Implementation of a Generic ICT Risk Model using Graph Databases

Stefan Schiebeck, Martin Latzenhofer,
Brigitte Palensky, Stefan Schauer
Digital Safety & Security
Austrian Institute of Technology
Vienna, Austria
e-mail: {stefan.schiebeck.fl | martin.latzenhofer |
brigitte.palensky | stefan.schauer}@ait.ac.at

Gerald Quirchmayr
Research Group Multimedia Information Systems
Faculty of Computer Science
University of Vienna
Vienna, Austria
e-mail: gerald.quirchmayr@univie.ac.at

Thomas Benesch
Research & Development
s-benesch
Vienna Austria
e-mail: thom@s-benesch.com

Johannes Göllner, Christian Meurers, Ingo Mayr
Department of Central Documentation & Information
National Defence Academy of the Austrian Federal
Ministry of Defence and Sports, Vienna, Austria
e-mail: {johannes.goellner | christian.meurers |
ingo.mayr}@bmlvs.gv.at

***Abstract*— Advanced Persistent Threats (APTs) impose an increasing threat on today's information and communication technology (ICT) infrastructure. These highly-sophisticated attacks overcome the typical perimeter protection mechanisms of an organization and generate a large amount of damage. Based on a practical use case of a real-life APT lifecycle, this paper shows how APTs can be tackled using a generic ICT risk analysis framework. Further, it provides details for the implementation of this risk analysis framework using graph databases. The major benefits of this graph database approach, i.e., the simple representation of the interconnected risk model as a graph and the availability of efficient traversals over complex sections of the graph, are illustrated giving several examples.**

Keywords- risk management; APT; ICT security; graph databases; interconnected risk model.

I. INTRODUCTION

Although internal attacks can be seen as today's biggest threat on information security [1], in practice, information security officers still put great emphasis on perimeter control. The internal area of a company's ICT network, e.g., the demilitarized zone (DMZ) or the intranet, is secured based on standard technical guidelines demanding, e.g., the logical separation of a network into subnetworks according to specific security requirements [2]. Nevertheless, the effort invested in monitoring the internal network is moderate. Intrusion detection and prevention systems are cost and time consuming and require a large amount of administration. Recent attack strategies like Advanced Persistent Threats (APTs) take advantage of this lack of internal control.

The term APT summarizes a family of highly sophisticated attacks on an ICT network or infrastructure. Usually, an APT runs over an extended period of time with the objective to steal data and maintain presence indefinitely without being detected. A continuous access allows collecting new data as it emerges, extending the achieved foothold over time, and using the site as a jumping point for the attack on other facilities. The adversary – usually a group of people – has a large amount of resources at hand and applies the whole range of digital, physical and social attack vectors to gain access to a system. The attack is specifically designed for a particular victim, i.e., a company or an organization, such that common security measures can be circumvented effectively. Thus, the adversary stays undetected over a long period of time. One particular technique recurrently used in APTs is social engineering, which exploits the human factor as a major vulnerability of an ICT system. Potential countermeasures, like increasing the staff's awareness concerning ICT security threats via training courses, are not very common. According to a Ponemon study [3], about 52% of the interviewed organizations do not offer respective training courses for their employees.

In the course of the last decade, APTs became one of the most significant kinds of threats on information security, causing a great number of security incidents all over the world [4]. Besides the most prominent APT attack, the application of the malware Stuxnet in an Iranian nuclear power plant, a number of other APT attacks have become known, e.g., Operation Aurora, Shady Rat, Red October or MiniDuke [5][6][7]. As it is shown in the Mandiant Report [4], some adversaries even have a close connection to governmental organizations. The former director of the US cyber command, General Keith Alexander, referred to the currently occurring industrial espionage and theft of intellectual property as "the greatest transfer of wealth in history" [8]. In Europe, the disclosures of Edward Snowden [9] have drawn great attention to this issue. Based on current numbers from cyber-crime reports, which show the growing

 ISBN: 978-1-61208-427-5

amount of damage [10][11], it is distressing how poorly evolved today's countermeasures seem to be.

This paper focuses on the implementation of a generic ICT risk model that can deal with the described issues. The implementation is based on graph databases and social network analysis concepts to provide a perspective that can focus on a specific aspect (node) and its influences (relationships). From a technological perspective, the advantages of the chosen approach are demonstrated, in particular concerning risk aggregation. Therefore, different types of assets, e.g., organizational aspects like processes and personnel, ICT components like IT systems and logical networks, and physical infrastructure objects, serve as examples of assets that are attacked in fictitious, but realistic ways. In detail, after a short overview of related work on graph-based models in Section II, Section III sketches the different steps of an APT attack for a fictitious scenario to illustrate the basic principles of this family of threats. Section IV shows the generic meta-risk-model depicted as a graph model and shortly discusses the pros and cons of an implementation via graph databases vs. relational databases. Section V provides a detailed description of how the generic risk model was implemented using a graph database. Finally, Section VI summarizes the results.

II. RELATED WORK

In general, graph-based models are used to capture relations among system entities at various abstraction levels. In [12], Chartis Research advises the introduction of graph analytics (based on graph databases) to the risk management activities of financial institutions so that they can discover so far unknown risks by revealing interconnected risk patterns. In [13], graph-based representations are applied in the area of risk management for critical infrastructures (CI). Bayesian Networks are used to learn (or simply estimate) CI service risks and their interdependencies. Additionally, a risk prediction is introduced and a case study to validate the model is carried out. However, some of the model's features, like risk prediction and the handling of cyclic dependencies, could not be verified because they simply did not occur during the run-time of the case study. The goal of the approach in [13] is to identify an abstract set of variables and their dependencies based on system measurements. Nevertheless, graph databases have not been used therein. In this paper, we introduce the explicit usage of graph databases (see Section IV for a discussion) to implement an already existing risk scheme retrieved from the IT-Grundschutz framework [14]. With this approach, cascading risks can be represented in a straight-forward way that allows us to run easily through a typical APT attack scenario. The underlying model and functional assessment concept presented in this paper, excluding the usage of a graph database for data manipulation, has been demonstrated in [15], although with the use of a relational database.

III. APT SCENARIO

In [4], the US security company Mandiant describes the typical lifecycle of an APT attack based on an analysis of how a Chinese cyber espionage group infiltrated several companies in the US and worldwide. In the following, the different steps in this APT lifecycle are briefly sketched to give an overview on the basic operations of an APT attack (cf. Figure 1). To provide a better illustration of the scenario, a fictional research facility, *Biomedical Research*, is used. It consists of four research laboratories with increasing degrees of security requirements (*Biosafety Level 1-4*) located in physically separated buildings. Additionally, the research facility runs two data centers, one located in the research building itself, and the other, which works as a backup, located at a distant administrative building. The information most valuable for an attacker is assumed to be hosted in Research Laboratory FL4, which is the one with the highest security level, or in one of the data centers. Based on this setting, a generalized APT attack can be outlined in eight steps.

As a first step, *Initial Recon*, the adversary tries to gain access to the organization's ICT infrastructure. Since the terminals in Research Laboratory FL1 are the only ones having full connection to the internet, a user in FL1 would be a primary target for a spear phishing attack (cf. (1) in Figure 1) in order to place a remote backdoor on either of these terminals. A potential user to be attacked can be identified for example using social engineering. In the second step, *Initial Compromise*, a user in FL1 receives a spear phishing mail and opens the infected attached file (e.g., a ZIP-file). During the execution of the ZIP-file, a basic backdoor (beachhead backdoor, cf. (2) in Figure 1) is installed on the terminal W1. Through this backdoor, a connection to the adversary's command and control server is established. In a third step, *Establish Foothold*, this initial connection is used to install a standard backdoor on the compromised terminal, giving the adversary an increased set of possibilities. Hence, the adversary is able to gain foothold at the application server S1 in FL1 (cf. (4) in Figure 1).

The following four steps (steps 4 to 7) are usually performed more than once, until the adversary acquires the desired information. In step 4, *Escalate Privileges*, the adversary gathers information on valid combinations of user names and passwords inside the internal networks. The attacker also gains additional information about the internal network structure (step 5 – *Internal Recon* – cf. (5) in Figure 1), potentially including internal authentication information. In the following step 6, *Move Laterally*, the adversary infiltrates the local data center as well as the backup data center to locate the valuable information. This is achieved using a vulnerability scan on the file servers S7.1 and S7.2 and an appropriate exploit allowing the compromise of both identically configured systems (cf. (6) in Figure 1). As a final step of this recurrent loop, *Maintain Presence*, all tracks are covered up and the adversary silently stays in the victim's system with an extended foothold (cf. (7) in Figure 1).

The final step, *Complete Mission*, starts when all the target information is collected. Covert channels are established (e.g., using cryptography/steganography) to extract the sensitive information from the file servers (cf. (8) in Figure 1). Afterwards, all traces of the attack are erased.

 ISBN: 978-1-61208-427-5

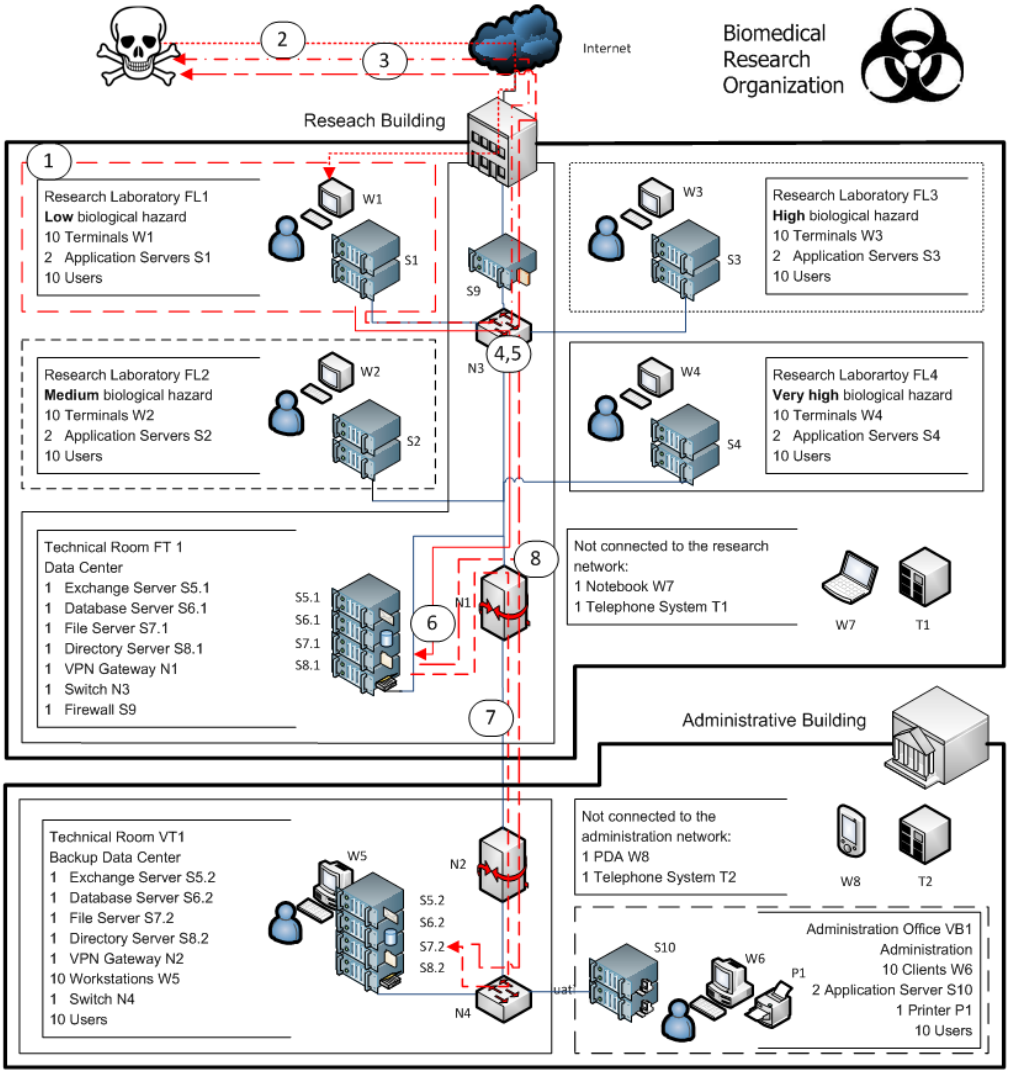

Figure 1 . Sample scenario of a typical APT attack.

IV. GRAPH-BASED META-RISK-MODEL

The meta-model used for risk management depicted in Figure 2 shows the risk graph and its semantic relations. Derived in a combined bottom-up/top-down way, it subsumes all the typical components of common risk management models, tools, processes, and control logic in a generic meta-model (cf. [16] for a more detailed description of the model).

In this work, as architectural background for the implementation of the model the graph database Neo4j [17] is used instead of a relational database. Graph databases provide the advantage of being able to perform near-real-time traversals and aggregations, efficient topology analyses, and the optimal finding of node neighbors [18]. The retrieval time of graph databases is usually significantly less than that of relational databases [19][20]. Moreover, the graph-based implementation ensures more flexibility for defining relationships between datasets. Whereas relational databases are difficult to extend, in graph databases, only a few edges and nodes have to be added to the graph. Thus, the adaption and extension capabilities of the generic meta-model are supported by the schemaless definition of data in graph databases. For instance, additional information on customer and competitor intelligence, responsibilities, or other quantifiable business data can be easily integrated in the database schema. When using Neo4j, the integrated declarative query language CYPHER [17] supports most of the work. Thus, analysis models with extended and adapted functionality are greatly simplified and the graph-based approach is more efficient than business code migration on the software backend.

In situations, where the data set is quite homogenous and rarely changed, other architectural designs, like relational databases or in-memory databases, may be more appropriate. They also offer more support as well as advantages in the field of maturity. Regarding security, MySQL has an extensive security support based on access control lists. In contrast, graph databases like Neo4j expect a trusted environment.

 ISBN: 978-1-61208-427-5

V. PROTOTYPICAL IMPLEMENTATIONS

In the following, we will describe the implementation details of the ICT risk model in graph databases. The underlying approach covers semi-quantitative analysis steps usually used within risk models applying ISO 27005 [21]. We focus on the interconnections in the graph-based meta-model (cf. Figure 2), their representation in the graph database, and the implementation of the APT scenario and a physical attack scenario therein.

A. Graph Model

We propose the following meta-model, illustrated as a graph in Figure 2. Pre-existing information including goals, boundaries, and requirements are narratively documented within the nodes of the risk model. Risk identification is carried out by the definition of organizational assets (*risk model involves asset*) that are depicted by modules (*asset described_by module*), threats (*module threatened_by threats*), safeguards (*threat mitigated_by safeguard*) and roles (*roles responsible_for safeguard*). Goals can be defined based on the usual protection criteria (confidentiality, integrity, availability), as well as on requirements derived from other taxonomies. IT-Grundschutz [14] defines a respective risk catalogue, providing a categorization by module type (applications, IT systems, networks, infrastructure, common aspects), threat type (basic, force majeure, organizational shortcomings, human error, technical failure, deliberate acts), and safeguard type (infrastructure, organization, personnel, hardware and software, communications, contingency planning). Moreover, additional goals and requirements (e.g., stakeholder needs, enterprise goals, IT-related goals, etc.) coming from different frameworks like COBIT [22] can be integrated using cross-references with IT-Grundschutz. The defined goals correlate with the respective exposure of the components within the risk model, which translate to several risk dimensions (*risk model analyses protection criteria*).

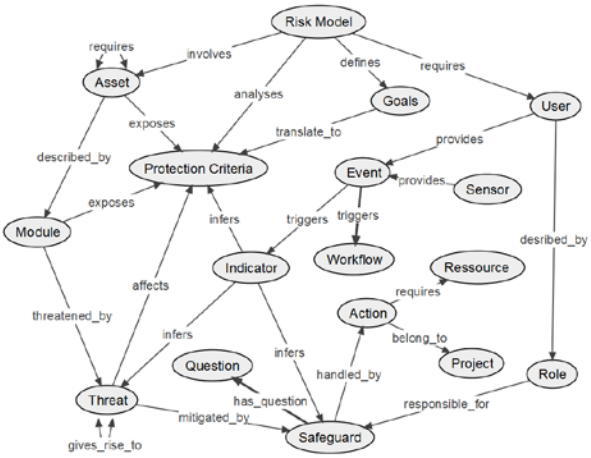

Figure 2 . Graph-based meta-model.

Risk estimation is based on the determination of safeguard maturities (supported by additional control questions, *safeguard has_question question*), threat likelihoods, and impacts on protection criteria. As a result of estimation, exposures are calculated for assets, modules, and threats separately (*asset/module/threat exposes protection criteria*) (cf. Section C).

Assets can optionally be related to each other during scenario analysis in order to depict their dependencies (*asset requires asset*). This supports business impact analysis and the option to perform risk propagation between scenario assets. Another optional step is to perform a detailed threat analysis by modeling threat cascades (*threat gives_rise_to threat*) based on the relationships of the pre-structured scenario model (*Asset requires Asset*).

Users can be associated to the risk model (*risk model requires user*) in specific roles (*user described_by role*) regarding the planning, implementation, and audit of required safeguards (*role responsible_for safeguard*). Users as well as automatable sensors using pre-aggregated data from external support systems (e.g., security information management solutions, i.e., security incident and event management systems (SIEM)) can provide measurements and events to the framework (*user/sensor provides event*), which can be used to trigger workflows (*event triggers workflow*), e.g., when a new IT system is detected.

The framework also provides a possible inference option between objective measurements and related subjective risk factors using fuzzy indicators (*event triggers indicator*) and an expert knowledge system. Inference targets are divided in indicators related to estimated impact, estimated likelihood, and estimated safeguard maturity (*indicator infers protection criteria/threat/ safeguard*).

In order to support basic risk management functionalities, safeguards can be summarized as organizational actions (*safeguard handled_by action*), which can be linked to resources and projects (*action belongs_to*). The intended purpose of the graph model is to provide an easy-to-extend and schemaless model with the ability to interrelate different types of nodes and to aggregate information across affected relationships.

B. Modeling the Scenario in the Graph Database

In the following, the graph-based model of the use case scenario described in Section III is discussed in detail (cf. Figure 3). Assets (blue ovals) are modeled by *_requires_* dependencies, which can be identified by a scenario analysis. The resulting structure defines the top-down inheritance between sub-systems and, at the same time, serves as default path for potential bottom-up threat cascades (*_gives rise to_*). Assets are connected to IT-Grundschutz modules (yellow hexagons) [14], where the referring relation is *described_by*. Threats (red trapezia) are linked to assets by *threatened_by* relations and associated with security measures (green rectangles) by *mitigated_by* relations. For the purpose of a detailed analysis, available threats can be combined to threat cascades via *gives_rise_to* relations. The business impact analysis model (*described_by*) and the IT-Grundschutz taxonomy itself indicate how these cascading paths might look like. This approach of modeling cascades might not address all of the potentially existing correlations, but it provides an easy way of dealing with chained probabilities.

 ISBN: 978-1-61208-427-5

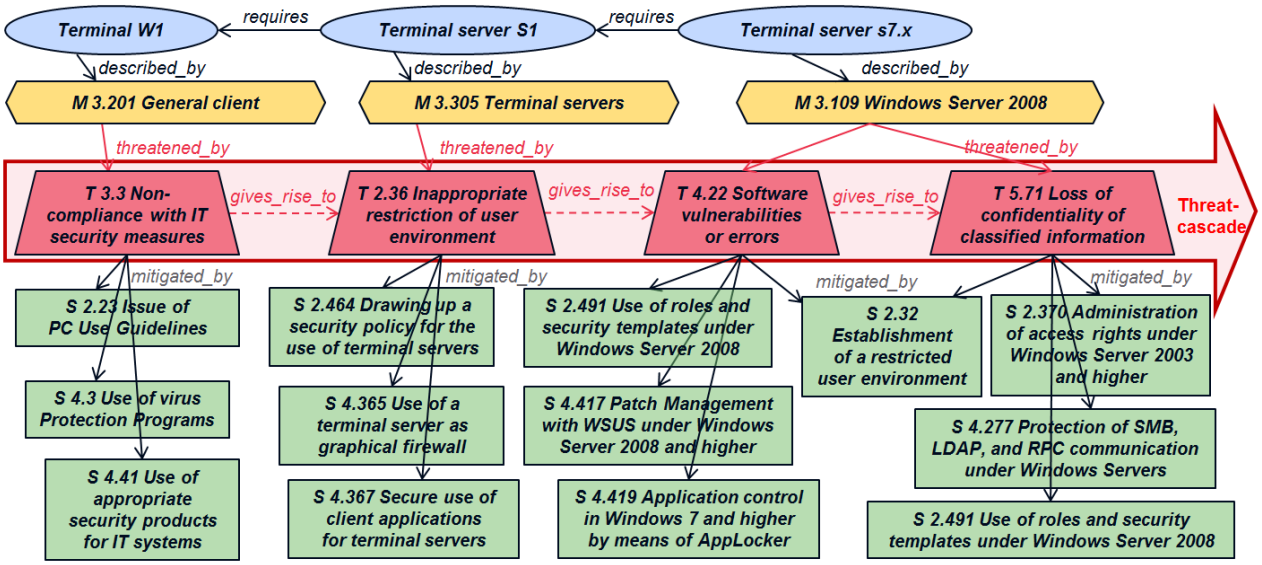

Figure 3 . Graph-based illustration of the scenario.

As described in Section III, at the Terminal W1 (*Module M 3.201 General client*) a user opens a spear phishing mail. This is an exploitation of the organizational threat *T 3.3 Non-compliance with IT security measures*, which is connected with the following security measures:

- *S 2.23 Issue of PC Use Guidelines*
- *S 4.3 Use of virus Protection Programs*
- *S 4.41 Use of appropriate security products for IT systems*

Afterwards, at the corresponding terminal server S1 (*Module M 3.305 Terminal servers*) a standard backdoor is installed. This is possible because of the threat *T 2.36 Inappropriate restriction of user environment*, which could have been addressed by the following security measures:

- *S 2.464 Drawing up a security policy for the use of terminal servers*
- *S 4.365 Use of a terminal server as graphical firewall*
- *S 4.367 Secure use of client applications for terminal servers*

Having gained access to the Terminal server S1, a software vulnerability scan is performed, helping the attacker to exploit the threat *T 4.22 Software vulnerabilities or errors* at the File server S 7.1 and, later on, at the file server S 7.2 (*Module 3.109 Windows Server 2008*). In the analyzed use case, the following security measures were not properly implemented:

- *S 2.32 Establishment of a restricted user environment*
- *S 2.491 Use of roles and security templates under Windows Server 2008*
- *S 4.417 Patch Management with WSUS under Windows Server 2008 and higher*
- *S 4.419 Application control in Windows 7 and higher by means of AppLocker*

At the same module, the follow-up threat *T 5.71 Loss of confidentiality of classified information* can be triggered, which is addressed by the following security measures:

- *S 2.32 Establishment of a restricted user environment*
- *S 2.370 Administration of access rights under Windows Server 2003 and higher*
- *S 2.491 Use of roles and security templates under Windows Server 2008*
- *S 4.277 Protection of SMB, LDAP, and RPC communication under Windows Servers*

In order to perform quantitative analyses, the risk inheritance between different components can be modeled by appropriate functions, e.g., maximum, sum, product, or minimum. More complex normalized, weighted, or bounded variants are also applicable. Possible candidates for the latter are weighted weakest link or prioritized sibling [15][23].

C. Results

In this section, it is demonstrated how the presented risk analysis approach can be used to derive (semi-)quantitative results (e.g. annualized loss expectancy (ALE)) based on semi-quantitative inputs (e.g., safeguard maturity levels according to the Capability Maturity Model Integration (CMMI) framework: 0.. Incomplete, 1.. Initial, 2.. Managed, etc.). In an analog way to the ATP attack example, the analyzed calculation example models the physical environment of an ICT infrastructure. A layered architecture is assumed (cf. Figure 4).

By using graph databases as model environment, the writing of complex business code for risk estimation can be avoided by performing the required assessments using CYPHER statements.

The outlined risk estimation method is a simplified variant of the method defined in [15]. The general view is that vulnerabilities of assets can be exploited by threat sources resulting in negative impacts on protection criteria. Thus, for risk estimation, the vulnerabilities of assets are explicitly taken into account; however, instead of using them directly, they are substituted by maturity gaps of safeguards.

 ISBN: 978-1-61208-427-5

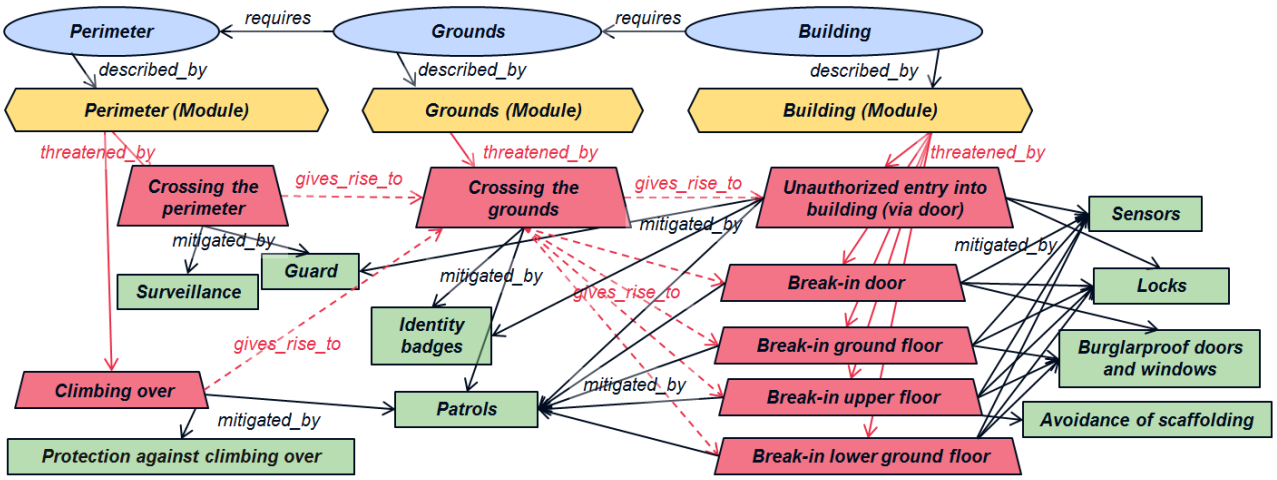

Figure 4 . Graph model for the physical environment scenario (excerpt).

This results in risk as a function of the likelihood of the occurrence of a threat, the maturity gap of an associated safeguard, and the impact that the unwanted event has on protection criteria (cf. (1)).

$$R := f(T_{likelihood}, S_{maturity\ gap}, I_{protection\ criteria}) \quad (1)$$

In an initial step, the safeguard requirements are derived from goals and estimated using maturity levels (from [0…5]). The product of the maturity gap (i.e., 1+maturity gap to evade division by zero) and the safeguard priority (from [1…4]) gives an estimation of the *safeguard exposure* (from [1…24]) (2). Additionally, the relation to the potential maximum exposure (based on the current goal definitions) is also calculated (cf. (3) (4) and Figure 5).

safeguard exposure *= (1+ maturity goal – estimated maturity) * safeguard priority* (2)

safeguard exposure max *= (1+ maturity goal) * safeguard priority* (3)

safeguard exposure % *= safeguard exposure / safeguard exposure max*100* (4)

```
match (a:USE_CASE:Asset{name_de:'x'})
-[r1:described_by]->(b:USE_CASE:Module{name_de:'x'})
-[r2:threatened_by]->(c:USE_CASE:Threat)
-[r3:mitigated_by]->(d:USE_CASE:Safeguard)
with (1+r3.target_maturity-r3.maturity)*r3.priority as
  exposure, (1+r3.target_maturity)*r3.priority as
  exposure_max, r3
set r3.exposure = exposure
set r3.exposure_max = exposure_max
set r3.exposure_rel = r3.exposure/r3.exposure_max*100
return r3
```

Figure 5 . Listing for the calculation of safeguard exposures.

After all safeguard exposures are calculated for each asset, the threat likelihoods are estimated for a specific timeframe (from [0…1], however, to simplify the CYPHER code, the null value is excluded to avoid a potential division by zero).

In a next step, the *threat exposures* are calculated. The threat exposure (from [0…20]) depends on the estimated likelihood (from [0…1]) and a function of its safeguard exposures (cf. (5)(6)(7) and Figure 6). For reasons of simplicity, here, the maximum function is used. In order to assess estimation variances, it may be appropriate to estimate the threat likelihood risk-averse (likelihood high) and risk-affine (likelihood low). Based on the calculation of current and potential maximum events, the risk factors within the model can be described either absolutely or relatively.

threat exposure *= likelihood(low) * MAX(safeguard exposure)* (5)

threat exposure max *= likelihood (high) * MAX (safeguard exposure max)* (6)

threat exposure % *= threat exposure / threat exposure max * 100* (7)

```
match (a:USE_CASE:Asset{name_de:'x'})
-[r1:described_by]->(b:USE_CASE:Module{name_de:'x'})-
[r2:threatened_by]->(c:USE_CASE:Threat)
-[r3:mitigated_by]->(d:USE_CASE:Safeguard)
with max(r3.exposure) as safeguard_exposure,
  max(r3.exposure_max) as safeguard_exposure_max, c
set c.exposure = c.likelihood*safeguard_exposure
set c.exposure_max =
  c.likelihood*safeguard_exposure_max
set c.exposure_rel = c.exposure/c.exposure_max*100
return c
```

Figure 6 . Listing for the calculation of threat exposures.

The threat exposures of all threats that have no incoming *gives_rise_to*-relationships are calculated first. The reason why the exposures of all uninfluenced threats are calculated initially is because no other threats have an effect on them (business impact analysis does not allow cyclic models).

After having calculated the threat exposures of all uninfluenced threats, the threat likelihood of all influenced threats (*gives_rise_to-relations*) can be updated based on the likelihood of their predecessors (chained likelihood). The calculation will be triggered as soon as all predecessors have been calculated. For reasons of simplicity, this is done by a simple multiplication of the original likelihood of the threat and the maximum of the likelihoods of its predecessors. Of course, a more complex function (weighting) representing the relative exposure of the threat to its influences can be

 ISBN: 978-1-61208-427-5

used. In the following example (cf. Figure 7), the originally estimated likelihood of threat *'y'* is multiplied with the maximum of all its incoming *gives_rise_to*-likelihoods.

```
match (a:USE_CASE:Asset{name_de:'x'})
-[r1:described_by]->(b:USE_CASE:Module{name_de:'x'})
-[r2:threatened_by]->(c:USE_CASE:Threat)
-[r3:gives_rise_to]->(d:USE_CASE:Threat{name_de:
  'Threat y', module_id:2})
with max(c.likelihood) as trigger_likelihood,
  d.likelihood as original_likelihood, d
set d.original_likelihood = original_likelihood
set d. likelihood = d. likelihood *trigger_ likelihood
return d
```

Figure 7 . Listing for the likelihood update of influenced threats.

After the likelihood update of all threats with incoming *gives_rise_to*-relations is finished, the remaining threat exposures can be calculated.

Depending on the desired level of detail, threats can be assessed individually or as generalized protection criteria related to assets (*asset exposures*), as illustrated by Figure 8. By extending the graph model, arbitrary aggregation layers can be defined. Here, to simplify the outlined use case, asset exposures are aggregated based on the maximum principle and risk is estimated based on the annualized loss expectancy (ALE) formula (cf. (8) and Figure 9). Again, additional lower and upper bounds could be integrated to express variance.

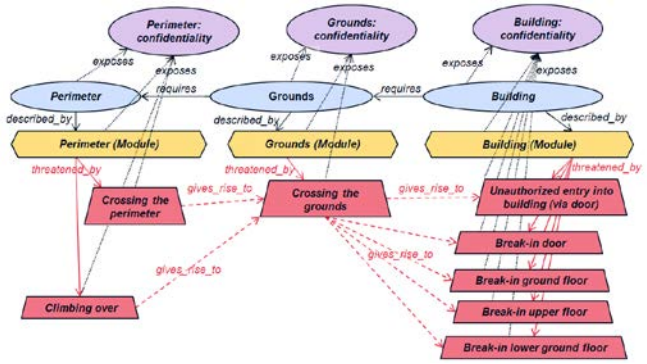

Figure 8 . Possible aggregation of exposures on asset-specific protection criteria (here: confidentiality).

$$\textit{asset risk} = \textit{estimated impact} * \textit{MAX (threat exposure \% / 100)} \quad (8)$$

```
match (a:USE_CASE:Asset{name_de:'x'})
-[r1:described_by]->(b:USE_CASE:Module{name_de: 'x'})
-[r2:threatened_by]->(c:USE_CASE:Threat)
with max(c.exposure) as threat_exposure,
  max(c.exposure_max) as threat_exposure_max, a
set a.exposure = threat_exposure
set a.exposure_max = threat_exposure_max
set a.exposure_rel = a.exposure/a.exposure_max*100
set a.ale_risk = a.impact*a.exposure_rel/100
return a
```

Figure 9 . Listing for the calculation of asset exposures and annualized loss expectancy (ALE) risks.

VI. CONCLUSION

This paper describes how a generic meta-model for risk management benefits from the quality of a graph-based implementation, especially from the features of schemaless information which can be parametrized based on the individual requirements of the organization, near-real-time traversals and flexible definitions of relationships between nodes, and the ability of easy model extension. An APT scenario is introduced to demonstrate a practical application of the presented meta-risk-model. The generic nature of the model allows addressing all kinds of threats - from the cyber over the physical to the business realm - and their dependencies. The consideration of cascading risk effects, including human-based information system vulnerabilities, is a necessary prerequisite for an effective defense against APTs, which exploit the full range of attack vectors, from social over digital to physical.

The presented approach shows the application of a combination of several analysis steps and different parts of existing methods, e.g., morphological matrices, fault-tree- and event-tree-analysis, scenario analysis, threat analysis, system decomposition, and functional relationships. The advantages of the presented combined approach are, for example, the possibility to focus on special requirements of information security and to cover a broader range of analysis depth and detail. These features cannot be achieved by using the previously mentioned methods on their own.

The introduced APT scenario is represented as a particular instance of a graph-based implementation of the generic meta-risk-model. The relevant risk components, which can be easily integrated into the graph-based meta-model, are provided by widely-accepted ICT risk frameworks, most importantly by IT-Grundschutz. The defined relations between relevant risk components within this framework give an excellent starting point for possible paths that potential cascading risk effects might take.

From a technical point, for modeling and inference analysis of threat cascades, the graph-oriented database Neo4j with its query language CYPHER was used. Threat cascades and their relations can be visualized by graph databases in a more optimized way compared to relational databases. The schemaless data model of graph databases allows an easier adaption during the modeling process and the application of traversals to integrate calculations without modifications of the business code. However, with regard to the correctness of the results, the domain has to be specified and defined with a low level of uncertainty, and the level of detail of the risk factors has to correlate with the granularity of the results to guarantee a consistent distribution of risk values. Within the discussed use cases, uncertainty resulting from subjective assessments, or inconsistencies and errors in modelling depth is not dealt with explicitly. It can be addressed, like any other aspect, by introducing semi-quantitative descriptors (e.g., assessment uncertainty, etc.), which can be aggregated within the graph model similar to other variables.

 ISBN: 978-1-61208-427-5

ACKNOWLEDGEMENT

The research project ”MetaRisk” (Project-Nr. 840905) is supported and partially funded by the Austrian National Security Research Program KIRAS (http://www.kiras.at/) [23].

REFERENCES

[1] T. W. Coleman, “Cybersecurity Threats Include Employees,” International Policy Digest. [Online]. Available: http://www.internationalpolicydigest.org/2014/05/12/cybersecurity-threats-include-employees/. [Accessed: 19-Mar-2015].

[2] SANS Institute, “Critical Security Controls: Guidelines.” [Online]. Available: http://www.sans.org/critical-security-controls/guidelines. [Accessed: 19-Mar-2015].

[3] Ponemon Institute, “Exposing the Cybersecurity Cracks: A Global Perspective. Part 2: Roadblocks, Refresh and Raising the Human Security IQ,” Traverse City, Michigan, USA, 2014.

[4] Mandiant Intelligence Center, “APT1. Exposing One of China’s Cyber Espionage Units,” Mandiant, Alexandria, Washington, DC, Feb. 2013.

[5] D. Moon, H. Im, J. Lee, and J. Park, “MLDS: Multi-Layer Defense System for Preventing Advanced Persistent Threats,” Symmetry, vol. 6, no. 4, Dec. 2014, pp. 997–1010.

[6] C. Tankard, “Advanced Persistent threats and how to monitor and deter them,” Network Security, vol. 2011, no. 8, Aug. 2011, pp. 16–19.

[7] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, “Combating advanced persistent threats: From network event correlation to incident detection,” Computers & Security, vol. 48, Feb. 2015, pp. 35–57.

[8] The Commission on the Theft of American Intellectual Property, “The IP Commission Report,” National Bureau of Asian Research, May 2013.

[9] G. Greenwald, No Place to Hide: Edward Snowden, the NSA, and the U.S. Surveillance State, Metropolitan Books, 2014.

[10] Internet Crime Complaint Center, “2013 Internet Crime Report,” Federal Bureau of Investigation, 2013.

[11] BMI, “Polizeiliche Kriminalstatistik 2013,” Bundes-ministerium des Innern, Berlin, 2013.

[12] Chartis Research, “Looking for Risk. Applying Graph Analytics to Risk Management. Leading practices from YarcData,” 2013.

[13] T. Schaberreiter, “A Bayesian Network Based On-line Risk Prediction Framework for Interdependent Critical Infrastructures,” Dissertation, University of Oulu, Oulu, Finlande, 2013.

[14] BSI, “IT-Grundschutz-catalogues 13th version 2013,” Bundesamt für Sicherheit in der Informationstechnik - Federal Office for Information Security, Bonn, Germany, 2013.

[15] S. Schiebeck, “An Approach to Continuous Information Security Risk Assessment focused on Security Measurements,” Dissertation, University of Vienna, Wien, 2014.

[16] J. Göllner, T. Benesch, S. Schauer, K. Schuch, S. Schiebeck, G. Quirchmayr, M. Latzenhofer, and A. Peer, “Framework for a Generic Meta Organisational Model,” paper presentation at the 14th FRAP - Finance, Risk and Accounting Perspectives Conference, Oxford, United Kingdom, 2014.

[17] Neo4j Graph Database, “Intro to Cypher - Neo4j Graph Database.” [Online]. Available: http://neo4j.com/developer/cypher-query-language/. [Accessed: 25-Mar-2015].

[18] R.-G. Urma and A. Mycroft, “Source-code queries with graph databases—with application to programming language usage and evolution,” Science of Computer Programming, vol. 97, Jan. 2015, pp. 127–134.

[19] C. Batra and C. Tyagi, “Comparative Analysis of Relational And Graph Databases,” International Journal of Soft Computing and Engineering, vol. 2, no. 2, May 2012, pp. 509–512.

[20] C. T. Have and L. J. Jensen, “Are graph databases ready for bioinformatics?” Bioinformatics, vol. 29, no. 24, Dec. 2013, pp. 3107–3108.

[21] ISO International Organizaion of Standardization, ISO/IEC 27005 - Information technology -- Security techniques -- Information security risk management, 2011.

[22] ISACA, “COBIT 5 - Enabling Processes,” Rolling Meadows, Illinois, 2012.

[23] C. Wang and W. A. Wulf, “Towards a Framework for Security Measurement,” in Proc. of 20th National Information Systems Security Conference, Baltimore, Maryland, 1997, pp. 522-533.

 ISBN: 978-1-61208-427-5