An Approach to Network Security Assessment based on Probalistic Relational Models

Fredrik Löf, Johan Stomberg,
Teodor Sommestad, Mathias Ekstedt
Industrial Information and Control Systems
Royal Institute of Technology (KTH)
Stockholm, Sweden

Jonas Hallberg, Johan Bengtsson
Swedish Defence Research Agency (FOI)
Linköping, Sweden

Abstract—**To assist rational decision making regarding network security improvements, decision makers need to be able to assess weaknesses in existing or potential new systems. This paper presents a model based assessment framework for analyzing the network security provided by different architectural scenarios. The framework uses a probabilistic relational model to express attack paths and related countermeasures. In this paper, it is demonstrated that this method can be used to support analysis based on architectural models. The approach allows calculating the probability that attacks will succeed given the instantiated architectural scenario. Moreover, the framework is scalable and can handle the uncertainties that accompany an analysis. The method has been applied in a case study of a military network.**

Keywords – **Probabilistic Relational Model, Network Security, Security Assessment, Attack Graph, Architecture Model**

I. Introduction

Many modern organizations depend on different types of information systems for business activities. An important aspect in these systems is network security as the consequences of a breach in security can be very damaging for the organization; from loss of trade secrets to theft and sabotage of critical infrastructure and services. With the importance of network security, it is natural to assess this aspect of a network. A security assessment might reveal unknown system weaknesses and show possible improvements, as well as work as a foundation for management and configuration decisions to find the most efficient application of resources when improving security.

A number of security assessment methods have been developed with different approaches to how security is evaluated. An overview of security measurement methods can be found in [32]. Attack graphs provide the foundation for several of these methods. An attack graph is a way to represent how an attacker can reach a goal in a system by defining what sub-goals the attacker must accomplish and plotting these different sub-goals as different paths or barriers the attacker must overcome [2,3,4,5]. In this way a security analyst can for instance find key points that must be protected or analyze a possible breach after the fact.

Scalability is an issue when constructing attack graphs. Recent results have decreased the complexity of the computations required to construct attack graphs [33,7,34]. However, the amount of input required to produce realistic attack graphs is considerable [35]. Also, it is often recommended that automated network scans should be avoided in operational industrial control system environments as these can disrupt operations [38,39]. To manually collect and update detailed network configuration data in this environment appears prohibitively expensive. Moreover, from a human point of view, attack graphs quickly become ungraspable due to their size and complexity. It is thus difficult to use the attack graphs to try out new, alternative solutions by manually changing some parameters in the graphs.

Probabilistic treatment of the relationship between different attack steps is an alternative solution to this challenge. This solution can also reduce the amount of input needed to model attacks. In [36] Bayesian networks are used to represent possible attacks more compactly and to calculate the probability that a network attack succeeds, as opposed to using deterministic attack graphs.

This paper describes a method to assess network security based on the Probabilistic Relational Model (PRM) formalism [1], which is a combination of Bayesian networks, attack graphs, and architectural models. The probabilistic approach can make predictions without all the details included in traditional attack graphs [35]. The drawback of this approach is the precision in the assessment since it by nature makes probabilistic estimations.

The basis for an analysis in the proposed method is a metamodel describing the system architecture, in terms of its components and their attributes, as well as possible attacks, in the form of conceptual attack graphs. With such a metamodel as a basis an analyst creates an instance model representing the architecture of the network. This instance model is used to calculate probabilities that an attacker might succeed with different potential attacks on the system architecture, thus providing decision-makers with information regarding network security.

The purpose of this article is to test whether PRMs can be used to assess network security. Thus, while the approach in general is intended for the security analysis of architecture models, this paper focuses on a PRM for communication networks.

The method has been applied in a study to assess the network security of a military network from the international interoperability exercise Combined Endeavour 07. A short overview of this case is also presented in this paper.

II. Probabilistic Relational Models

A Probabilistic Relational Model [1] specifies the metamodel for the architecture models and the probabilistic dependencies between attributes of the architecture objects. A PRM defines a probability distribution over the attributes of the objects in an instantiated architecture model. The probability distribution can be used to infer the values of unknown attributes. This inference can also take into account evidence on the state of observed attributes.

A. Architecture metamodel

An architecture metamodel, M, describes a set of classes. Each class X is associated with a set of descriptive attributes and a set of relationships between attributes and classes. For example, a class Firewall might have the descriptive attribute Bypass Packet Filtering, with the domain {True, False} and the relationship Perimeter Defense to the class Host (cf. Figure 1). A relationship between attributes from these classes can then be defined through the class relationship. Every attribute has a conditional probability table (CPT) describing the probability distribution for the values of the attribute.

B. Architecture instance models

An architecture instantiation I (or an architecture model) specifies the set of objects in each class X, and the values for attributes, X.A, and relationships, X.r, of each object. For example, Figure 2 presents an instantiation of the metamodel described in Figure 1. It specifies a particular Firewall (Färist, [37]), two Authentication Services, two NIDS (Network Intrusion Detection System), one HIDS (Host-based Intrusion Detection System) and one Host.

C. Probablistic model over attributes

A PRM specifies a probability distribution over all instantiations I of the metamodel M. Like a Bayesian network [8] it consists of a qualitative dependency structure and associated quantitative parameters. The qualitative dependency structure is defined by associating attributes X.A with a set of parents Pa(X.A). This is done by specifying a search path through relationships in the instance model that describes the parents. For example, the attribute Host.MaliciousCodeAttack has a parent Host.*PerimeterDefense*.BypassContentFiltering, meaning that the possibility of an attack with malicious code against a host depends on the probability that the attack bypasses the content filtering of the host's perimeter defense. It is the class relationship between Firewall and Host – called Perimeter Defense – that makes the attribute relationship possible.

We can now define a PRM for a metamodel M as follows. For each class X and each descriptive attribute A ∈ At(X), we have a set of parents Pa(X.A), and a conditional probability distribution that represents P(X.A|Pa(X.A)).

An attribute A only has one CPT but can have multiple parents of the same type. This is solved with an aggregation function such as MAX or MIN on the relationships between the attribute and its' parents. The aggregation functions work over all the parents of the same type and find one value to use for the instance attribute CPT.

A PRM thus enables the calculation of the probabilities of various architecture instantiations. This makes it possible to infer the probability that a certain attribute assumes a specific value, given some – possibly incomplete – evidence about the rest of the architecture instantiation.

III. Architectural Metamodel

A PRM can have many perspectives, this PRM consider the probability of successful attacks. In the study that was performed on the Combined Endeavour 07, the main target, and the focus of the analysis of the PRM was to investigate the difficulty of acquiring administrator level rights on a host. The metamodel with attribute relationships is presented in Figure 1.

A. Development of the metamodel

The metamodel was developed in two steps. First, a qualitative core structure describing different system components and their interconnected relationships was developed. Second, quantitative values were added to populate the conditional probability tables. The model was validated by consulting multiple domain experts.

The qualitative structure of the PRM was defined from a literature study that addressed common security components – how they can affect possible attacks and protect other network components. Literature was drawn from NIST [9, 10, 11, 12], NISCC [13, 14] and papers [15,16,17,18,19,20,21,22,24,25,26]. Five classes representing components and a number of component attributes were selected as the most relevant for the model. The selections were deemed to be representative for common protection technologies and possible attack vectors, with a high level of abstraction. With the classes and attributes in place, their relationships with other classes and attributes were addressed. The resulting core model was validated by multiple domain experts in a number of interviews where they confirmed the attribute selections and the definition of the qualitative structure as relevant. The experts consulted were a senior security consultant from a leading European network security firm, as well as multiple senior members and alumni of the student computer association at the Royal Institute of Technology – some of them with many years of experience working in the field of network administration and security.

In the second step of development, quantitative data was added to the model in the construction of the CPTs. The CPTs require a probability distribution to be defined for the conditions introduced by the qualitative dependency structure.

A few statistical studies were found regarding the efficiency of certain security functions given certain conditions. For instance, [27] was used to specify the conditional probability for *Bypass Anomaly Based Detection* in the NIDS class. To complement the literature, the network experts mentioned above were consulted in further interviews. They provided approximations for the probability of certain types of attacks to succeed different conditions, as well as participated in discussions regarding the validity of other numbers drawn from the literature. This way most CPTs were discussed and validated. See part III.D for two examples of attributes.

Table 1 list all attributes in the PRM. This table also shows the literature used to define the qualitative structure and the quantitative parameters associated with each parent. The qualitative structure is the parents defined in the PRM; the quantitative structure is the CPT for the attribute. The table also lists an uncertainty level for every CPT that is further described in the list of the attributes. The attributes have support in an average of three references.

B. The resulting PRM

Figure 1 shows the final PRM metamodel. The model consists of five *classes* representing common security components in a network: Firewall, Authentication, NIDS, HIDS and Host. The classes have *class relationships* between them, such as Perimeter Defense that represents that a firewall can provide perimeter defense for other components. Every class has a number of *attributes* representing attack steps that can be directed towards the component, or security functions that the component can provide. The latter case represents functions that an intruder must bypass. For instance, the attribute *Exploit Remote Access* in the class *Firewall* represents how an attacker might attempt to gain control over a firewall through remote configuration. Another example is the attribute Bypass Content Filtering representing the chances of bypassing the content filtering of a firewall during an attack. Between the attributes there are arrows showing *parent relationships*. In the class Firewall there is an arrow from Exploit Remote Access to Bypass Content Filtering showing that the probability of bypassing content filtering is influenced by the probability of successfully exploiting the remote configuration, e.g. an attacker might disable the content filtering through remote configuration.

C. List of attributes

Table 1 presents all the attributes in the metamodel along with related references, sorted under their respective classes. In the first column is the name of the attribute. The second column lists the qualitative references that were used to define the parents of the attribute. The third column lists the quantitative references for the CPT of attributes. The asterisks indicate that the corresponding CPTs are defined through the logic of the model structure. For instance: Pa(*SpoofAttack*)= *BypassSpoofCountermeasure* as the only parent. If there is a countermeasure and the bypass attack succeeds then the spoof attack will also succeed. The asterisk references primarily function as logic OR. Double asterisks show tables that are based solely on expert approximation.

The fourth column represents the uncertainty level of the CPT, graded as Low (L) or High (H). Low means that the CPT has multiple sources and that the experts were more certain in their estimates. High means that the CPT should be prioritized for further research as it is primarily based on expert approximation and can be improved with more systematic case studies.

D. Sample attributes

To show the development of the attributes and what they represent in a modeling context, two attributes will be described in detail. The first example is *Bypass Signature-based Detection* from the class NIDS. Signature-based detection is described in [9, 15, 16] and is a core security

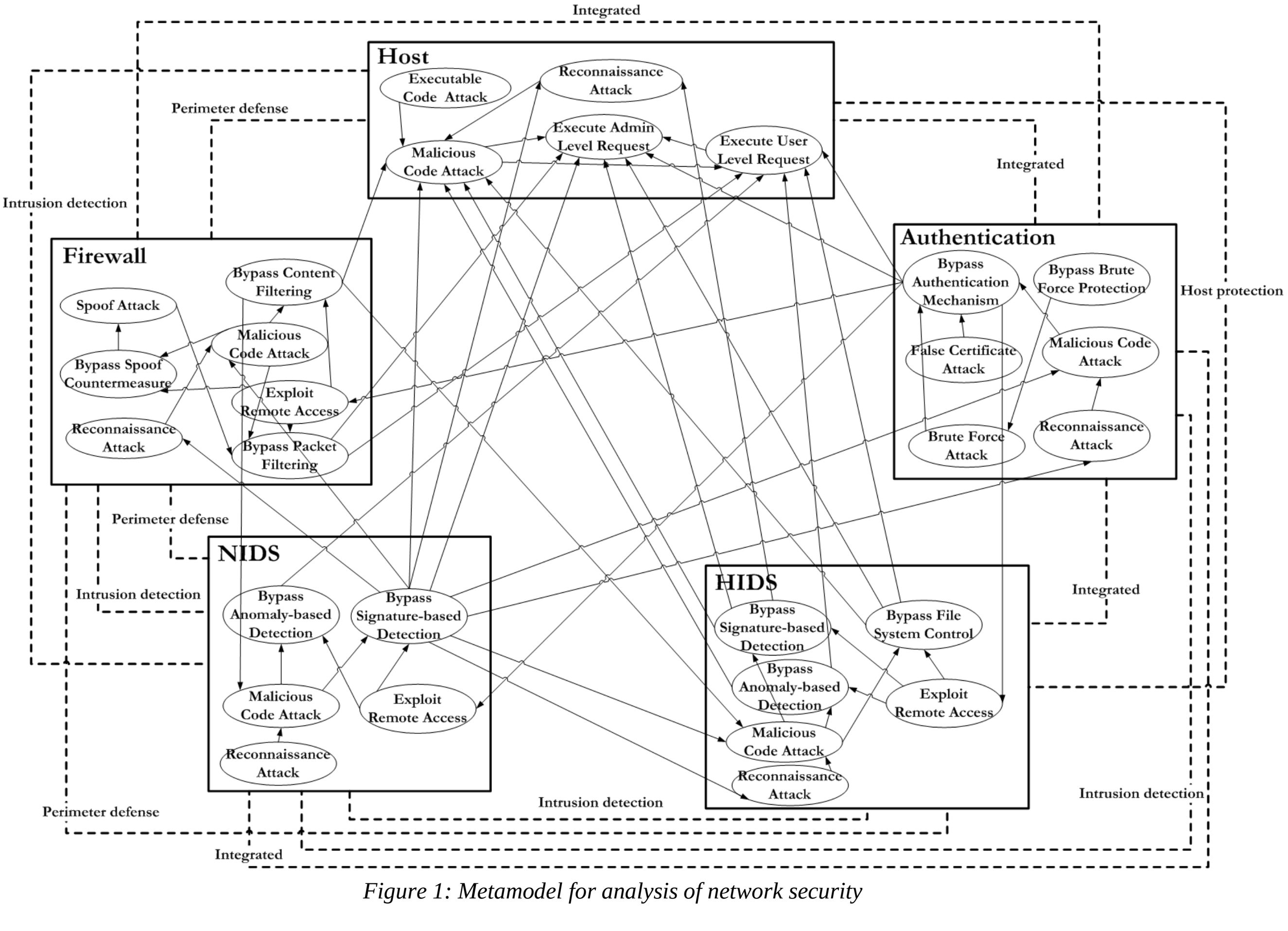

Figure 1: Metamodel for analysis of network security

principle used by different NIDSs. The attribute represents the attack step of evading this detection, which in this paper is described as a bypass attack.

Table 1: List of attributes in metamodel

Classes and attributes	Qualitative	Quantitative	Uncertainty
Firewall Class			
Bypass Packet Filtering	[10,13, 17, 18]	**	H
Spoof Attack	[10, 13,19]	*	L
Bypass Spoof Countermeasure	[10,13,19,11]	**	H
Reconnaissance Attack	[9,15,20,16,21]	**	L
Bypass Content Filtering	[10,17,19]	**	H
Malicious Code Attack	[18,19,22,17]	[23,24]	H
Exploit Remote Access	[10,18]	*	L
Authentication Service Class			
Bypass Authentication mechanism	[17,22,19,25]	*	L
False Certificate Attack	[17,11]	**	H
Brute Force Attack	[11,25]	*	L
Bypass Brute Force Protection	[11,17,25]	[11,17]	H
Reconnaissance Attack	[26]	**	L
Malicious Code Attack	[19,21,26,12]	[23,24]	H
NIDS Class			
Bypass Signature Based Detection	[9,15,16]	[27]	L
Bypass Anomaly Based Detection	[9,15,16,19]	[27]	L
Reconnaissance Attack	[26]	**	L
Malicious Code Attack	[19,28,17,25]	[23,24]	H
Exploit Remote Access	[22]	*	L
HIDS Class			
Bypass Signature Based Detection	[9,15,16]	[27]	L
Bypass Anomaly Based Detection	[9,15,16]	[27]	L
Bypass File System Control	[9,15,20,14]	[29]	L
Exploit Remote Access	[22]	*	L
Reconnaissance Attack	[26]	**	L
Malicious Code Attack	[19,28,17,25]	[24,23]	H
Host Class			
Admin Level Request	[26,14]	**	H
User Level Request	[26,14]	**	H
Malicious Code Attack	[26,22,21]	[23,24]	H
Reconnaissance Attack	[26]	**	L
Executable Code Attack	[19,12]	[30]	L

The attribute has two states, either the attack step succeed in bypassing the detection or it does not: True (T) or False (F) respectively. For this attribute there are two parents: *Malicious Code Attack* and *Exploit Remote Access*. A malicious code attack is an attack where the goal is to exploit a vulnerability in the target through remote code execution. Exploiting remote access refers to an attack against a remote configuration interface on the NIDS. Remote configuration over the network facilitates the work of system administrators, but also represents a significant weakness as an attacker might gain unauthorized access and shut down security measures.

The CPT for the attribute Bypass Signature-based Detection is captured in Table 2. If it is possible to do a malicious code attack against the NIDS, then the signature-based detection will be disabled (the third and fourth columns in Table 2), i.e. *Bypass Signature-based Detection* is true. Analogously, the detection can also be disabled if it is possible to exploit the remote access of the NIDS (column one and three in Table 2). If none of these attacks succeed, then the detection rate of the NIDS defines how often the intrusion will fail, in this case 38 % (the last column in table 2). This value is drawn from a study of Snort without customized signatures or any system-specific training: *"The Snort has a flat low detection rate of 38% with any rate of false alarms."* [27]. The results of the study are generalized and assumed representative for many IDSs, while the other numbers are derived from the modeling logic. Finally, the CPT is assumed to have low uncertainty as the IDS study examines a common version of Snort under given conditions, giving exactly the type of attack statistics preferable for the model.

Table 2: CPT for the attribute Bypass Signature-based Detection

NIDS.MaliciousCodeAttack		T		F	
NIDS.ExploitRemoteAccess		T	F	T	F
NIDS.BypassSignatureBasedDetection	T	1	1	1	0.62
	F	0	0	0	0.38

The second example is the attribute *Bypass Spoof Countermeasure* from the Firewall class. This attribute is described in [10,19,13,17] and represents the ability of the firewall to detect or prevent address spoofing. As in the first example, the attribute has two states in this case signifying whether an attack with a spoofed address bypasses the countermeasure. The parents are *Malicious Code Attack* and *Exploit Remote Access* that both function basically the same as in the first example.

The CPT for *Bypass Spoof Countermeasure* is shown in table 3. If either of the two attacks directed towards the spoof countermeasures succeed, the countermeasures will be disabled and thus bypassed (columns three through five). If both types of attacks fail, then the efficiency of the countermeasures will determine whether the spoofing succeeds. In this case there was no appropriate literature with relevant statistics so the experts were consulted to find the needed numbers. An interview was conducted with three experts, with the first question being a discussion about spoofing and spoof countermeasures in general. Different types of countermeasures were discussed, such as ingress and egress filtering. The experts were asked to judge the efficiency of such countermeasures for a properly configured network with a security level that could be described as “awareness”, e.g. not the chief concern and neither an ignored subject in the administration of the network.

Table 3: CPT for attribute Bypass Spoof Countermeasure

Firewall.MaliciousCodeAttack		T		F	
Firewall.ExploitRemoteAccess		T	F	T	F
Firewall.BypassSpoofCountermeasure	T	1	1	1	0.05
	F	0	0	0	0.95

According to the experts, spoofing is straightforward to perform and something that an attacker can be assumed to succeed with in nearly every case if there is no spoof countermeasure. If the attacker is unable to exploit remote access or perform a malicious code attack to disable the spoof countermeasures, the attacker might be able to bypass the countermeasures in a small number of cases. The domain experts suggested that approximately 5% of installed firewalls would allow spoof countermeasures to be bypassed even if both parents are false. This is represented in the rightmost column.

IV. Instance model – CE07

The metamodel was applied in an assessment of network security in the military network Combined Endeavor 07. Combined Endeavor is an annual international interoperability exercise between the defense forces of NATO and Partnership for Peace, including more than 40 nations in 2007. The participants were organized in different regions and every region built an IP-backbone for interconnection of all subnets, transmission links and network components. In the study one region was evaluated through two different scenarios wherein attacks against two targets were modeled. Instance models were created by finding possible attack paths and adding instances of the network components that could affect the hypothetical attack. With the metamodel as a basis, the user needed only provide a small amount of information to perform this analysis. One instance model is shown in Figure 2. Attribute relationships are not presented in the figure due to lack of space, but they of course follow the metamodel.

An attack towards a host behind a Färist firewall, multiple IDSs and with authentication systems is described with an instance model. This gives seven objects that are connected as shown in Figure 2. Evidence is added in the instance model where the attributes of the actual components are known to deviate from the default values of the metamodel. One example is the NIDSs which lack anomaly-based detection, and this is described by setting the attribute *Bypass Anomaly-based Detection* to True.

To find the results, the inferred values of all attributes are calculated in the software Enterprise Architecture Tool (EAT) which is being developed for this type of system evaluation [31]. In this scenario the result is defined as the probability of executing a request in the target, i.e. the value of the attributes *Execute User Level Request* and *Execute Admin Level Request* in the object *Target host*. The result was two percentage values giving an estimate of system security, in this case for both attributes a 0.00702 probability of an attacker successfully executing a request. An alternate scenario was also tested where anomaly-based detection was added to the IDSs, and this gave a value of 0.00364.

V. Discussion

There are some issues regarding scalability, model scope, and data collection that should be noted. The scalability problem of attack graphs is partly solved by this approach as the metamodel and classes compartmentalize many factors

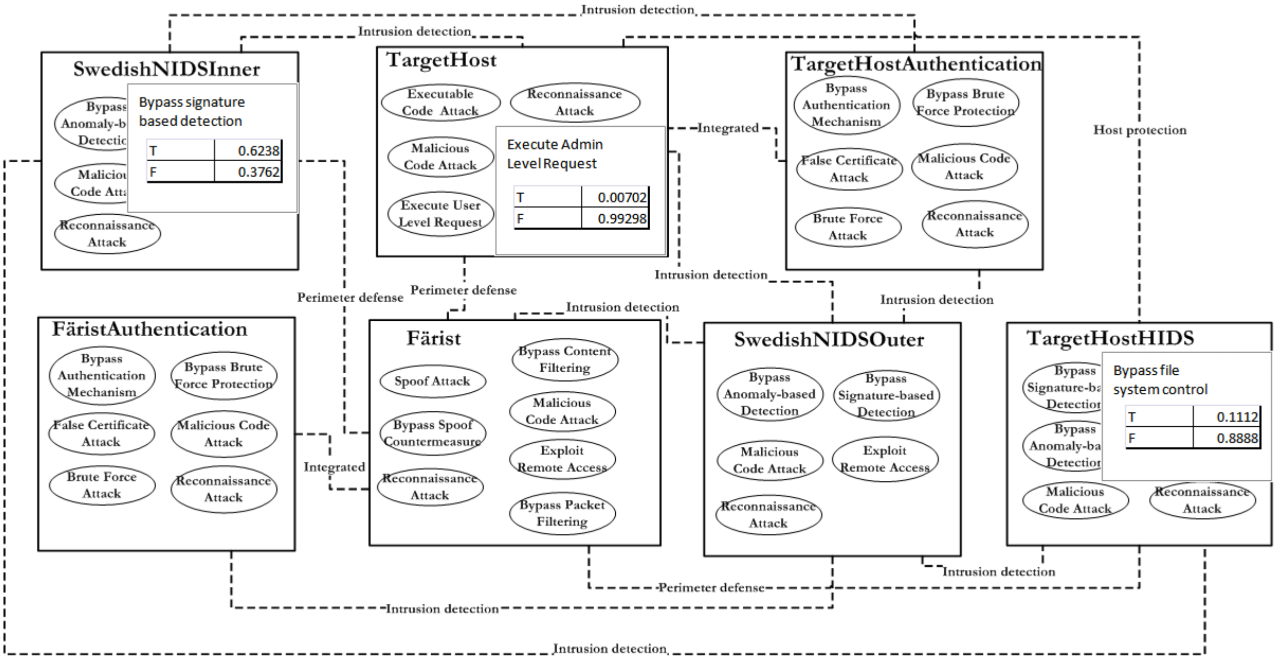

Figure 2: Instance model from security assessment scenario

regarding attacks. While a large network might result in a complex instance model with many classes and class relationships, it is still a relatively low number of entities to handle for the analyst.

Another aspect to consider is the scope of the model. The metamodel constructed for this study has a high level of abstraction rather than extensive technical details, which naturally means that some factors that could be argued to be relevant have been excluded or combined in abstract attributes.

By using probabilistic methods the model considers many factors indirectly. Compared to deterministic methods the probabilistic method does not need to explicitly include every aspect and the abstraction level can be higher. It must be acknowledged though, that the model can be improved upon in this regard. Statistics and averages are easier to apply with a probabilistic method and this is sufficient information for practical use of the PRM. All information does not have to be exact and correct to facilitate a decision when relative comparisons are done.

An additional benefit with this approach is that the instantiated models can be quite easily comprehended by industrial security analysts as well as other system engineers and analysts since the models are not overwhelmingly large. The models are aligned with many commonly used metamodels for specifying system architectures.

When performing the study, the main problem was finding good sources for the large number of probability tables. While some data is based on the consensus of multiple experts' opinions, the model would certainly benefit from more objective data. Almost all the experts found it hard to provide the requested data before they got a rigorous explanation of the method. One expert whom had worked with attack graphs earlier found it much easier than the others. Thus, it is possible to improve the metamodel regarding the level of abstraction and numbers in the probability tables. A further refinement could be done with a deeper level of detail and future research regarding intrusion statistics. With comprehensive component data this might give very accurate results, even though the method aims for simplicity rather than describing the whole truth of security matters.

VI. Conclusions

After applying this method in a study it can be concluded that it is possible to use PRM methodology for security assessments. The development of the metamodel takes time and requires a large amount of data to populate the CPTs with numbers, but it is also a very important step in the process. This is the main strength of the method, as a community of experts can collect and define the necessary theory in a metamodel that an analyst then can apply quickly and easily without the same level of knowledge in security theory.

Another advantage is the flexibility and abstraction in theory construction, as the metamodel can be further customized depending on the exact subject and focus of study. Depending on requirements, organizational policies and the type of network to study, the theory model can be defined to cover the precise aspects needed for a study or an organization. This flexibility also means that the PRM can be extended to cover more classes and attributes.

The developed network PRM and CPTs give a good basis for a basic intrusion and security analysis. While the results from the study give a good guideline and approximation of security, the method does not claim to describe the complete truth. The metamodel focuses on a range of technical aspects with a high level of abstraction regarding subjects such as malicious code and user behavior. Areas such as on-site security, attacker profiles or administrator action are not directly covered and could be added in further refinement of the metamodel.

VII. References

[1] Getoor L, Taskar B. *Introduction to statistical relational learning.* s.l. : MIT Press, 2007.

[2] C. Ramakrishnan, R. Sekar. Model-Based Analysis of Configuration Vulnerabilities. *Proceedings of the 7:th ACM Conference on Computer and Communication Security.* November 2000.

[3] R. Ritchey, P. Ammann. Using Model Checking to Analyze Network Vulnerabilities. *Proceedings of the IEEE Symposium on Security and Privacy.* 2000.

[4] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J.Wing. Automated Generation and analysis of Attack Graphs. *Proceedings of the IEEE Symposium on Security and Privacy.* 2000.

[5] C.Phillips, L.Swiler. A Graph-Based System for Network-Vulnerability Analysis. *Proceedings of the New Security Paradigms Workshop.* 1998.

[6] P. Ammann, D. Wijesekera, S.Kaushik. Scalable, Graph-Based Network Vulnerability Analysis. *Proceedings of the 9:th ACM Conference on Computer and Communications Security.* November 2000.

[7] X. Ou, W. Boyer, M. McQueen. A Scalable Approach to Attack Graph Generation. *Proceedings of the 13:th ACM conference on Computer and Communications security.* 2006.

[8] Jensen, F V. *Bayesian Networks and Decision Graphs.* Secaucus, NJ : Springer New York, 2001.

[9] *Guide to Intrusion Detection and Prevention Systems.* Scarfone K, Mell P. 2007, NIST SP800-94.

[10] *Guidelines on Firewall and Firewall Policy.* Scarfone K, Hoffman P. 2008, NIST SP800-41.

[11] *Recommendation for Key Management.* et al, Barker. 2007, NIST SP800-57.

[12] *An Introduction to Computer Security: The NIST Handbook.* Sterne, et al. 1995, NIST SP800-12.

[13] *Understanding Firewalls.* NISCC. 2005, NISCC Technical Note 10/04.

[14] *Understanding Intrusion Detection Systems.* NISCC. 2003, NISCC Technical Note 09/03.

[15] *Study of Intrusion Detection Systems (IDSs) in Network Security.* Wu J, Hu Z. 2008, Conference on Wireless.

[16] *Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems.* Garuba, et al. 2008, Fifth International Conference on Information Technology: New Generations.

[17] Orrey, et. al. Penetration Testing Framework 0.54. *vulnerabilityassessment.co.uk.* [Online] 10 08, 2009. [Cited: 10 08, 2009.] http:www.vulnerabilityassessment.co.uk.

[18] *FireCracker: A Framework for Inferring Firewall Policies using Smart Probing.* Samak, et al. 2007, IEEE International Conference on Network Protocols.

[19] *Just How Secure Are Security Products?* Geer, D. 2004, Computer June.

[20] *Network Security on the Intrusion Detection System Level.* Vokorokos, et al. 2006, INES.

[21] *Network Intrusion Detection – Automated and Manual Methods Prone to Attack and Evasion.* Chaboya, et al. 2006, IEEE Privacy & Security, Volume 4 Issue 6.

[22] *On the Use of Security Metrics Based on Intrusion Prevention System Event Data: An Empirical Analysis.* Chrun, et al. 2008, 11th IEEE High Assurance Systems Engineering Symposium.

[23] *Common Vulnerability Scoring System.* Scarfone, et al. 2006, Security & Privacy, Nov-Dec.

[24] *Estimating Software Vulnerabilities.* J, Jones. 2007, Security & Privacy, July-Aug.

[25] *Packet Saga, Using Strategic Hacking To Terrorize Commercial And Governmental Entities On The Internet.* Nassar K, Ali W. 2005, 3rd International Conference on Information & Communication Technology.

[26] *Policy Management for Network-based Intrusion Detection and Prevention.* Chen Y, Yang Y. 2004, Network Operations and Management Symposium.

[27] *Defending Distributed Systems Against Malicious Intrusions and Network Anomalies.* Hwang, Chen, Liu. 2005, 19th IEEE International Parallel and Distributed Processing Symposium.

[28] *Towards Survivable Intrusion Detection System.* Yu D, Frincke D. 2004, Proceedings of the 37th Hawaii International Conference on System Sciences.

[29] McAfee. *Anti-Malware Detection Rates Comparative Testing.* s.l. : West Coast Labs , 2008.

[30] Cisco Systems. *Understanding Remote Worker Security: A Survey of User Awareness vs. Behavior.* s.l. : Cisco Systems White Paper, 2006.

[31] Department for industrial information and control systems, KTH. *Project page for Enterprise Architecture Tool.* [Online] http://www.ics.kth.se/eat.

[32] V. Verendel, "Quantified security is a weak hypothesis: a critical survey of results and assumptions," *New Security Paradigms Workshop*, 2009.

[33] Zhang, B., Lu, K., Pan, X., & Wu, Z. (2009). Reverse Search Based Network Attack Graph Generation. In *2009 International Conference on Computational Intelligence and Software Engineering* (pp. 1-4). IEEE. doi: 10.1109/CISE.2009.5365235.

[34] Anming Xie, Guodong Chen, Yonggang Wang, Zhong Chen, Jianbin Hu, "A New Method to Generate Attack Graphs,", pp.401-406, 2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement, 2009

[35] Roschke, S., Cheng, F., Schuppenies, R., & Meinel, C. (2009). Towards Unifying Vulnerability Information for Attack Graph Construction. In *Proceedings of the 12th International Conference on Information Security* (p. 233). Springer.

[36] Y. Liu, M. Hong, Network vulnerability assessment using Bayesian networks, in proceedings of SPIE, , pp. 61-71, Orlando, Florida, USA, 2005.

[37] Tutus Digital Gatekeepers, Retrieved at: http://www.tutus.se/farist-fw.html, 2010-03-19.

[38] Stouffer, K., Falco, J., & Kent, K. (2008). Guide to Industrial Control Systems (ICS) Security Recommendations of the National Institute of Standards and Technology. *NIST Special Publication*, *800*-82.

[39] Finco, G., & others. (2007). Cyber Security Procurement Language for Control Systems. *Idaho National Labs*, (August).