Behavioural Profiling in Cyber-Social Systems

Perno, Jason; Probst, Christian W.

Published in:
HAS 2017: Human Aspects of Information Security, Privacy and Trust

Link to article, DOI:
10.1007/978-3-319-58460-7_35

Publication date:
2017

Document Version
Peer reviewed version

Citation (APA):
Perno, J., & Probst, C. W. (2017). Behavioural Profiling in Cyber-Social Systems. In T. T. (Ed.), *HAS 2017: Human Aspects of Information Security, Privacy and Trust* (Vol. 10292, pp. 507-517). Springer. https://doi.org/10.1007/978-3-319-58460-7_35

Behavioural Profiling in Cyber-Social Systems

Conference Paper · May 2017

DOI: 10.1007/978-3-319-58460-7_35

Jason Perno
Utica College

Behavioural Profiling in Cyber-Social Systems

Jason Perno[1] and Christian W. Probst[2]

[1] Utica College, Utica, NY, USA
jwperno@utica.edu
[2] Technical University of Denmark, Kongens Lyngby, Denmark
cwpr@dtu.dk

Abstract. Computer systems have evolved from standalone systems, over networked systems, to cyber-physical systems. In all stages, human operators have been essential for the functioning of the system and for understanding system messages. Recent trends make human actors an even more central part of computer systems, resulting in what we call "cyber-social systems". In cyber-social systems, human actors and their interaction with a system are essential for the state of the system and its functioning. Both the system's operation and the human's operating it are based on an assumption of each other's behaviour. Consequently, an assessment of the state of a system must take the human actors and these interactions into account. However, human behaviour is difficult to model at best. While socio-technical system models promise the inclusion of human actors into a basis for system assessment, they lack the modelling mechanisms for human behaviour. Existing behavioural models, on the other side, mostly aim at explaining actions after an event. In this paper we discuss, how behavioural models can be used to profile actor behaviour either online or in simulations to understand the potential motivation and to test hypotheses.

1 Introduction

In many computer systems, human actors and their interactions with the system are essential for the state of the system and its functioning. Consequently, an assessment of the state of a system must take the human actors and these interactions into account. This need results from computer systems evolving from standalone systems, over networked systems, to cyber-physical systems. In all stages, human operators have been essential for the functioning of the system and for understanding and interpreting system messages. These recent trends make human actors an even more central part of computer systems, resulting in what we call "cyber-social systems".[3]

Explaining human behaviour is – in principle – easy: all we need is a concise model of human behaviour that integrates dependencies on surroundings, a precise surveillance system, and an evaluation system to draw conclusions from

[3] As discussed in Section 3, we consider cyber-social systems at the system level, opposed to Stanford University's Cyber-Social Systems [24].

input. Of course, such a model and its components are neither "easy" to realise, nor desirable, and many aspects depend on legal regulations. As a result, human behaviour is difficult to model at best, be it at the societal or the individual level. While socio-technical system models promise the inclusion of human actors into a basis for system assessment, they lack the modelling mechanisms for human behaviour. Existing behavioural models, on the other side, mostly aim at explaining actions after an event, for example, to help analysts understand and explain, what has happened.

In this paper we discuss, how cyber-social systems can be represented as a combination of socio-technical systems and behavioural models, and how they can be used to profile actor behaviour. This profiling can be performed online or in simulations to understand the potential motivation.

The rest of this article is structured as follows. The next section introduces some background material about socio-technical systems, attack representations, and behavioural models, followed by a discussion of cyber-social systems and behavioural trees, which are our behavioural model, in Section 3. In Section 4 we discuss, how these systems can be used to perform behavioural profiling. Finally, Section 5 concludes this article, and discusses future research directions.

2 Background

The work presented in this paper builds upon findings and developments in three main areas: socio-technical system models, attack representations, and models for explaining insider threats.

2.1 System Models

Recently, several system models have been introduced that inspire our work. ExASyM [17,19], Portunes [3] and ANKH [15] models follow similar ideas - the modelling of infrastructure and data, and analysing the modelled organisation for possible threads. The semantics of both ExASyM and Portunes is formalised using a variant of the Klaim family of process calculi [13]. However, Portunes supports mobility of nodes, instead of processes, and represents the social domain by low-level policies that describe the trust relation between people. The latter is used to represent social engineering. In contrast to the above two models, ANKH has a flat structure and the formal representation is a hyper-graph where the hyper-edges represent containment. The modelling formalism heavily depends on policies, which must be well defined in order to avoid unrealistic cases.

Pieters *et al.* consider policy alignment to address different levels of abstraction of socio-technical systems [16], where policies are interpreted as first-order logical theories containing all sequences of actions (the behaviours) and expressing the policy as a "distinguished" prefix-closed predicate in these theories. In contrast to their use of refinement for policies we use the security refinement paradox, *i.e.*, security is *not* generally preserved by refinement, in order to discover attacks.

2.2 Attack Representations

Attack trees [21,22] specify an attacker's main goal (or a main security threat) as the root of a tree; this goal is then disjunctively or conjunctively refined into sub-goals. This refinement is repeated recursively, until the reached sub-goals represent basic actions that correspond to atomic components. Disjunctive refinements represent alternative ways of how a goal can be achieved, whereas conjunctive refinements depict different steps an attacker needs to take in order to achieve a goal [20,10]. Techniques for the automated generation of attack graphs consider computer networks only [14,23], or general policies [7,8].

2.3 Behavioural Models

Legg *et al.* [12] address the complex and dynamic problem posed by insiders against organisations. Their three-tier model incorporates a tier representing the real world, a tier representing measurements or observables, and a tier representing hypotheses. The goal of the model is to support the analyst in detecting potential insider threats. On the real world tier, a large set of elements exist that correlate with insider threats, for example, activities, physical behaviour, and psychological mindset. Since most of these elements can not be observed directly, the analyst and the hypothesis tier must rely on measurements provided by the middle tier of the model. The confidence in observations made by elements in this layer depends on how directly they are able to observe the real world: the technical ones probably have high confidence in the associated values, whereas the psychological and behavioural ones can only be observed indirectly through a small set of indicators, and consequently provide a much lower level of confidence.

System dynamics models represent complex systems in order to understand their nonlinear behaviour. Models contain flow, feedback loops, and time delays, and can model complex interaction between different actors. System dynamics has been used to model and analyze the dynamic nature of the insider threat problem [2,5], especially with focus on modelling human behaviour.

3 Cyber-Social Systems

Cyber-social systems result from cyber-physical systems by integrating human actors into the system and the reasoning about it. Computer systems have evolved from standalone systems, over networked systems, to cyber-physical systems. In all stages, human operators have been essential for the functioning of the system and for understanding system messages. Recent trends make human actors an even more central part of computer systems, resulting in what we call "cyber-social systems". As mentioned above, we consider cyber-social systems at the level of the actual system and actors interacting with it. This complements the work by, *e.g.*, the Stanford Cyber Initiative, which investigates how cyber-technologies interact with existing social systems to understand cyber-social systems [24].

In cyber-social systems, human actors and their interaction with a system are essential for the state of the system and its functioning. On a societal level, these may be influenced by markets, political systems, and policies. We are interested in instances of such systems, and the processes at the system and actor level. Decisions and behaviour at this level may be influenced by more abstract concepts, but that is currently beyond the scope of our work.

To reason about cyber-social systems, we represent them as a combination of a *socio-technical system model*, which represents the context of the system being analysed, and a *behavioural model* for the human actors in that system. Cyber-social systems thus enhance socio-technical systems with components for the actors' behaviour similar to approaches for externalizing behaviour in system models [6]. The models for systems and behaviour are parameters of a cyber-social system. Based on the application and the goal, these components can be chosen as needed. In the remainder of this section, we briefly present candidates for each of these.

It is important to note that the techniques described in this paper are independent from the underlying models, similar to earlier developments by Ivanova *et al.* [6]. Figure 1 shows their system model, which extends actors with individual behaviour. The main contribution of the current work is the development of behavioural trees and their embedding in cyber-social systems.

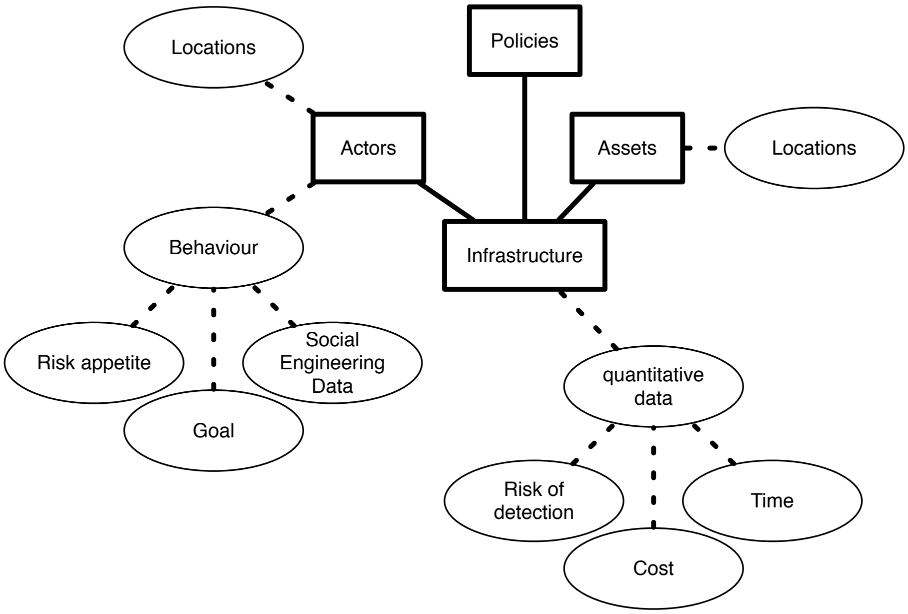

Fig. 1. A system model structure with explicit behaviour and quantitative data [6]. Together, these components form a cyber-social system, which is parameterised with the underlying system model and behavioural model.

3.1 The Socio-Technical System Model

The socio-technical system model is closely related to existing models [17,26]. It is based on a process calculus that represents the three layers of socio-technical models – the physical, the virtual, and the social layer – as parts of a graph-based representation with processes for describing functionality at the virtual and social layer.

The socio-technical system model represents the system infrastructure as nodes in a directed graph [17], representing rooms, access control points, and similar locations. Processes also represent actors and can possess data and items that are relevant in the modelled scenario. Elements in the model can be annotated with values, *e.g.*, the likelihood of being lost. Data and items can be attached to locations or processes; those attached to processes move around with that actor. Processes perform actions on locations, including physical locations or other processes. These actions are restricted by policies, which consist of required credentials and enabled actions. Credentials represent the data or assets an actor needs to provide in order to enable the actions in a policy, and the enabled actions describe the gained rights by doing so [25]. Policies are used both for access control and for organisational policies.

3.2 Behavioural Trees

Behavioural trees capture two components: they structure larger behavioural patterns into sub-actions, and they include dependencies that represent how the actor's disposition towards certain actions changes based on events. A behavioural tree extends the embedding of behaviour in system models [6] by encoding an analyst's experience and strategy. In structure, behavioural trees are similar to attack trees and attack templates [27].

Attack trees as described in Section 2.2 are a very flexible and loosely defined tool to represent steps in possible attacks. Their success is to a large extent due to their loose definition. Behavioural trees follow a similar strategy by offering a simple structure for defining human behaviour and enabling factors. Just like for attack trees, however, extensions will be needed to model, *e.g.*, prohibiting events, which could be represented similar to attack-defense trees [9].

Behavioural trees contain similar nodes as attack trees [1]:

- Disjunctive nodes represent options of which one must be present,
- Conjunctive nodes represent options, which all must be present, but may appear in arbitrary order,
- Temporal disjunctive nodes represent options that are tried from left to right, and of which one must be present, and
- Temporal conjunctive nodes represent options that must occur in that order from left to right.

In attack trees, leafs usually describe basic actions, and inner nodes are mostly used to label the "meaning" of the sub-tree rooted in these nodes. In behavioural trees, both leafs and inner nodes describe actions, events, and decisions taken by an actor. Both leafs and inner nodes may also be part of one or more system dynamics overlays, which describe, how the events and actions of the actor or the environment influence the actor's behaviour and disposition towards certain behaviour.

Figure 2 shows an example for a behavioural tree that represents two possible actions: stealing an asset and going to a competitor, which both are influenced by parts of the system dynamics overlay [2]. The node with the double frame represents a temporal conjunctive node: in order to steal an asset, the actor must first have the desire to steal, and then get the chance. The dashed lines connect nodes in the behavioural tree, the solid arrows connect nodes in the system dynamics model, and describe direct or opposite changes in the value

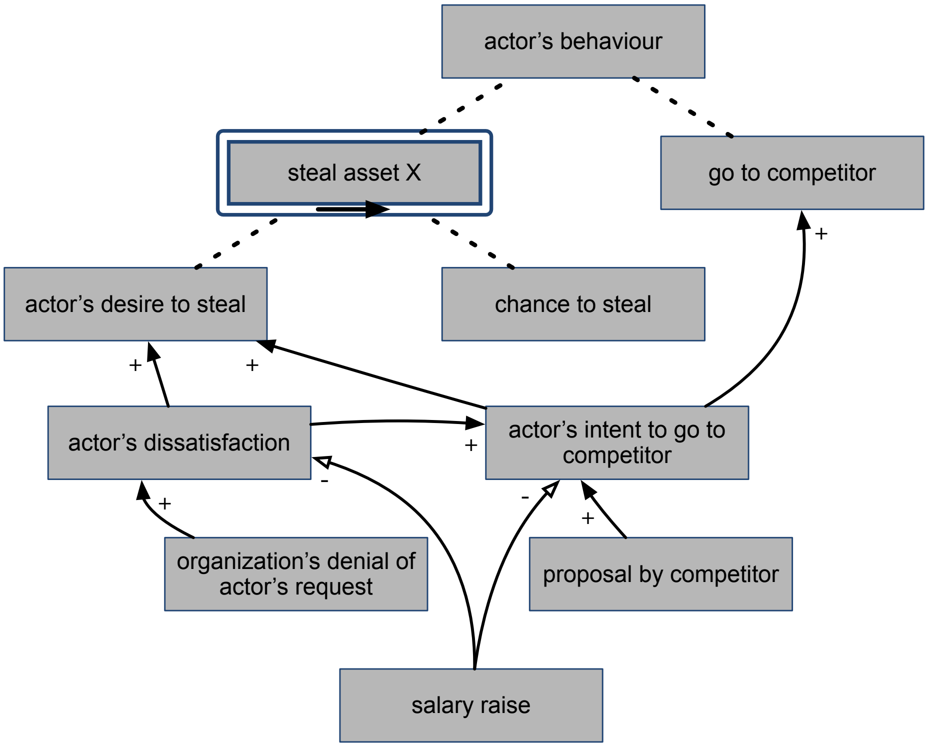

Fig. 2. An example for a behavioural tree that represents two possible actions: stealing an asset and going to a competitor, which both are influenced by parts of the system dynamics overlay [2]. The dashed lines connect nodes in the behavioural tree, the solid arrows connect nodes in the system dynamics model, and describe direct or opposite changes in the value of the target node based on changes at the source node. The node with the double frame represents a temporal conjunctive node.

of the target node based on changes at the source node. Nodes may be part of either the behavioural tree, the system dynamics model, or both.

It is noteworthy that the changes induced to one node in the system dynamics model based on changes at another node do not need to be constant, but can vary based on time, the current value, or other factors influenced by the overall state of the system. Also, the threshold at which the action at a node is enabled is usually not binary, but changes continuously.

Behavioural trees need not and cannot be complete, since it is impossible to predict all aspects of human behaviour, its dependencies on inside and outside events, and the relevant events influencing behaviour. However, some of the quantitative measures can be initiated based on personality testing and lifestyle polygraphs, as are often performed as part of job interviews.

Furthermore, parts of the behavioural trees are similar for all actors, probably with different factors, and thus can be shared across populations. Furthermore, behavioural trees can be extended during the behavioural profiling with newly observed actions. While these actions initially do not have quantitative properties, they can be initialised with heuristic values to feed the analysis and simulation, which in turn will refine the initial values to more sensible ones.

4 Behavioural Profiling

Legg *et al.* [12] discuss two applications of their model: *bottom-up* and *top-down*, where the top tier is the hypothesis, as described above, and the bottom layer is the real world.

Based on direct observations (measurements) of the real world, bottom-up reasoning begins with making indirect observations, for example, based on statistical profiles for each individual, and profiles capturing their traits and behaviours. These indirect observations then feed into hypotheses, which are the building blocks for the analyst to formulate more complex hypotheses, triggering alerts, for example, if the collected measurements of an indicator exceed the expected values [12].

On the other hand, top-down reasoning begins from a concrete concern, for example based on input from a whistle blower or from a trigger-based alert from the bottom-up analysis. In this case, the analyst will formulate a hypothesis, and the model would attempt to “fulfil” this assumption given possible observations from the measurement tier.

In this section we describe how behavioural trees can be applied to model this workflow in automatic analyses in two different cases: the backward-looking analysis explaining observed events, and the forward-looking analysis predicting future events. Finally, we discuss methods for refining the values in behavioural trees by combining these two analyses, and how to refine values through statistical model checking.

All analyses described in the following can be applied in general on a population of actors, in which case they result in conditions that potential actors must fulfil, or on a specific actor, in which case they confirm or invalidate a hypothesis with respect to that actor.

4.1 Explaining Past Behaviour

Explaining past behaviour is equivalent to the top-down reasoning described above. In this setting, the analyst has a concern and tries to understand, what happened and how, and which observations to look out for.

In this application scenario, behavioural trees are traversed top down, from behaviour to actions. At transitions to the system dynamics model, this part is explored backwards. This exploration provides the analysis with possible reasons, why a certain action was performed, and with events that can be expected to have occurred. At transitions to the behavioural tree, the top down exploration continues from the nodes that are triggered by the system dynamics model.

In the tree in Figure 2, for example, if the theft of an asset has been observed, the analysis will identify the desire to steal and the chance to steal as necessary pre-conditions. The desire is influenced positively by the actor's dissatisfaction and an intent to go to a competitor, but negatively by a possible salary raise.

4.2 Predicting Future Behaviour

Predicting future behaviour is equivalent to bottom-up reasoning, which builds hypotheses that the analyst can use to setup surveillance mechanisms.

As before, also in this application scenario we traverse behavioural trees top down, but with a different goal: now we aim at identifying the actions and events to look out for, and possibly also actors who are likely to perform these actions or trigger these events. The system dynamics model is now explored *in both* directions: backward to identify possible reasons and triggering events for actions, and forward to identify possible followup events and actions to lookout for. The backward events must be handled with care, since some of them are likely to have occurred before the analysis started; this must be accounted for in the reasoning.

In the behavioural tree presented in Figure 2, for example, the type of asset defines the applicable actions for obtaining it, *e.g.*, logging in remotely, the use of flash drives, or emails. The analysis may use this information to suggest where to set up surveillance mechanisms to alert a human operator or online surveillance mechanisms [18]. Especially the notification of the organisation is promising, since many events influencing behaviour are difficult to formalise and measure, *e.g.*, that an actor might be on the verge of leaving the organisation.

4.3 Combining Past and Future

Executing either of the two analyses described above after the other, as well as iterations alternating between the two phases, is beneficial to understanding and profiling behaviour:

- Results of an analysis of past behaviour provide the analysis with input to make better predictions of the future behaviour, and similarly,
- Results of an analysis of future behaviour, that is which events and actions to look out for, guide the analysis of past events towards those parts of the behavioural tree that may influence this future behaviour.

Typically, we expect several changes of direction in such an analysis: based on results for the future, the analysis of the past is refined, and vice versa, providing more input for explaining future events, or extending the possible set of future events and trying to identify more supporting data from the past.

Another dimension are combinations of behavioural trees for different actors, which extend the search space for possible motivations. Also here, the observed past events or identified future events help the analysis to limit exploration to those actors that potentially may perform the actions or perform relevant actions that may influence the behaviour of an actor under scrutiny.

4.4 Refining Values

As mentioned before, precise models of human behaviour cannot be built, and consequently behavioural trees are incomplete and the values and factors in the trees will not be precise. However, behavioural trees describe possible behaviour of actors, and as such can be used together with the socio-technical system model for simulations of this behaviour in statistical model checking [11].

Simulations through statistical model checking provide the behavioural profiling of cyber-social systems, and they provide the means to verify computed and observed likelihoods of actions and events. The simulation applies the analysis results for past behaviour to future behaviour, by simulating the behaviour of actors. This simulated behaviour can be predetermined, randomised, or follow more involved strategies [6].

5 Conclusion and Future Work

In this article, we have described how to perform behavioural profiling for cyber-social systems, which combine socio-technical systems and behavioural trees. Cyber-social systems are the next step in integration of computer systems by making human actors an even more central part of these systems, which have evolved from standalone systems, over networked systems, to cyber-physical systems. In all stages, human operators have been essential for the functioning of the system and for understanding system messages. Now, human actors and their interaction with a system are essential for the state of the system and its functioning. Both the system's operation and the human's operating it are based on an assumption of each other's behaviour. Consequently, an assessment of the state of a system must take the human actors and these interactions into account.

Behavioural profiling based on a combination of behavioural trees and socio-technical system models promises the simulation of analysts' workflows [12], and the verification of results using statistical model checking. Behavioural trees are by definition incomplete, but can be extended during the analysis with newly observed actions. While these actions initially do not have quantitative properties, they can be initialised with heuristic values to feed the analysis and simulation, which will refine them to more sensible values.

We are currently working on a theory for cyber-social systems and their application to behavioural profiling. This involves refining behavioural trees and relevant properties, as well as heuristics for choosing new actions and events to add to the tree. Especially psycho-analytical based profiling models, as well as studying personality traits, are interesting to benchmark and refine the profiling in cyber-social systems. In automated approaches it is in general impossible to observe the context and circumstances that dictate and predict criminal behaviour, let alone to understand them. However, there are many similarities between Weber's sociological explanation of the social situation and the collective explanandum [4], and abstraction and realisation applied in the computation of fix points in formal methods.

References

1. Aslanyan, Z., Nielson, F., Parker, D.: Quantitative verification and synthesis of attack-defence scenarios. In: Proceedings of the 29th Computer Security Foundations Symposium (CSF) (2016)
2. Cappelli, D.M., Moore, A.P., Trzeciak, R.F.: The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes. Addison-Wesley Professional (2012)
3. Dimkov, T.: Alignment of Organizational Security Policies – Theory and Practice. University of Twente (2012), `http://eprints.eemcs.utwente.nl/21578/`
4. Esser, H.: Soziologie Allgemeine Grundlagen. Campus (1993)
5. Gonzalez, J.J., Sawicka, A.: A framework for human factors in information security. In: Proceedings of the WSEAS International Conference on Information Security (2002)
6. Ivanova, M.G., Probst, C.W., Hansen, R.R., Kammüller, F.: Externalizing behaviour for analysing system models. In: 5th International Workshop on Managing Insider Security Threats (MIST 2013) (2013)
7. Kammüller, F., Probst, C.W.: Invalidating policies using structural information. In: 2nd International IEEE Workshop on Research on Insider Threats (WRIT'13). IEEE (2013), co-located with IEEE CS Security and Privacy 2013
8. Kammüller, F., Probst, C.W.: Combining generated data models with formal invalidation for insider threat analysis. In: 3rd International IEEE Workshop on Research on Insider Threats (WRIT'14). IEEE (2014), co-located with IEEE CS Security and Privacy 2014
9. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Attack–Defense Trees. Journal of Logic and Computation (2012)
10. Kordy, B., Piètre-Cambacédès, L., Schweitzer, P.: Dag-based attack and defense modeling: Don't miss the forest for the attack trees. Computer Science Review 13-14, 1 – 38 (2014), `http://www.sciencedirect.com/science/article/pii/S1574013714000100`

11. Legay, A., Delahaye, B., Bensalem, S.: Statistical model checking: An overview. In: Proceedings of the First International Conference on Runtime Verification (2010)
12. Legg, P., Moffat, N., Nurse, J.R., Happa, J., Agrafiotis, I., Goldsmith, M., Creese, S.: Towards a conceptual model and reasoning structure for insider threat detection. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 4(4), 20–37 (2013)
13. de Nicola, R., Ferrari, G.L., Pugliese, R.: KLAIM: A kernel language for agents interaction and mobility. IEEE Trans. Softw. Eng. 24(5), 315–330 (May 1998), `http://dx.doi.org/10.1109/32.685256`
14. Phillips, C., Swiler, L.P.: A graph-based system for network-vulnerability analysis. In: Proceedings of the 1998 workshop on New security paradigms NSPW 98. vol. pages, pp. 71–79 (1998)
15. Pieters, W.: Representing humans in system security models: An actor-network approach. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 2(1), 75–92 (2011)
16. Pieters, W., Dimkov, T., Pavlovic, D.: Security policy alignment: A formal approach. IEEE Systems Journal 7(2), 275–287 (2013)
17. Probst, C.W., Hansen, R.R.: An extensible analysable system model. Information Security Technical Report 13(4), 235–246 (Nov 2008)
18. Probst, C.W., Hansen, R.R.: Analysing access control specifications. 2009 Fourth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering pp. 5,6 (2009)
19. Probst, C.W., Hansen, R.R., Nielson, F.: Where can an insider attack? In: Proceedings of the 4th international conference on Formal aspects in security and trust. pp. 127–142. FAST'06, Springer (2007)
20. Qin, X., Lee, W.: Attack plan recognition and prediction using causal networks. In: 20th Annual Computer Security Applications Conference. pp. 370–379 (Dec 2004)
21. Salter, C., Saydjari, O.S., Schneier, B., Wallner, J.: Toward a secure system engineering methodology. In: Proceedings of the 1998 Workshop on New Security Paradigms (NSPW '98). pp. 2–10. Charlottesville, Virginia, United States (Sep 1998)
22. Schneier, B.: Attack Trees: Modeling Security Threats. Dr. Dobb's Journal of Software Tools 24(12), 21–29 (1999), `http://www.ddj.com/security/184414879`
23. Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.M.: Automated generation and analysis of attack graphs. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy (S&P'02). vol. 129, pp. 273–284 (2002)
24. Stanford Cyber Intiative: Understanding "cyber-social systems". Available online at `https://cyber.stanford.edu/sites/default/files/stanford_cyber_initiative_.pdf`, last accessed March 10, 2017.
25. The TRE$_S$PASS Project: Deliverable D1.2.2: The final TRE$_S$PASS policy-specification language. Available online at `https://www.trespass-project.eu/node/222` (2015), last accessed March 10, 2017.
26. The TRE$_S$PASS Project: Deliverable D1.3.4: The TRE$_S$PASS socio-technical security model and specification languages. Available online at `https://www.trespass-project.eu/node/302` (2016), last accessed March 10, 2017.
27. The TRE$_S$PASS Project: Deliverable D5.4.2: The integrated TRE$_S$PASS process. Available online at `https://www.trespass-project.eu/node/315` (2016), last accessed March 10, 2017.