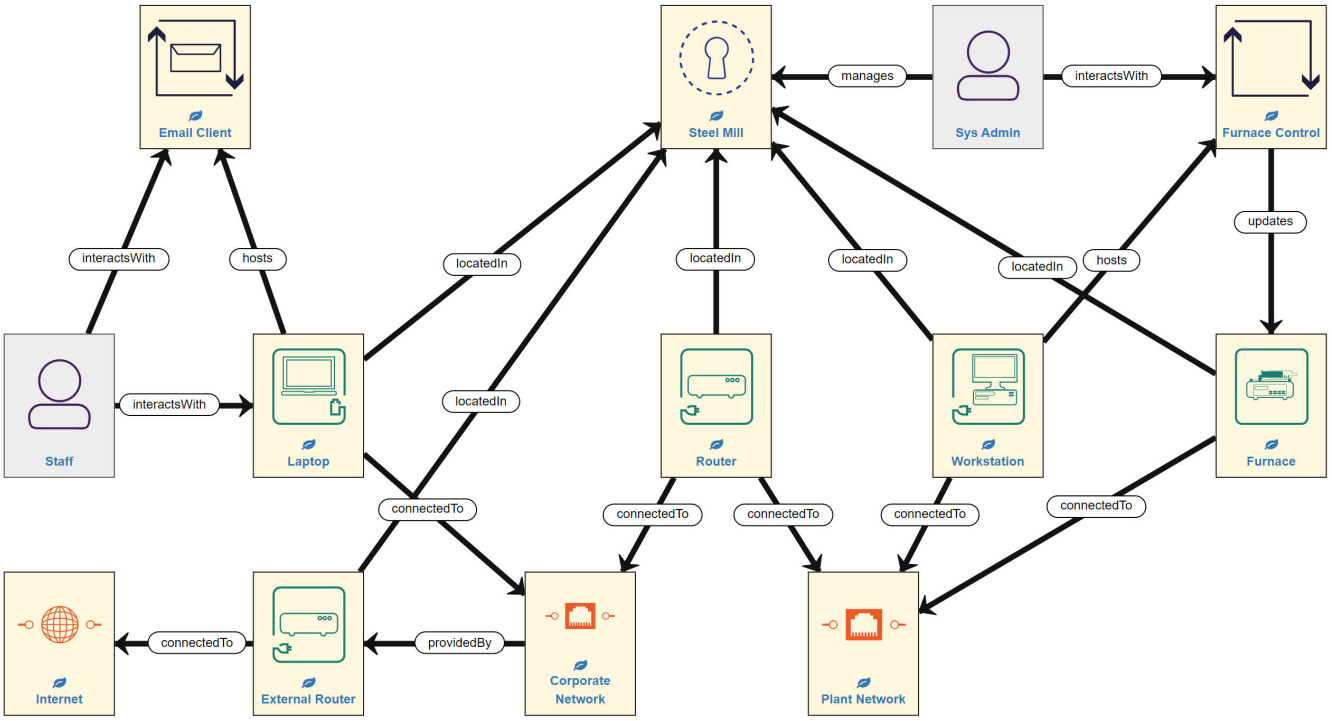

FIGURE 5. Screenshot of the steel mill system model.

TABLE 5. Risk levels in the network knowledgebase.

Risk level	Meaning
Very high	Cannot be tolerated, even if avoidance involves shutting down the whole system
High	Control measures should be introduced immediately, by shutting down parts of the system if necessary
Medium	Can be tolerated for a short time while measures are found that minimise loss of system functionality
Low	Can be tolerated for a longer time, until addressed by routine measures
Very low	Can be tolerated indefinitely, no action required as the risk can be accepted

TABLE 6. Configured impact levels for the furnace asset.

Consequence	Impact Level	Generic Meaning
Loss Of Control at Furnace	High	Untrusted, potentially malicious agents gain admin rights in some context
Loss Of Reliability at Furnace	High	The device, process or human is liable to make errors with an unacceptable frequency or extent. Caused by internal failings including lack of expertise, software bugs, etc., by using forged, corrupt, or inaccurate information as input, or by a dependency on some other asset that is not reliable

"High" and "Medium" impact instead of the default for IoT Controllers of "Negligible". The chosen parameters can be found in Table 6.

We have assumed a small baseline set of *Controls* in the System Model which would be expected, and which ensure that it is not wide-open to attack (see Table 7).

2) BASELINE RISK ASSESSMENT

The threat discovery process is launched with a single button-press in the Spyderisk software, which first causes the creation of inferred *Assets* and *Relations* via the *Construction Patterns*, and then the creation of *Threats,* taking less than one minute.

The validation example shown above contains 13 asserted assets and 19 asserted relations. Through the execution of the *Construction Patterns,* 104 inferred *Assets* and 667 inferred *Relations* are added.

An example *Construction Pattern* can be seen in Fig. 6. It matches the situation of a Host which "hosts" a Process and a Human which "interacts with" with the same Process. This pattern is found in the System Model in the "Staff" / "Email Client" / "Laptop" sub-system. The *Relation* "interacts with" is added between the Human and the Host (if it does not already exist). This saves the risk analyst from tediously adding such *Relations* to the System Model.

A more complex example, that adds in information that the risk analyst could not be expected to define, is shown in Fig. 7. The pattern transforms a single asserted IoT "Controller" *Asset* into a complex representation of an IoT "Controller", showing the embedded Data and Process *Assets*. By modelling the Controller with connected Data and Process *Assets*, we can apply the same *Threats* to it that are used across many

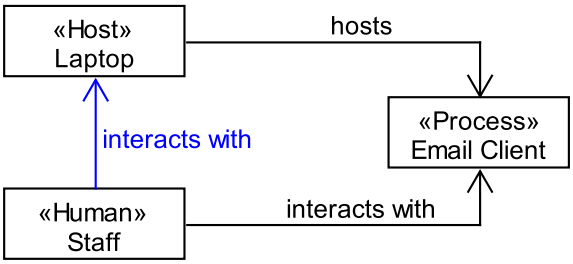

FIGURE 6. A construction pattern which adds a link (in blue) between the human and host.

parts of the System Model. When inferred *Assets* are created, they are automatically labelled by combining the *Asset* class and the label(s) of the asserted *Assets* in the pattern and adding square brackets.

Once the *Construction Patterns* have been executed, the System Model is searched for occurrences of all the *Threats* in the Knowledgebase. The search discovers 871 *Threats*, with 277 *Consequences*, and 128 *Control Strategies* comprising 127 (shared) *Controls*. Two of the *Threats* are described here.

The Network Knowledgebase *Threat* shown in Fig. 8 occurs in the steel mill System Model and represents the situation where an attacker remotely exploits a vulnerability which does not require authentication in the software of a Host (the ''Workstation''). The white and grey nodes in the Figure and their links represent the *Matching Pattern* that is looked for in the System Model, with the grey node being optional. The white nodes show the *Asset* type from the pattern along with the label of the *Asset* in the System Model that is matched in this case. The *Threat* includes the inferred Interface and Network Path *Assets*, found by the previously executed *Construction Patterns*.

As this is a *Primary Threat,* its *likelihood* is influenced by the *Trustworthiness Attributes*, shown as blue ellipses. The highest level *Trustworthiness Attribute* will determine the maximum likelihood of the *Threat*:

- the ''Extrinsic VN Trustworthiness'' of the Host indicates whether it has a vulnerability that can be accessed from a remote network;
- the ''Extrinsic AU Trustworthiness'' of the Host indicates whether it has a vulnerability that can be accessed without authentication; and
- the ''Network User Trustworthiness'' in the Logical Subnet indicates the trustworthiness of users on the remote network (which can communicate with the Host).

The *Consequence* (shown as a red ellipse) is to lower the ''Exploit Trustworthiness'' *Trustworthiness Attribute* of the Host: in a sense, recording that the situation has made an exploit on the Host more likely. If the Human *Asset* is present (as it is in the steel mill: the ''Sys Admin''), then the *Threat likelihood* can be limited by having a ''Software Patching'' policy in place on the Host and ''Security Training'' for the Human (so that the ''Sys Admin'' knows how to find and apply patches).

There are other similar *Threats* in the Network Knowledgebase representing other combinations of local, adjacent, and remote attacks and authenticated or anonymous access. Another set of *Threats* made more likely by reduced ''Exploit Trustworthiness'' on the host represent the various outcomes of an exploit. Such outcomes could be to disable the host, obtain user or admin rights on the host, or delete, read, or alter data on the host. These threats are automatically chained onto the one previously described to create threat paths representing all appropriate combinations.

The *Consequences* of *Threats* commonly go on to directly cause other *Secondary Threats* or to undermine related *Trustworthiness Attributes* which then enable other *Primary Threats*. This chaining of local cause and effect extends across the whole model creating a complex and powerful analysis. An example of a matched *Secondary Threat* can be seen in Fig. 9. The pattern represents the situation of the embedded Furnace control process receiving inauthentic data which then inevitably causes a loss of control in the Furnace itself. As with many Secondary Threats, there are no *Control Strategies* to prevent this step, as it is inevitable given the dependency between the involved assets.

With the *Threat* discovery process completed, and the *Asset Trustworthiness Attributes* and impact levels being defined as described above, the risk level calculation can then be performed, assigning a risk level to each *Consequence* and *Threat*. We then list the *Consequences* in order of risk, showing two high risk *Consequences*, as described in Table 8.

3) THREAT ANALYSIS

The Spyderisk software automatically shows that the root causes of both *Consequences* are the same:

1) Execution of malicious email attachment by ''Email Client'' on ''Laptop'' in ''Steel Mill''; or
2) Malware infection at ''Laptop'' due to execution of viral attachment in email client ''Email Client'' on ''Laptop''.

The analysis discovers various *Threats* that can cause loss of control or reliability in the Furnace, but some of these have low likelihood. The most important *Threats* to address first are those which cause (directly or indirectly) the Medium likelihood / High risk *Consequences* which we want to avoid. Attack graphs and trees [57] are an established way of describing possible chains of steps through a system which result in an undesirable consequence. Through our own tool [58] we are able to create and plot a variety of attack graphs using the cause-and-effect data contained in the System Model.

We generated an attack graph for the System Model including only the highest likelihood *Primary Threats* on the shortest paths from root cause to High risk *Consequences* to which we manually added three lower-likelihood nodes in grey (Fig. 10). The Figure shows the external environment

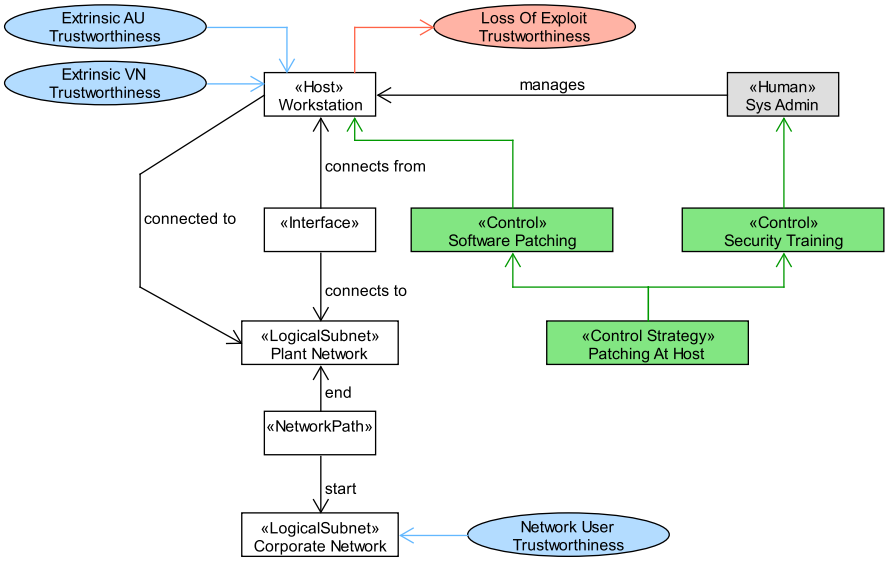

FIGURE 7. The primary threat representing an attacker exploiting a remote unauthenticated software vulnerability on the Host ("Workstation"). The white nodes and their links describe the pattern that is looked for in the System Model. The blue ellipses show the causes. The red ellipse shows the Consequence. The grey "≪Human≫" node is optional in the pattern so is not required for the threat to be present. The green nodes show the controls and control strategy.

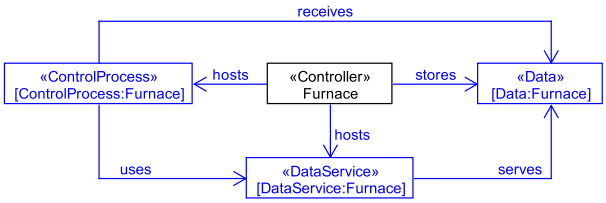

FIGURE 8. A construction pattern which transforms a single asserted Controller asset into a complex sub-system, adding the parts in blue.

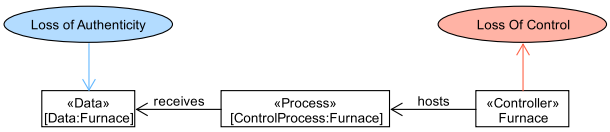

FIGURE 9. A secondary threat representing the inevitable loss of control over the Furnace if inauthentic data is received by its embedded control process.

and possible sequences of (deliberate) *Primary Threats* which lead to the *Consequences* having a Medium *likelihood* and hence High risk. In this Figure, the highest likelihood, shortest path *Threats* have Medium likelihood. Where branches join, a logic node has been added to avoid ambiguity. The initial gold nodes represent the environmental factors that influence both the likelihood of the initial root cause threat and the likelihood of the exploits. The gold node "Astuteness of Staff is Medium" influences the root cause likelihood (in red) of the 'Execution of a malicious email attachment by "Email Client" on "Laptop" in "Steel Mill"'. With no relevant *Control Strategies* in place, the likelihood of the root cause is itself Medium.

Once the attacker has access to the "Corporate Network" there are various options available, with three shown in Fig. 10. The report on the steel mill exploit described a loss of control of, and subsequent damage to, the furnace, which is seen in the left fork. The right fork illustrates how remote DoS attacks on either the "Furnace" or the "Workstation" (hosting the "Furnace Control") could both cause loss of reliability of the "Furnace". The 'Remote anonymous exploit on device "Workstation" from "Corporate Network" via "Plant Network"' in the left-most fork is a Primary Threat. To succeed, an exploit of the right type must be present in the "Workstation" and the attacker must be able to find it. The Medium "Extrinsic Trustworthiness" of the "Workstation" (in gold) results in a Medium likelihood of the necessary vulnerabilities being found ("VN" meaning one that can be exploited by sending a message from any network; "AU" meaning it can be exploited without authentication) resulting in this *Threat* having Medium likelihood. In addition, the "Workstation" must be connected to the "Plant Network" and a connection from the "Corporate Network" to the "Plant Network" must be permitted (as it is in this case). Once the attacker has moved to the "Workstation" they gain administrator privileges (through the "M" type vulnerability which permits control over the affected asset) and proceed to compromise the "Furnace Control" process, also using their ability to send messages on the "Plant Network", to gain access to the "[DataService:Furnace]" and inject "fake" content into the flow such that the attacker effectively has control of the "Furnace" and would be able to damage it.

The analysis matches the report of the actual incident: a malicious email attachment, opening a back-door and letting

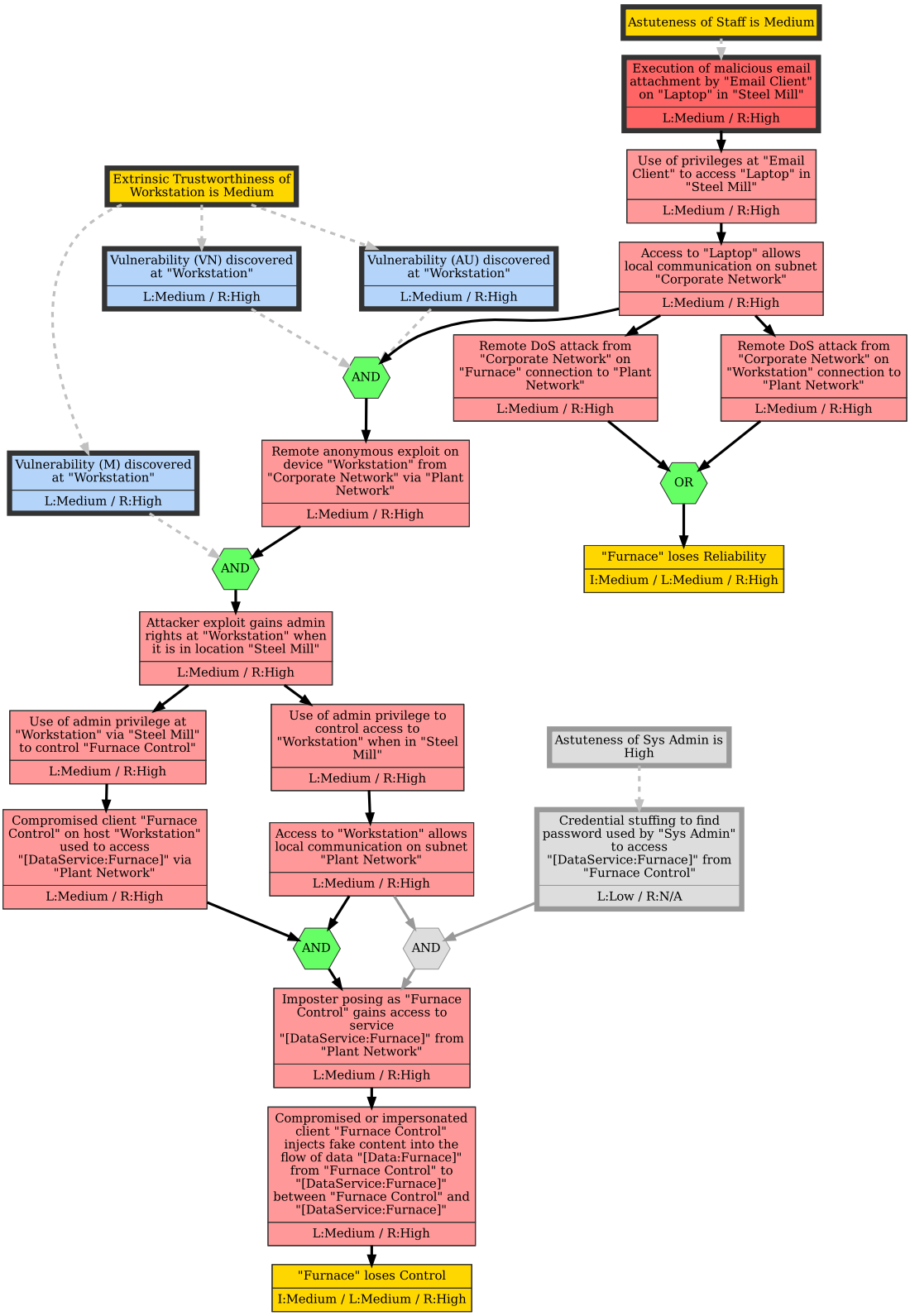

FIGURE 10. The shortest path, highest likelihood, attack graph for the baseline Steel Mill risk assessment, showing the "external causes" (gold, outlined) and "initial causes" (blue, outlined), root cause (red, outlined), Primary Threats along the paths (pink), and Consequences (gold). The grey nodes have been added for illustration. For Threats, the likelihood (L) and system risk (R) levels are shown; for Consequences the impact (I), likelihood (L) and direct risk (R) levels are shown.

TABLE 7. Controls added to the steel mill system model.

Control	Meaning	Assets
Access control	The asset (device or service) has an enforcement point (PEP) preventing unauthorised access. Normally used in conjunction with an authentication mechanism.	[DataService:Furnace]
Chip and PIN card	The human has a registered chip and pin card for identification purposes	Sys Admin Staff
Chip and PIN lock	A physical lock prevents access to a space, which incorporates a means to identity authorised users of the space using a chip and pin card	Steel Mill
Continuous occupation	Used at a private space to indicate that the space is secure due to it being continuously occupied at times when undetected physical intrusion is feasible, e.g., a user residence occupied at night when intrusion is most likely, or a business premises that operates 24x7	Steel Mill
Device certification	The device has been independently tested and certified as secure to a suitable evaluation assurance level	External Router
Password	The human has registered a password for identification purposes, which may be stored in a process acting on their behalf.	Furnace Control
Password verifier	The host or process has a means to verify a password given by an authorised user.	[DataService:Furnace]
Secure BIOS	The device has a secure BIOS and boot up sequence, ensuring its security cannot be bypassed by rebooting using an external (e.g., USB) boot device	External Router Laptop Router Workstation
Secure config	Removal of security vulnerabilities arising from insecure default configurations prior to entry of the affected device into the system	External Router Furnace Laptop Router Workstation

TABLE 8. The impact, likelihood, and risk levels for the high risk furnace consequences.

Consequence	Asset	Asserted Impact Level	Computed Likelihood Level	Resulting Risk Level
Loss of Control	Furnace	High	Medium	High
Loss of Reliability	Furnace	High	Medium	High

the attacker traverse from the corporate to the plant network, before taking control of the furnace.

To further illustrate the depth of the analysis and the mechanism of the likelihood calculation, three nodes in grey were manually added to the Figure. These represent part of one of many lower-likelihood paths which are present in the system: the (Low *likelihood*) activity of successfully guessing the ''Furnace Control'' password once the (Medium *likelihood*) ability to communicate on the ''Plant Network'' has been obtained. As both Threats are required (meeting in an ''and''), we combine their likelihoods by taking the minimum one (Low) to give the likelihood of the next step, but this Low *likelihood* route is masked by the higher likelihood path.

4) SECURING THE STEEL MILL

The Network Knowledgebase contains *Control Strategies* to reduce the likelihood of some of the *Threats* on the shortest path. The root cause itself can be addressed by three different strategies: security training at ''Staff''; spam filtering at ''Email Client''; or anti-malware at ''Laptop'' combined with software patching at ''Laptop''.

An alternative, which we have chosen, is to add a ''Firewall Block'' *Control* on the ''Router'' to prevent messages on the (inferred) open network segment from the ''Corporate Network'' to the ''Plant Network''. This is to improve the isolation of the ''Plant Network'' which should help in many cases, not just the most likely attack described above. Adding this *Control*, results in the likelihood of loss of control and reliability of the ''Furnace'' being reduced to Low (with a corresponding Medium risk level).

The attack graph summary is shown in Fig. 11. The first three *Threats* are the same as before and have Medium likelihood. With the connection between the networks blocked by the firewall the most likely next step is a remote anonymous exploit on the ''Router'', giving the attacker administrative rights. The exploit has a Low likelihood because the ''Router'' has a High Extrinsic Trustworthiness, meaning that the chances of vulnerabilities in its software being discovered (of the same types as previously described for the ''Workstation'') have Low likelihood. As the ''VN'' and ''AU'' vulnerabilities in the ''Router'' (with Low *likelihood*) along with the attacker being able to communicate with the ''Router'' from the ''Corporate Network'' (with Medium *likelihood*) are all required for a successful exploit, the minimum *likelihood* of the prerequisites is used to give a Low *likelihood* of a successful exploit.

Once the attacker has gained administrative rights on the ''Router'', the right-hand fork then shows that the attacker could re-enable routing between the networks (by removing the firewall block) and proceed with one of the DoS attacks as discussed previously. The DoS attacks now have a Low *likelihood* resulting from taking the minimum of the Low *likelihood* of removing the firewall and the Medium *likelihood* of access to the laptop: essentially the route to achieve them is now made harder by the need to first remove the firewall. The left-hand fork shows how the attacker, now able to communicate on the ''Plant Network'' could use the same credential stuffing attack shown earlier to gain access to the ''[DataService:Furnace]'' and then proceed as before to take control of the ''Furnace''. With the ''Sys Admin'' astuteness being configured as High, the credential stuffing attack has a Low *likelihood*. This is now the most likely path, in comparison to Fig. 10 where other attacks were easier.

To reduce the likelihood of loss of control of the ''Furnace'' even further, a second firewall rule can be added to prevent connections from the ''Corporate Network'' to the

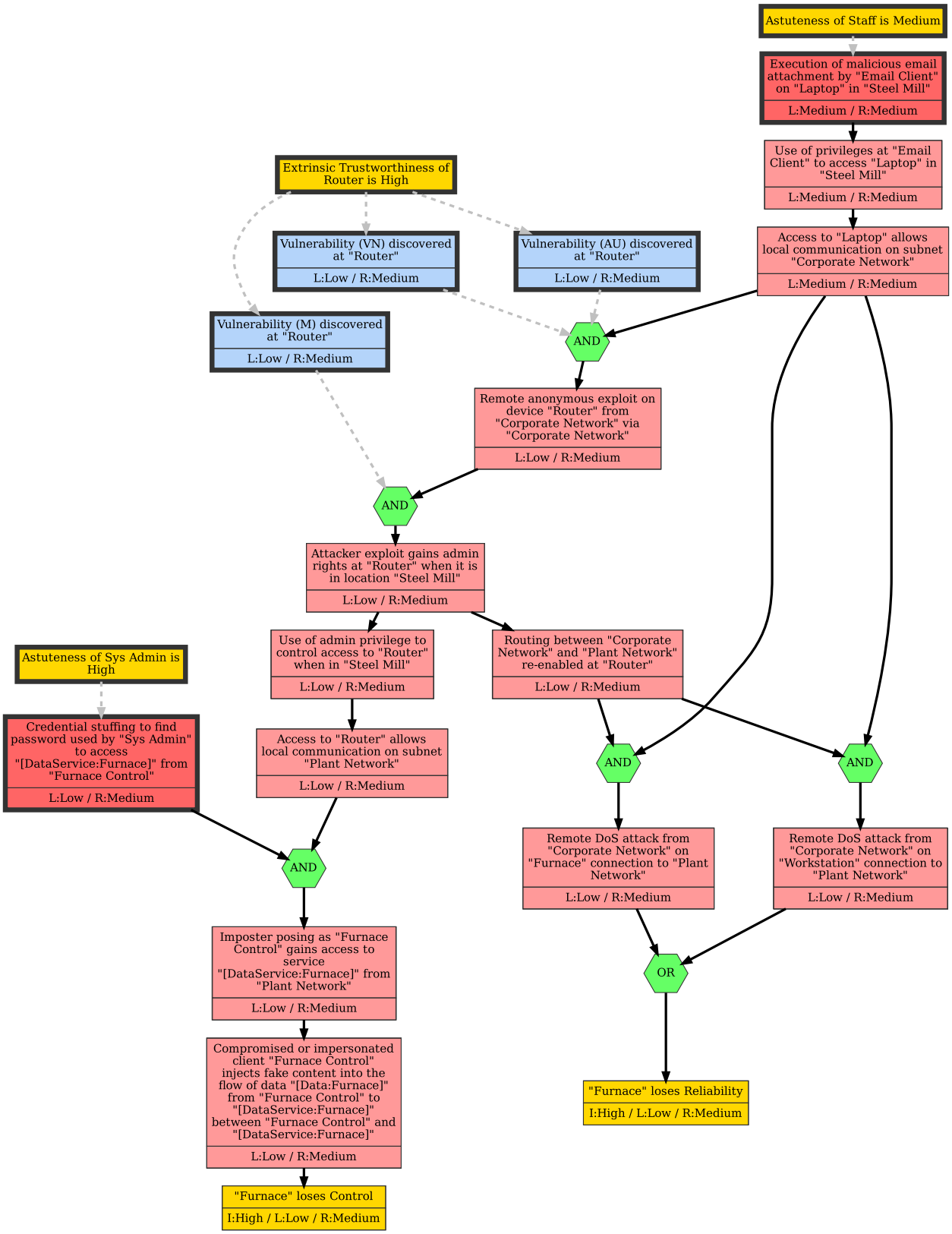

FIGURE 11. The shortest path, highest likelihood, attack graph for the Steel Mill risk assessment once network connections from the "Corporate" to the "Plant" network are blocked. Colours and abbreviations match the previous Figure.

"Router" itself, and the "Device certification" Control can be added to the "Router" (meaning that it certified as secure to some level). With these *Controls* in place the risk analysis results in a Negligible likelihood of loss of control at the

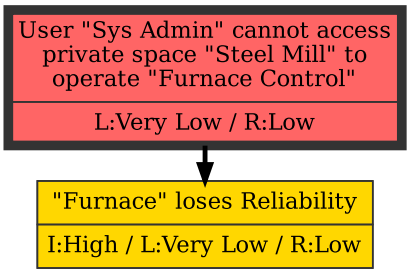

FIGURE 12. Attack graph summary for the secured system.

"Furnace" (with resulting Very Low *risk* level) and a Very Low *likelihood* of loss of reliability (with a Low *risk* level). The resulting attack graph is shown in Fig. 12.

The remaining Threat is that the Sys Admin may not be able to get into the Steel Mill secure space because of a problem with the "Chip & PIN" access control system (e.g., they may have lost their access card).

VII. CONCLUSION

This paper describes a simulation-based approach for automated risk assessment of complex cyber-physical systems to support implementers of ISO 27005. The approach uses a knowledgebase that defines classes of system assets and their possible relationships, along with the associated threats, causes and effects in a generic context. The target system can then be modelled in terms of assets and their relationships, and the knowledgebase used to identify threats along with their causes and effects, and to generate a cause-and-effect simulator. This allows threat likelihood to be determined from inputs describing trust assumptions and the presence of security controls in the system.

The approach has been implemented by the open source Spyderisk project [2], and validated by modelling a published case study of an attack on a German steel mill [56]. The case study shows some of the advantages of this approach – the ease with which potential threats can be found, the fact that the model documents trust assumptions and baseline security measures, the ability to analyse attack paths and secondary effect cascades, find potential control strategies, and explore the effect of adding further security measures to the system (including their side effects). A key point is that the risk analyst does not need to consider the impact levels to set on related supporting assets as this is done automatically by the software's risk calculation. The approach thereby supports some of the most difficult steps in implementing ISO 27005.

Creating the initial model of a target system does take time. However, the knowledgebase includes rules (Construction Pattens) to infer the presence of assets or relationships that users may overlook or whose inclusion is too laborious a task. These inferred assets represent network capabilities such as interfaces, routes and extended network paths, client-service trust relationships, and serialized and deserialized states in the data lifecycle (e.g., stored copies, data flows, and data held in volatile memory for processing). Other benefits over the lifetime of a system include the fact that building system models helps bring together different specialisms, allowing knowledge of the system to be pooled, and the ability to rapidly reanalyse risks if there are changes in the system, the trust assumptions, or if a new class of threat is discovered and added to the knowledgebase.

The implementation is now being applied to research problems in cybersecurity across a range of application sectors, including healthcare systems. To further validate the approach and the network knowledgebase, we are working with SINTEF to compare the Spyderisk approach with CORAS.

One part of our current research is about how to keep a model of a system up to date as a digital twin of an operational system, automatically adjusting the model to incorporate information from sensors (such as vulnerability scanners or SIEMs) and providing a short-term risk assessment with appropriate immediately actionable control strategy recommendations. Because the knowledgebase is a separate element of the Spyderisk toolset, one can customize the set of assets and threats in each application sector. We are developing of appropriate sub-models to capture privacy threats and controls, including those relating to synthetic data, and threats from and to the use of AI, especially applications of AI in healthcare. Another planned enhancement in the user interface is automated extraction of documentation on the input assumptions, security measures to be implemented, how they address each type of threat, and the residual risks.

Another potentially important area for investigation in future is how the generic concepts used in the cause-and-effect simulator relate to previous ontologies and standards. One example of this is the absence of the concept of 'vulnerability' in our core ontology, as discussed in Section IV. Another is how software vulnerabilities in assets like hosts and process (e.g., services) are modelled in the knowledgebase, using trustworthiness attributes aligned with the metrics from CVSS v2. This allows any new vulnerability to be added to a system model by altering trustworthiness levels and ensures that a wide range of vulnerabilities and exploits can be modelled. Frustratingly, CVSS v3 metrics proved less useful because the new features encourage users to consider indirect effects (in some system context), which may lead to inappropriate cause-and-effect relationships if applied in other contexts. This suggests the core ontology used in the cause-and-effect simulation may have value in the future development of security ontologies and standards, or in establishing best practice for their use.

Finally, the use of a cause-and-effect simulation can help with the modelling of value networks and other extended enterprises. Future work in this area will explore how models can be used both to communicate security information unambiguously between stakeholders and assess risks over the value network. This will build on previous research concerning trust modelling in multi-stakeholder systems but addressing the challenges of implementing ISO 27005 in multi-stakeholder networks, e.g., in IoT applications

involving device manufacturers, service providers and application operators as well as users.

ACKNOWLEDGMENT

Spyderisk builds on work done in 20 projects since 2008 at the IT Innovation Centre, University of Southampton, in the area of trust and risk assessment. Many researchers have contributed to its creation. In particular, still active at the IT Innovation Center and contributing to the development are Laura Carmichael, Nic Fair, Kenneth Meacham, Panagiotis Melas, J. Brian Pickering, Samuel Senior, and Daniel Shearer.

REFERENCES

[1] *CVE*. Accessed: Nov. 15, 2023. [Online]. Available: https://cve.mitre.org/
[2] *Spyderisk*. Accessed: Nov. 15, 2023. [Online]. Available: https://purl.org/spyderisk/paper/2023-overview/software
[3] *Information Security, Cybersecurity and Privacy Protection Guidance on Managing Information Security Risks*, Standard ISO/IEC 27005:2022, International Organization for Standardization (ISO), Oct. 2022.
[4] *Information Security, Cybersecurity and Privacy Protection Information Security Management Systems Requirements*, Standard ISO/IEC 27001:2022, International Organization for Standardization (ISO), Jul. 2023
[5] ENISA. *ENISA*. Accessed: Oct. 25, 2023. [Online]. Available: https://www.enisa.europa.eu
[6] G. Wangen, C. Hallstensen, and E. Snekkenes, "A framework for estimating information security risk assessment method completeness," *Int. J. Inf. Secur.*, vol. 17, no. 6, pp. 681–699, Nov. 2018, doi: 10.1007/s10207-017-0382-0.
[7] *Guide for Conducting Risk Assessments*, document (SP) 800-30 Rev. 1, National Institute of Standards and Technology, NIST Special Publication, Sep. 2012, doi: 10.6028/NIST.SP.800-30r1.
[8] C. Alberts, A. Dorofee, J. Stevens, and C. Woody, *Introduction to the OCTAVE Approach*. Fort Belvoir, VA, USA: Defense Technical Information Center, Aug. 2003, doi: 10.21236/ADA634134.
[9] jegeib. *STRIDE*. Accessed: Oct. 25, 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats
[10] K. Wuyts, L. Sion, and W. Joosen, "LINDDUN GO: A lightweight approach to privacy threat modeling," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Sep. 2020, pp. 302–309, doi: 10.1109/EuroSPW51379.2020.00047.
[11] J. Freund and J. Jones, *Measuring and Managing Information Risk: A Fair Approach*. Oxford, U.K.: Butterworth-Heinemann, 2015.
[12] M. S. Lund, B. Solhaug, and K. Stlen, *Model-Driven Risk Analysis: The CORAS Approach*, 1st ed. Cham, Switzerland: Springer, 2010.
[13] T. R. Peltier, *Information Security Risk Analysis*, 3rd ed. Boca Raton, FL, USA: Auerbach Publications, 2010.
[14] *Information Technology Security Techniques Information Security Management Systems Overview and Vocabulary*, Standard ISO/IEC 27000:2018, International Organization for Standardization (ISO), Feb. 2018.
[15] *NIST Cybersecurity Framework*. Accessed: Oct. 25, 2023. [Online]. Available: https://www.nist.gov/cyberframework
[16] S. Fenz and A. Ekelhart, "Formalizing information security knowledge," in *Proc. 4th Int. Symp. Inf., Comput., Commun. Secur.* New York, NY, USA: Association for Computing Machinery, Mar. 2009, pp. 183–194, doi: 10.1145/1533057.1533084.
[17] Ì. Oliveira, T. P. Sales, R. Baratella, M. Fumagalli, and G. Guizzardi, "An ontology of security from a risk treatment perspective," in *Conceptual Modeling*. Cham, Switzerland: Springer, 2022, pp. 365–379, doi: 10.1007/978-3-031-17995-2_26.
[18] É. Dubois, P. Heymans, N. Mayer, and R. Matulevič ius, "A systematic approach to define the domain of information system security risk management," in *Intentional Perspectives on Information Systems Engineering*. Berlin, Springer, 2010, pp. 289–306, doi: 10.1007/978-3-642-12544-7_16.
[19] V. Agrawal, "Towards the ontology of ISO/IEC 27005: 2011 risk management standard," in *Proc. Int. Symp. Hum. Aspects Inf. Secur. Assurance*, 2016, pp. 101–111.
[20] I. Meriah and L. B. A. Rabai, "Comparative study of ontologies based ISO 27000 series security standards," *Proc. Comput. Sci.*, vol. 160, pp. 85–92, Jan. 2019, doi: 10.1016/j.procs.2019.09.447.
[21] P. E. Kaloroumakis and M. J. Smith, "Toward a knowledge graph of cybersecurity countermeasures," MITRE Corp., McLean, VA, USA, Tech. Rep., 2021.
[22] D. Angermeier, H. Wester, K. Beilke, G. Hansch, and J. Eichler, "Security risk assessments: Modeling and risk level propagation," *ACM Trans. Cyber-Phys. Syst.*, vol. 7, no. 1, pp. 1–25, Feb. 2023, doi: 10.1145/3569458.
[23] G. Kavallieratos, G. Spathoulas, and S. Katsikas, "Cyber risk propagation and optimal selection of cybersecurity controls for complex cyberphysical systems," *Sensors*, vol. 21, no. 5, p. 1691, Mar. 2021, doi: 10.3390/s21051691.
[24] Z. Shi, K. Graffi, D. Starobinski, and N. Matyunin, "Threat modeling tools: A taxonomy," *IEEE Secur. Privacy*, vol. 20, no. 4, pp. 29–39, Jul. 2022, doi: 10.1109/MSEC.2021.3125229.
[25] D. Granata and M. Rak, "Systematic analysis of automated threat modelling techniques: Comparison of open-source tools," *Softw. Qual. J.*, vol. 32, no. 1, pp. 125–161, Mar. 2024, doi: 10.1007/s11219-023-09634-4.
[26] *Trike*. Accessed: Nov. 15, 2023. [Online]. Available: https://www.octotrike.org/
[27] *Microsoft Threat Modeling Tool*. Accessed: Nov. 15, 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool
[28] *OWASP Threat Dragon*. Accessed: Nov. 15, 2023. [Online]. Available: https://owasp.org/www-project-threat-dragon/
[29] *IriusRisk*. Accessed: Nov. 15, 2023. [Online]. Available: https://www.iriusrisk.com/
[30] *ThreatModeler*. Accessed: Nov. 15, 2023. [Online]. Available: https://threatmodeler.com/
[31] F. De Rosa, N. Maunero, P. Prinetto, F. Talentino, and M. Trussoni, "ThreMA: Ontology-based automated threat modeling for ICT infrastructures," *IEEE Access*, vol. 10, pp. 116514–116526, 2022, doi: 10.1109/ACCESS.2022.3219063.
[32] N. Maunero, F. De Rosa, and P. Prinetto, "Towards cybersecurity risk assessment automation: An ontological approach," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Nov. 2023, pp. 0628–0635, doi: 10.1109/dasc/picom/cbdcom/cy59711.2023.10361456.
[33] D. Papamartzivanos, S. A. Menesidou, P. Gouvas, and T. Giannetsos, "A perfect match: Converging and automating privacy and security impact assessment On-the-Fly," *Future Internet*, vol. 13, no. 2, p. 30, Jan. 2021, doi: 10.3390/fi13020030.
[34] G. Gonzalez-Granadillo, S. A. Menesidou, D. Papamartzivanos, R. Romeu, D. Navarro-Llobet, C. Okoh, S. Nifakos, C. Xenakis, and E. Panaousis, "Automated cyber and privacy risk management toolkit," *Sensors*, vol. 21, no. 16, p. 5493, Aug. 2021, doi: 10.3390/s21165493.
[35] M. Frydman, G. Ruiz, E. Heymann, E. César, and B. P. Miller, "Automating risk analysis of software design models," *Scientific World J.*, vol. 2014, pp. 1–12, Jun. 2014, doi: 10.1155/2014/805856.
[36] A. Brazhuk, "A set of semantic data flow diagrams and its security analysis based on ontologies and knowledge graphs," 2023, *arXiv:2303.11198*.
[37] V. Casola, A. D. Benedictis, C. Mazzocca, and R. Montanari, "Toward automated threat modeling of edge computing systems," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2021, pp. 135–140, doi: 10.1109/CSR51186.2021.9527937.
[38] A. T. Al Ghazo, M. Ibrahim, H. Ren, and R. Kumar, "A2G2 V: Automatic attack graph generation and visualization and its applications to computer and SCADA networks," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 50, no. 10, pp. 3488–3498, Oct. 2020, doi: 10.1109/TSMC.2019.2915940.
[39] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proc. IEEE Symp. Secur. Privacy*, May 2002, pp. 273–284, doi: 10.1109/SECPRI.2002.1004377.
[40] M. Surridge, A. Chakravarthy, M. Bashevoy, J. Wright, M. Hall-May, and R. Nossal, "SERSCIS-Ont: Evaluation of a formal metric model using airport collaborative decision making," *Int. J. Adv. Intell. Syst.*, vol. 4, no. 3, Apr. 2012, Art. no. 3.
[41] M. Surridge, B. Nasser, X. Chen, A. Chakravarthy, and P. Melas, "Runtime risk management in adaptive ICT systems," in *Proc. Int. Conf. Availability, Rel. Secur.*, Sep. 2013, pp. 102–110, doi: 10.1109/ARES.2013.20.

[42] M. Surridge, G. Correndo, K. Meacham, J. Papay, S. C. Phillips, S. Wiegand, and T. Wilkinson, "Trust modelling in 5G mobile networks," in *Proc. Workshop Secur. Softwarized Netw., Prospects Challenges*. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 14–19, doi: 10.1145/3229616.3229621.

[43] M. Surridge, K. Meacham, J. Papay, S. C. Phillips, J. B. Pickering, A. Shafiee, and T. Wilkinson, "Modelling compliance threats and security analysis of cross border health data exchange," in *Communications in Computer and Information Science*. Cham, Switzerland: Springer, 2019, pp. 180–189, doi: 10.1007/978-3-030-32213-7_14.

[44] B. Pickering, C. Boletsis, R. Halvorsrud, S. Phillips, and M. Surridge, "It's not my problem: How healthcare models relate to SME cybersecurity awareness," in *HCI for Cybersecurity, Privacy and Trust*. Cham, Switzerland: Springer, 2021, pp. 337–352.

[45] C. Boletsis, R. Halvorsrud, J. Pickering, S. Phillips, and M. Surridge, "Cybersecurity for SMEs: Introducing the human element into socio-technical cybersecurity risk assessment," in *Proc. 16th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2021, pp. 266–274. https://www.scitepress.org/Link.aspx?doi=10.5220/0010332902660274

[46] *German Federal Office for Security in Information Technology (BSI), IT Grundschutz Manual*, Federal Office Inf. Secur., Germany, 2004.

[47] *Spyderisk System Modeller*. Accessed: Nov. 15, 2023. [Online]. Available: https://purl.org/spyderisk/paper/2023-overview/software/system-modeller

[48] *Keycloak*. Accessed: Nov. 15, 2023. [Online]. Available: https://www.keycloak.org/

[49] *Spyderisk System Modeller Documentation*. Accessed: Nov. 15, 2023. [Online]. Available: https://purl.org/spyderisk/paper/2023-overview/documentation/system-modeller

[50] *Spyderisk Network Knowledgebase*. Accessed: Nov. 15, 2023. [Online]. Available: https://purl.org/spyderisk/paper/2023-overview/software/knowledgebase

[51] *Spyderisk Network Knowledgebase Documentation*. Accessed: Nov. 15, 2023. [Online]. Available: https://purl.org/spyderisk/paper/2023-overview/documentation/knowledgebase

[52] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An integrative model of organizational trust," *Acad. Manage. Rev.*, vol. 20, no. 3, p. 709, Jul. 1995, doi: 10.2307/258792.

[53] *Alert: Apache Log4j Vulnerabilities*. Accessed: Nov. 15, 2023. [Online]. Available: https://www.ncsc.gov.uk/news/apache-log4j-vulnerability

[54] *Common Vulnerability Scoring System*. Accessed: Nov. 15, 2023. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss

[55] *Risk Management Guidelines*, ISO/IEC 31000:2018, International Organization for Standardization (ISO), Feb. 2018

[56] *ICS CP/PE (Cyber-to-Physical or Process Effects) Case Study Paper German Steel Mill Cyber Attack*. Accessed: Nov. 15, 2023. [Online]. Available: https://assets.contentstack.io/v3/assets/blt36c2e63521272fdc/blt5bd1acefa6ad7c17/6323756c1ad88e716559ed66/ICS-UseCase2-ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf

[57] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100219, doi: 10.1016/j.cosrev.2019.100219.

[58] S. C. Phillips. *Spyderisk Attack Graph Tool*. Accessed: Nov. 15, 2023. [Online]. Available: https://purl.org/spyderisk/paper/2023-overview/software/plot-attack-graph

STEPHEN C. PHILLIPS received the double honours B.Sc. degree in chemistry and mathematics and the Ph.D. degree in chemical physics from the University of Southampton, U.K., in 1997 and 2001, respectively. He is currently a Principal Enterprise Fellow of applied computer science with the IT Innovation Centre. He has authored over 40 peer-reviewed publications. His research interests include risk management in complex systems and information modeling and presentation.

STEVE TAYLOR received the Ph.D. degree in computer science (artificial intelligence) from the University of Greenwich, in 1997. He is currently a Principal Research Engineer with the University of Southampton, U.K., with over 20 years of experience in collaborative projects. He has published over 40 academic articles. His research interests include risk management in complex systems bringing aspects, such as cybersecurity, privacy, and AI trustworthiness together. He is a member of the Networked European Software and Services Initiative (NESSI) Steering Committee.

MICHAEL BONIFACE is currently a Professorial Fellow of information technology and the Director of the IT Innovation Centre, School of Electronics and Computer Science, University of Southampton. His research interests include the advancing mechanisms of data governance, artificial intelligence, and interaction for health and well-being tackling challenges across public health, chronic disease management, and integrated care. He is a fellow of the Institute of Engineering and Technology, with over 20 years of experience in applied research in federated systems management, interactive systems, and data science across healthcare, creative industries, cloud computing, and telecommunications.

STEFANO MODAFFERI received the Ph.D. degree in information engineering from Politecnico di Milano, in 2007. He is currently a Principal Enterprise and a Research Fellow with the University of Southampton, U.K., with over 20 years of experience in collaborative projects. He is also the Leader of the Big Data Value Association (BDVA) task force on business models. He has published over 30 academic articles. His research interests include information modeling, business processes, and socio-technical systems.

MIKE SURRIDGE received the Ph.D. degree in theoretical physics in Southampton, in 1986. He was with the University of Amsterdam before returning to Southampton to start a career in applied computer science. His first focus was parallel high-performance computing architectures and systems, which led to work on wide-area network computing and thence to an interest in cyber security in distributed and multi-stakeholder systems. He coordinated the EU SERSCIS Project, which developed cyber security models and automated threat identification and led ultimately to Spyderisk. His research interests include cyber security, risk management, and trust in dynamic multi-stakeholder systems, including clouds, 5G, and the IoT applications, with a special focus on healthcare. He was the Co-Founder and the Chair of the EC/NESSI Software and Services Trust and Security Working Group, for many years, and a member of the EC Future Internet Architecture Board and the EC 5G Vision and Societal Challenges Working Group.