A Generic Metamodel for IT Security – Attack Modeling for Distributed Systems

André Miede, Nedislav Nedyalkov, Christian Gottron, André König, Nicolas Repp, and Ralf Steinmetz
Technische Universität Darmstadt, Multimedia Communications Lab (KOM), Darmstadt, Germany
Corresponding Author: Andre.Miede@KOM.tu-darmstadt.de

***Abstract*—Understanding and discussing the security aspects of IT systems during their development is challenging for both domain specialists and IT experts – neglecting this aspect leads to communication problems and, eventually, to less secure systems. An important factor for these challenges is the distribution and variety of basic IT security concepts, attacks, and countermeasures, e. g., in the standard literature. In this paper, we propose a generic metamodel for IT security capturing both its major concepts and their relationships to each other. With a focus on attacks, we show how this model is applied to different scenarios in distributed systems, i. e., Peer-to-Peer systems, Service-oriented Architectures, and Mobile ad hoc Networks. This allows for a better understanding of IT security in general and attacks in particular, thus, enabling effective communication between different parties during the development of security-critical IT systems.**

***Keywords*-Security, Metamodel, Attack Modeling, Distributed Systems**

I. Introduction and Motivation

As information technology (IT) gets more and more pervasive and ubiquitous in our daily lives, government and business scenarios, the importance of IT security increases rapidly. IT security becomes both an enabler and a necessity, facing challenging demands in order to achieve the basic security goals such as confidentiality, authentication, authorization, non-repudiation, integrity, and availability [1]–[3].

In addition to domain experts and system engineers, whose collaboration towards the functional effectiveness of a system is already challenging, security engineers [4] are required as early as possible in the system life cycle. Thus, all involved parties need a common understanding of IT security concepts and methodological support for communicating about these concepts.

The goal of this paper is to assemble and analyze the major elements of IT security and their relationships to each other. For this, a metamodel for IT security is developed in order to facilitate a better understanding of IT security in general and attacks in particular. This aims at enabling effective communication between different parties during the development of security-critical IT systems both in industry and research.

The rest of the paper is structured as follows: Section II gives an overview of the proposed IT security metamodel which is based on the standard literature in this area. It comprises a common core of security concepts, countermeasures, and attacks. Section III shows how the metamodel is applied to modeling attack scenarios in distributed systems, i. e., Peer-to-Peer systems, Service-oriented Architectures, and Mobile ad hoc Networks. Section IV sums the findings up and closes with a brief outlook on future work, i. e., extending the metamodel regarding countermeasures.

II. A Generic Metamodel for IT Security

IT security is a complex system consisting of various different concepts and their relationships. Thus, modeling these concepts becomes an important task in order to specify and visualize the structure of the system and to document the definitions and consensus achieved [5].

The Object Management Group (OMG) offers insights on the understanding of metamodels and models [6]: a metamodel is a model of a model, a means to describe and design models. A model then instantiates the metamodel, providing support to describe various problem domains. In the end, there are the concrete instances of model elements. To illustrate this for the IT security domain, a metamodel element would be *Confidentiality* as a *Security Goal*. A possible model for this is *Encryption*, of which a concrete instance is the *Twofish algorithm*. This can be continued up to specific implementations.

The metamodel which we propose to serve as a foundation for modeling IT security problems is shown in Figure 1. It consists of three main parts: a *Core* of basic IT security concepts, *Countermeasures*, and *Attacks*. For simplicity reasons, not all possible relationships are shown, only the ones we consider the most important. Some concepts are also known under other names, than the ones presented here. Due to their importance and space constraints of this paper, the focus is on presenting the components of attacks, while the other two parts are only briefly introduced.

A. Core and Countermeasures

The *Core* of the metamodel brings together the most basic IT security concepts (denoted by black boxes), assets, attacks, and countermeasures. In the following, the core and countermeasure elements are outlined.

Assets are the objects and concepts to be protected from attacks [8] by striving for common security goals such as confidentiality, authorization etc. A *Threat* is a possible way a system can be attacked [8]. Threats can be categorized in four classes according to their consequences [9]: *Disclosure* (unauthorized access to data), *Deception* (provision of false

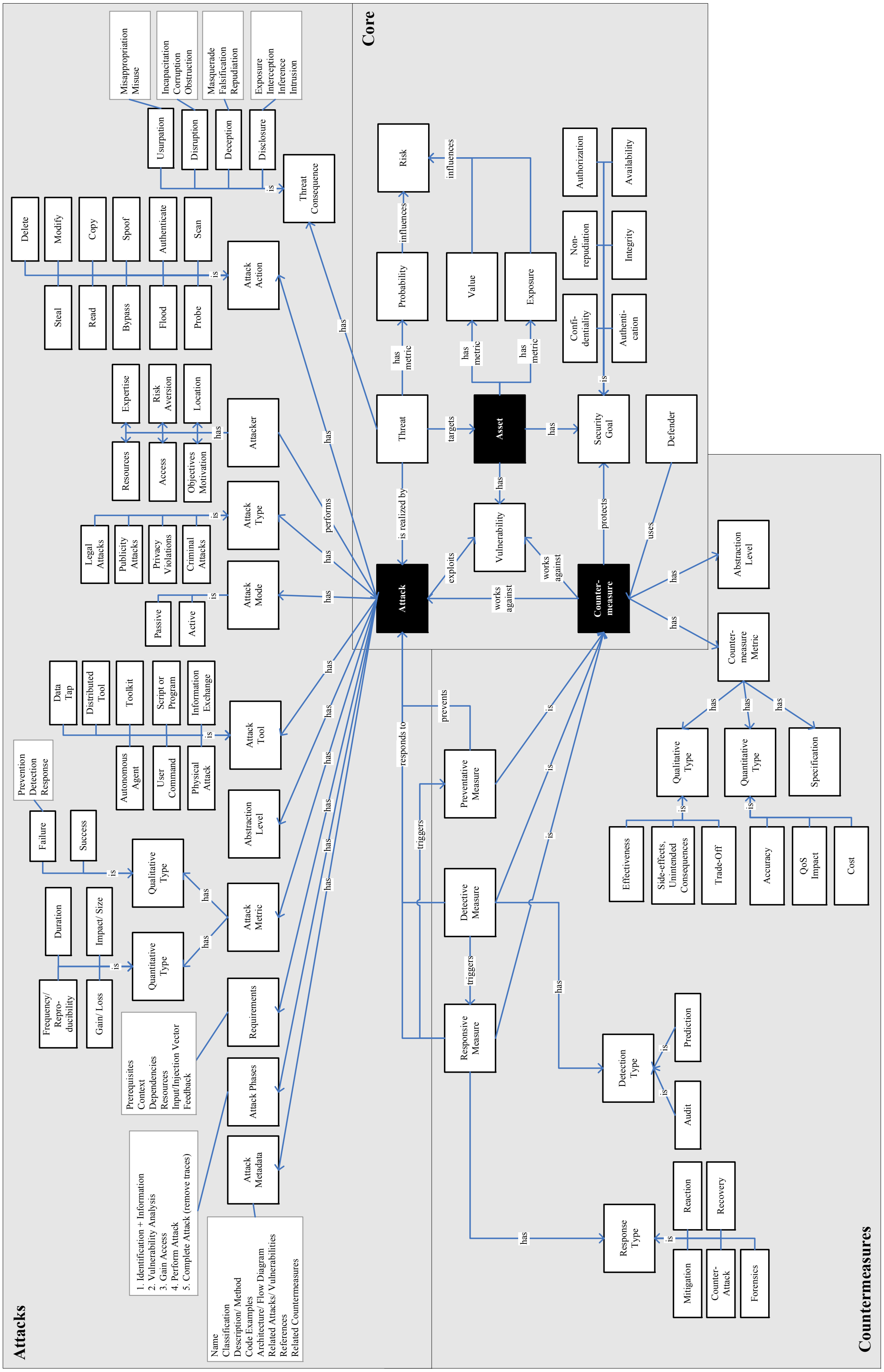

Figure 1. Metamodel for IT security (based on [1]–[4], [7]–[15])

data which is believed to be true), *Disruption* (preventing an asset from correct operation), *Usurpation* (losing control of the asset to an unauthorized entity). The *Probability* of a threat together with the *Exposure* of an asset and the *Value* its owner assigns to it determine the *Risk* of the asset. A *Vulnerability* is a point or characteristic of an asset which enables an attacker to bypass the security mechanisms of a system [1], [4].

Attacks realize threats and consist of intentional, unwarranted, non-authorized access (or attempt) to a system, which is not necessarily illegal [1], [8], [14]. To provide protection for the security goals, *Countermeasures* are used, these are separate and independent components [8]. Countermeasures can be divided into three general categories, i. e., how they relate to attacks [8]: preventative (e. g., encryption), detective (e. g., intrusion detection systems), and responsive (e. g., forensics). Good security systems make use of all three and how they build on each, e. g., detection triggers response, which then triggers additional prevention. Regarding the evaluation of countermeasures, both quantitative (e. g., how much they cost or what their Quality of Service penalty is) and qualitative metrics (e. g., whether they are effective or how they annoy users of the system) are used.

B. Attacks

This metamodel's goal is to illustrate the variety of concepts correlating to an attack in the field of IT security. For the sake of completeness, the metamodel tries to include all possible aspects of an attack in different contexts. Thus, when using the metamodel to represent a concrete attack there could be elements (or respectively sub-elements) of the metamodel that are not relevant for the specific attack under consideration. An attack can have the following elements associated with it:

ATTACK METADATA: Contains elements which facilitate the cataloguing of attacks and the understanding of how an attack is executed. Another aspect is to aid programmers in the reproduction and implementation of the respective attack. In this manner the attack could be used to test the security of a system with respect to the common security goals which are meant to be protected. Metadata of an attack consists of the following elements:

- Name and Classification: A unique, concise identification of the attack described [12].
- Description/ Method: A precise description of the attack, including all relevant details needed for understanding and execution [12]. Such details may be code/ implementation examples [11], data-input specification, (order of) execution steps. For a better visual representation of the attack, different diagrams such as communication, interaction, and control flow diagrams may be particularly helpful [10].
- Related Attacks/ Vulnerabilities: Each attack may be related to other attacks of the same type and may try to exploit a specific vulnerability which is similar to other known vulnerabilities. To better define an attack and understand the context and circumstances in which it is executed, such related concepts are needed.
- References: Publications which describe the attack or related information are listed here.
- Related Countermeasures: To get a better picture of the countermeasures that are possible against this attack it is useful that some of these are also described.

ATTACK PHASES: In order to execute an attack there is a series of phases which must be carried out. A good description of these phases contributes to the overall better understanding of the attack and its aspects. Such phases are described in the following, the first two phases combined lead to the more general concept of the *vulnerability discovery phase* while the remaining ones belong to the *exploitation* part of the attack [2], [11]:

1) Identification of vulnerabilities and overall information gathering.
2) Analysis of the found vulnerabilities.
3) Gaining access to the target software through a found vulnerability.
4) Performing the actual attack on the target.
5) Completing the attack, e. g., removal of traces.

REQUIREMENTS: For an attack to be executed there are different requirements that have to be met beforehand. Such requirements may be divided in the following categories:

- Prerequisites: A description of the necessary conditions, some special functionality, or characteristics of the target which must be present so that the attack can be successful [12].
- Context: A description of the contextual requirements of an attack. For example the period in which an attack can be executed.
- Dependencies: A description of dependencies the target system must have to other systems regarding attack phases or attack success must be taken into consideration before executing an attack.
- Resources: A catalogue of hardware and software which is needed for the successful execution of the attack as well as the know-how of the attacker [12].

ATTACK METRIC: Metrics are suitable for quantitative as well as for qualitative assessments. The following are proposals for different metrics that are useful for qualifying an attack in different ways after it has been executed. Possible subdivisions of the *Qualitative Metrics* are success and failure, i. e., due to prevention, detection, and/or response. Possible subdivisions for *Quantitative Metrics* are:

- Frequency/ Reproducibility: How often could this particular attack be executed so that there are good chances it would be a success for the attacker. Also it could measure how easy or difficult it is to perform the attack, e. g., hours of effort.

- Gain/ Loss: What does the attacker gain (quantitatively) or what could he lose if the attack fails. On the other hand it can be measured what the defenders or owners of the target system lose if the attack is successful or what possible gain could be possible for them, e. g., valuable information (cf. honeypots).
- Duration: Measures what the typical duration of performing the attack is.
- Impact/ Size: Describes the the size of the damage and the attack's impact on the target system. For example, the number of users of the targeted software affected by the attack. The idea for these metrics comes from the DREAD risk rating system [10].

ABSTRACTION LEVEL: IT systems can be attacked on different levels of abstraction, e. g., ranging from hardware-related attacks to social engineering on the business level. The underlying abstraction levels depend on the application area or the specific attack scenario and can also be used for classifying attacks according to their point of attack.

ATTACK TOOL: This describes the means used to exploit an asset's vulnerability, ranging from physical attacks over toolkits to user commands [15].

ATTACK MODE: According to the actions that an attacker takes during the execution of an attack, the latter could be labeled as *Passive* or *Active*. During a passive attack the attacker could eavesdrop on the traffic between two parties but does not interfere or alter anything. Whereas during an active attack the attacker could take actions to change the content of the information sent, disrupt communications or software processes and workflows [1].

ATTACK TYPE: As described in [2] there are different classes of attacks which could be executed on any kind of activity for various reasons: *Legal Attacks* (flaws in legislative systems), *Publicity Attacks* (receiving attention), *Privacy Violations* (data harvesting, espionage), *Criminal Attacks* (fraud, scams, destructive attacks, intellectual property theft, identity theft).

ATTACKER: In order to better understand an attack the resources, know-how, and other relevant information of the attacker should be described [2].

- Resources: Different entities could be marked as resources. Such could be financial resources of the attacker, software or hardware needed for the attack or additional specialists. Everything which facilitates an attack possible could be considered a resource.
- Expertise: The technical or soft-skills of the attacker which enable him to carry out a specific task or to start, control, or stop different processes are considered as expertise. If not available, expertise might be bought via resources.
- Access: Having specific kinds of information such as user credentials or a root-kit installed on a target machine define the extent of the access of an attacker to internal components and data.
- Risk Aversion: The amount of risk an attacker is willing to expose himself to and more importantly the measures taken to prevent any negative consequences for the attacker after an attack fall into this category.
- Objectives/ Motivation: These overlap to some extent with the element "Threat Consequences" from the model. An addition to these consequences could be, e. g., financial profit, political advantage, or simply a challenge for an attackers skills.
- Location: This is the physical location of the attacker or his organizational location, i. e., inside or outside.

ATTACK ACTION: This captures the activities performed by humans or machines to prepare, execute, and finish an attack. Examples for such actions are probe, scan, flood, authenticate, bypass, spoof, read, copy, steal, modify, delete [15].

C. Related Work

In this section we discuss different approaches for models and patterns which aim to define and describe the polymorphic security aspects of IT systems.

One of the current approaches for defining an attack on an IT system is presented in [11]: *Attack Patterns*. There, an attack pattern is defined as the blueprint for exploiting a software vulnerability whereas vulnerabilities could be a bug in the implementation or a flaw in the design of a software. The idea is continued in [12], where they use the notion of a *Design Pattern* in a slightly different context to define recurring ways of attacking and breaking software. The design pattern approach is used therefore in a destructive way rather than the constructive original idea. An attack pattern is described and considered as a general framework for executing an attack on a software system. Such an attack could be a particular method for exploiting a buffer overflow. In their paper, an attack pattern describes the approach used by attackers to generate an exploit against software.

A similar idea to attack patterns is presented in [10]. There, the approach for defining and describing attacks on software is regarded as *Threat Modeling*. Among other concepts, entry points (software modules or locations where data is transferred between applications), assets, and trust levels are discussed. A threat model thereby consists of elements which partially overlap with the elements of an Attack pattern as described in [12], i. e., ID, name, description, classification, mitigation.

Beside the pattern approach discussed so far there is also a taxonomy approach for defining and synthesizing security relevant aspects of attacking software. Such a taxonomy is described in [15]. It comprises the ideas of an incident consisting of an attacker description, tools used to execute the attack, vulnerability, action, target, unauthorized result, and objectives. Each of these concepts could be instantiated with a variety of proposed specific elements.

A different kind of approach focuses on the different possibilities to make a software secure and robust against attacks. Standard literature on the topic is, e. g., [16], introducing "Security Patterns" where the design pattern paradigm is applied in a constructive way with the specific goal of preventing attacks against software and summarizing knowledge on the topic in a concise and understandable way. The approach of Security Patterns however leaves out the actual attack and focuses only on preventive and possibly detective and reactive measures.

III. Application Scenarios (Attacks)

It this section, parts of the previously discussed meta-model for IT security are used to build attack models for different application scenarios in distributed systems: *Incorrect Lookup Routing* on Peer-to-Peer systems, *XML-Bombs* on Service-oriented Architectures, and *Black Hole Attacks* on Mobile ad hoc Networks.

We consider this an important step towards describing, communicating, and understanding attack knowledge. Of course, this is *not* to make future attacks easier and *not* to educate attackers, who already know about these things – as discussed by Anderson in the preface to his book [4]. The main cause is to lay the foundation for designing and evaluating countermeasures, which requires a solid understanding of what to protect against. Attack knowledge must not be left exclusively to attackers.

A. Peer-to-Peer Systems

Peer-to-Peer (P2P) systems are widespread nowadays and more than 50% of the Internet traffic is related to P2P [17]. The latest generation of P2P systems are the so-called distributed hashtables (DHTs). In these completely decentralized systems, an index of the content (data, services,. . .) that is available in the P2P system is maintained by the peers themselves. Depending on its ID, each peer is responsible for a specific part of the content (i. e., it knows where the content is hosted). To minimize the size and the complexity of the routing tables required for content lookup, each peer only maintains connections to a subset of other peers.While a DHT architecture enables scalable and efficient P2P services, security is a major issue due to the vulnerability of the DHT content lookup process.

Attack Metadata:

- Name: Incorrect Lookup Routing [18].
- Description: The misbehaving peer denies correct forwarding of received lookup requests. As a result, the provider of the requested content cannot be found. The misbehaving peer may route requests to arbitrary peers or simply drop them.
- Related Attacks/ Vulnerabilities: Partitioning [18].
- References: [18], [19].
- Related Countermeasures: Improve robustness of the lookup process [18] or distribute copies of the objects in the network [20].

Attack Phases:

- Identification and Vulnerabilities Analysis: Not necessary.
- Gain access: Sybil Attack or Incorrect Routing Updates [18] to become responsible for a large amount of content or specific content.
- Perform attack: Drop or redirect lookup request.
- Complete attack: Detection may be prevented by colluding misbehaving peers.

Requirements:

- Prerequisites: Attacker must have access to the P2P system.
- Context: The attack can only be performed upon receiving a lookup request.
- Dependencies: The misbehaving peer must be used to forward the lookup request for the particular content or must be directly responsible for it.
- Resources: Depending on the desired effect, from one up to an arbitrary number of misbehaving peers.

Attack Metric:

- Frequency/ Reproducibility: Depending on the frequency of received lookup requests. May be performed on each request or statistically.
- Gain/ Loss: If successful, the attack results in a Denial of Service on the attacked content. The chance of finding a specific object decreases with the number of misbehaving peers in the P2P system.
- Impact/ Size: Depending on the structure of the overlay (ring, tree,. . .) and on the number of connections maintained by each peer.

Abstraction Level: P2P overlay, application layer.
Attack Tool: Modified P2P client.
Attack Mode: Active.
Attack Type: Criminal attack (destructive attack).
Attacker:

- Resources: Access to the P2P system, i. e., a device with an Internet connection.
- Expertise: Advanced expertise in form of knowledge on the particular P2P overlay is required to implement an attack tool. Only basic knowledge on P2P systems required if tools are already available.
- Access: Not relevant.
- Risk Aversion: Low risk since detection is difficult.
- Objectives/ Motivation: Denial of Service.
- Location: Any.

Attack Action: Join the system with one or an arbitrary number of peers. Drop or modify incoming route requests. A collusion of misbehaving peers may amplify the effects.

B. Service-oriented Architectures

The paradigm of *Service-oriented Architectures* (SOA) [21] offers technological and organizational possibilities to improve flexibility and integration capabilities of IT architectures, i. e., in an enterprise setting. Web services are a successful implementation technology for SOA, an attack example from this field are so-called *XML-Bombs* [22], [23].

ATTACK METADATA:

- Name: XML-Bomb.
- Description: The attack pattern exploits the XML language which is a simple recursive language. In some situations parsers expand even small XML-messages to a multiple of their original size, thus, taking up memory and CPU time. This can lead to a system crash facilitating a Denial of Service attack. The methods used are recursive entity declarations where the reference entity expands to multiple other entities. Code examples, architecture, etc. are omitted here for space limitations, but are helpful for further information.
- Related Attacks/ Vulnerabilities: Oversized Payload [24], Coercive Parsing [24], Schema Poisoning [22].
- References: [22], [23].
- Related Countermeasures: Introduction of XML schema validation before the actual parsing and imposing strict XML-schemata as standard. However, this might be costly for the service provider from a Quality of Service standpoint, i. e., due to increased processing time.

ATTACK PHASES:

- Identification: A trial and error phase where services are tested for the existence of schema validation.
- Vulnerabilities Analysis: Schema validation is either existent or non-existent.
- Gain access: Not relevant.
- Perform attack: Send a poisoned message, the XML-Bomb.
- Complete attack: In order to leave no traces, spoofing may be executed on the TCP/IP level.

REQUIREMENTS:

- Prerequisites: The attacked application must have an XML parser which does not perform schema-validation before the parsing.
- Context: There are no specific context requirements for the attack. It can be executed, e. g., at any given point in time, platform-independently.
- Dependencies: There are no dependency requirements, in particular there is no dependency between the attacked service and any schema validation software.
- Resources: A computer with internet connection, if applicable, software for IP-spoofing.

ATTACK METRIC:

- Frequency/ Reproducibility: Any attack frequency is possible.
- Gain/ Loss: If successful, the attack may result in a Denial of Service on the attacked service. Gain of the attacker and respectively loss of the defender of the services is that the service is no longer available to the public (Disruption, Obstruction).
- Impact/ Size: All consumers are affected, until normal execution is restored they cannot use the service.

ABSTRACTION LEVEL: Application level, network level if, e. g., IP-spoofing is also executed.

ATTACK TOOL: Script, i. e., a manipulated XML file.

ATTACK MODE: Active, i. e., aiming at disturbing operation of the service provider.

ATTACK TYPE: Criminal attack (destructive attack), others are possible but depend on the attacker and the scenario.

ATTACKER:

- Resources: No or minimal financial resources required, a computer and a single attacker can perform the attack
- Expertise: Finding and exploiting this vulnerability does not require much skill. A novice with knowledge of the XML language can perform a successful attack.
- Access: Not relevant.
- Risk Aversion: Little risk if the attack is executed along with IP spoofing, the attacker could also claim that the actual message causing the problem is not an attack but a normal message which was malformed due to technical reasons (and without malicious intent).
- Objectives/ Motivation: Reduce availability of the target service. A possible scenario is, e. g., that the attacker provides a similar service of his own. The Goal could be that the consumers call the attacker's service instead of the targeted service.
- Location: Any.

ATTACK ACTION: Scan/ probe in order to determine the existence of schema validation, modification of the target service in form of the poisoned message, and possibly spoofing to obscure traces.

C. Mobile Ad hoc Networks

Wireless mobile ad hoc networks (MANET) facilitate establishing communication networks without infrastructure components [25]. To this end, the functionality of a MANET depends on the cooperation of the nodes involved. To enable communication of nodes that are not within each others transmission range, nodes located in between source and destination act as relays. This mode of operation makes MANETs susceptible to novel kinds of misbehavior from physical to application layer [26], [27]. The *Black Hole Attack* as a well-known example can be conducted with little effort, yet has a devastating effect on the functionality of the MANET.

ATTACK METADATA:

- Name: Black Hole Attack.
- Description: The misbehaving node redirects the routes from the destination intended to itself. After successful redirection, all communication is dropped instead of being relayed appropriately. Redirection of routes requires the injection of falsified routing information into a MANET. The particular steps necessary for

this are determined by the routing protocol deployed. Depending on the topology (physical and logical size, neighbors per node, etc.) of the MANET and the location of the attacker, already one black hole node can be sufficient to render a MANET inoperable.

- Related Attacks/ Vulnerabilities: Gray Hole, Wormhole.
- References: [26], [27].
- Related Countermeasures: Harnessing the promiscuous mode to overhear whether neighbored nodes forward packets correctly [28]. If not, misbehaving nodes can be excluded from routes.

ATTACK PHASES:

- Identification: Traffic analysis to determine the routing protocol deployed in the MANET.
- Vulnerabilities Analysis: Identify metrics used for selection of routes such as age and distance.
- Gain access: Not relevant.
- Perform attack: Inject falsified routing information for specific or arbitrary targets and drop packets.
- Complete attack: Forwarding with limited transmission power may be performed instead of dropping to avoid detection.

REQUIREMENTS:

- Prerequisites: Data transmission in the MANET must be based on unicast routes, not on broadcast flooding.
- Context: The attack has to be performed during the route discovery process. Yet, enforcing route discovery is possible and some protocols are also vulnerable after a route was established during route maintenance.
- Dependencies: For some routing protocols that perform an expanding ring search during route discovery limitations regarding the distance of the attacker from the source of a route exist.
- Resources: A device with a wireless network interface that supports packet injection. Detailed knowledge on MANET routing protocols is required to implement attack tools. Only basic knowledge is required to use available attack tools.

ATTACK METRIC:

- Frequency/ Reproducibility: The attack can be performed during each route discovery. Depending on the routing protocol, performing a black hole attack is possible also after a route was established successfully.
- Gain/ Loss: If successful, the attack results in a breakdown of network functionality (high loss rates).
- Impact/ Size: Depending on the routing protocol and network topology, already one black hole may cause a breakdown of the entire MANET.

ABSTRACTION LEVEL: Network layer/ routing protocol.

ATTACK TOOL: Packet injection, i. e., assembling packets that are sent by the wireless network interface without modification by the protocol stack of the operating system.

ATTACK MODE: Active.

ATTACK TYPE: Criminal (destructive). Privacy violations are possible inherently.

ATTACKER:

- Resources: Minimal financial resource required. Only a device with a wireless network interface supporting packet injection is needed.
- Expertise: Profound knowledge on MANET routing protocols for implementation of an attack tool. Basic knowledge to apply existing tools.
- Access and Location: Physical and logical access to the MANET is required, i. e., the attacker must be located within the area covered by the MANET.
- Risk Aversion: Depending on the scenario in which the MANET is deployed from low risk (e. g., at conferences, exhibitions) to high risk (e. g., police or military networks). Techniques such as sending with adaptive transmission power or directional antennas can be used to prevent detection.
- Objectives/ Motivation: Rendering the network itself unavailable.

ATTACK ACTION: Identify the routing protocol deployed, assemble and inject falsified routing information to redirect routes, drop network traffic.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a generic metamodel for IT security which brings together the most important concepts of IT security and their relationships. We have shown how at the *Core* of this model assets, threats, vulnerabilities, risks, security goals etc. relate to each other. Furthermore, the basic structure and concepts of *Countermeasures* were presented. Due to its significance and space constraints, the focus was set on *Attacks*, thus, laying the foundation for describing and understanding the different elements of attacks in an IT security context.

In order to show the applicability of the presented metamodel to real-life scenarios, three attacks on distributed systems were modeled: *Incorrect Lookup Routing* on Peer-to-Peer systems, *XML Bombs* on Service-oriented Architectures, and *Black Hole Attacks* on Mobile ad hoc Networks. These attack models provide both practitioners and researchers with a foundation to understand and discuss such attacks, enabling them eventually to design effective countermeasures.

Our next steps will be the extension of the metamodel in general and of the countermeasures in particular. Building on the knowledge about attacks and the already existing research on preventive, detective, and responsive measures will be a fruitful field to enhance the metamodel and check its validity. In this context it is important to strive for a balance between the completeness of the metamodel and keeping its complexity under control. We plan to make further advances and refinements available via a "living" technical report [29].

Acknowledgments

This work is partially supported by E-Finance Lab e. V., Frankfurt am Main, Germany (www.efinancelab.de), BearingPoint Management and Technology Consultants, and by CASED (www.cased.de). This work was partially funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 01 BK 0806, GLab). The authors alone are responsible for the content of the paper. Furthermore, the authors would like to thank the SOA security working group of TeleTrusT Deutschland e. V. for fruitful discussions.

References

[1] C. Eckert, *IT-Sicherheit: Konzepte – Verfahren – Protokolle*, 5th ed. Oldenbourg, 11 2007, (in German).

[2] B. Schneier, *Secrets and Lies: Digital Security in a Networked World*, 1st ed. Wiley, 1 2004.

[3] M. Bishop, *Computer Security: Art and Science*. Addison-Wesley Professional, 12 2002.

[4] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2nd ed. Wiley, 4 2008.

[5] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide, (Addison-Wesley Object Technology Series)*, 2nd ed. Addison-Wesley Professional, 5 2005.

[6] Object Management Group (OMG), "OMG Unified Modeling LanguageTM (OMG UML), Infrastructure," http://www.omg.org/spec/UML/2.2/, 2 2009.

[7] A. Jaquith, *Security Metrics: Replacing Fear, Uncertainty, and Doubt*, 1st ed. Addison-Wesley Professional, 4 2007.

[8] B. Schneier, *Beyond Fear: Thinking Sensibly About Security in an Uncertain World.* Springer, 5 2003.

[9] R. W. Shirey, "Security Architecture for Internet Protocols: A Guide for Protocol Designs and Standards," Internet Draft, 11 1994.

[10] F. Swiderski and W. Snyder, *Threat Modeling (Microsoft Professional)*. Microsoft Press, 7 2004.

[11] G. Hoglund and G. McGraw, *Exploiting Software: How to Break Code (Addison-Wesley Software Security Series)*. Addison-Wesley Professional, 2 2004.

[12] S. Barnum and A. Sethi, "Attack Patterns," "Build Security In" Initiative Website of the National Cyber Security Division of the U.S. Department of Homeland Security, 11 2006, https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/attack.html.

[13] A. P. Moore, R. J. Ellison, and R. C. Linger, "Attack Modeling for Information Security and Survivability," Carnegie Mellon University, Software Engineering Institute, Technical Note CMU/SEI-2001-TN-001, 2001.

[14] E. Amoroso, *Fundamentals of Computer Security Technology*. Prentice Hall PTR, 4 1994.

[15] J. D. Howard and T. A. Longstaff, "A Common Language for Computer Security Incidents," Sandia National Laboratories, Sandia Report SAND98-8667, 10 1998.

[16] M. Schumacher, *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications (Lecture Notes in Computer Science)*, 1st ed. Springer, 9 2003.

[17] R. Steinmetz and K. Wehrle, Eds., *Peer-to-Peer Systems and Applications (Lecture Notes in Computer Science)*, 1st ed. Springer, 10 2005.

[18] E. Sit and R. Morris, "Security Considerations for Peer-to-Peer Distributed Hash Tables," in *Peer-to-Peer Systems*. Springer, 2002.

[19] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, "Secure Routing for Structured Peer-to-Peer Overlay Networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 299–314, 2002.

[20] M. Sànchez-Artigas, P. G. López, and A. Skarmeta, "Making Replication Secure over Structured P2P Systems: Defending against Omission Attacks," in *5th Intl. Workshop on DBISP2P*, 2007.

[21] M. P. Papazoglou, "Service-oriented Computing: Concepts, Characteristics and Directions," in *Proceedings of WISE 2003*, 2003, pp. 3–12.

[22] Bundesamt für Sicherheit in der Informationstechnik, "SOA-Security-Kompendium: Sicherheit in Service-orientierten Architekturen," 2008, (in German).

[23] N. M. Josuttis, *SOA in Practice: The Art of Distributed System Design (Theory in Practice)*. O'Reilly Media, Inc., 8 2007.

[24] M. Jensen, N. Gruschka, R. Herkenhoner, and N. Luttenberger, "SOA and Web Services: New Technologies, New Standards - New Attacks," in *Proc. ECOWS*, 2007, pp. 35–44.

[25] Z. J. Haas, J. Deng, B. Liang, P. Papadimitratos, and S. Sajama, "Wireless Ad Hoc Networks," in *Encyclopedia of Telecommunications*, J. Proakis, Ed. John Wiley, 2002.

[26] D. Djenouri, L. Khelladi, and N. Badache, "A Survey of Security Issues in Mobile Ad Hoc and Sensor Networks," *IEEE Comm. Surveys & Tutorials*, vol. 7, pp. 2–28, 2005.

[27] B. Wu, J. Chen, J. Wu, and M. Cardei, "A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks," in *Wireless/Mobile Network Security*, ser. Network Theory and Applications, Y. Xiao, X. Shen, and D.-Z. Du, Eds. Springer, 2006, vol. 17, ch. 12.

[28] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*. New York, NY, USA: ACM Press, 2000, pp. 255–265.

[29] A. Miede, C. Gottron, A. König, N. Nedyalkov, N. Repp, and R. Steinmetz, "Cross-organizational Security in Distributed Systems," TU Darmstadt, Tech. Rep. KOM-TR-2009-01, 2009.