Attack Taxonomies and Ontologies

Natascha Abrek
Advisor: Nadine Herold
Seminar Future Internet SS2014
Chair for Network Architectures and Services
Department of Informatics, Technical University of Munich
Email: abrek@in.tum.de

ABSTRACT

In the past few years network security threats have increased significantly. Methods for attacks have not only grown in diversity but also became more sophisticated. The increased need for security mechanisms and countermeasures requires a comprehensive understanding of those attacks and their characteristics. To organize the knowledge of attacks a large variety of classifications were proposed in form of taxonomies and ontologies. The development of these classifications has emerged as an effective means for developing awareness systems and creating common descriptive languages. However, due to the high diversity of attacks no standard classification of network attacks exists so far. In this paper, a survey of existing attack taxonomies and ontologies is presented to create an overview of conducted work in this field of research. Furthermore, issues and drawbacks are discussed in a comparative analysis. The conducted survey has shown the need of a flexible, standardized classification of attacks and vulnerabilities to enable efficient knowledge sharing among the scientific community.

Keywords

Attack taxonomy, security ontology

1. INTRODUCTION

Latest publicized cyber-attacks against corporate and public organizations highlight the persistent threat against network security. The variety of methods to target personal, corporate or financial information has significantly increased and attacks became more sophisticated. New network vulnerabilities and attack possibilities were discovered by aggressors. New developments such as blended threats and information warfare techniques evolved. To protect from this wave of network threats robust countermeasures are necessary. However, the development of such security measures requires a comprehensive understanding of network attacks and their classifications. Taxonomies help to classify threats into well-defined categories. Bishop and Bailey [2] define a taxonomy as a system of classification which allows the unique identification of objects. Taxonomies help to organize knowlege and can serve as a helpful tool in the modeling process of system security and security policies. In the past, there have been numerous attempts to develop attack taxonomies [13, 18, 23, 27]. They range from general taxonomies to taxonomies which cover specific application domains or attack fields [12, 6]. Although numerous taxonomies have been introduced in the literature, no standard classification was developed so far. Research has shown several other drawbacks of taxonomies as well. The lack in consistency and extensibility makes them deficient in defining semantic relationships. In addition, the hierarchical order in taxonomies limit the possibilities of reuseability[26]. Therefore, the transition to ontologies is neccessary. Although both concepts are similar, the main difference is that an ontology complements the hierarchical order of a taxonomy with additional relationships. According to Gruber [11] an ontology is an explicit specification of a conceptualization. Ontologies represent powerful means to organize and represent knowledge in a structured and formal way. Additionally, they ease the process of communication and knowledge sharing [26]. Already several ontologies were developed in the area of network security. Still, also in the field of attack ontologies the development of a consistent ontology has not been accomplished so far. In this paper, a systematic survey of existing literature on attack taxonomies and ontologies is conducted. Thereby, two representing taxonomies and ontologies are selected and discussed. The selected papers cover research of network attacks in general. Classifications with aspect to specific fields were not considered. Furthermore, the focus lies on research conducted in recent years, reducing the selection to research papers published between 2012 and 2014. Following, a systematic analysis is carried out comparing the most relevant aspects. Goal of this work is to create an overview of conducted research in this field and to help researchers to take further steps towards a standardized classification of network attacks.

The remainder of this paper is organized as follows. In Section 2 characteristics of good taxonomies are discussed which also build the criteria for the following analysis. Furthermore, two selective taxonomies are presented. Section 3 covers the benefits of the transition from taxonomies to ontologies and the presentation of two existing security ontologies. In Section 4 an analysis is conducted discussing differences, advantages as well as disadvantages of the presented taxonomies and ontologies according to the defined criteria. Section 5 shows an overview of related literature surveying existing attack taxonomies and ontologies. Finally, in Section 6 the conclusion of this survey is presented.

2. ATTACK TAXONOMIES

In the field of network and computer security a great number of taxonomies classifying security threats and vulnerabilities were developed. In the following section, first the main characteristics of sufficient taxonomies are described. Then two selected attack taxonomies are presented in detail.

doi: 10.2313/NET-2015-03-1_01

2.1 Characteristics of a taxonomy

While computer and network attacks have become a consistent threat, the methods used to describe them are often inconsistent. In addition, the attack classification and detection represents a challenging task due to the highly increased number of threats during the years. That is why classification schemes such as taxonomies are pervasive means in the field of computer and network security engineering. The objective of a taxonomy is to provide a consistent instrument to classify attacks based on their characteristics. For an attack to be launched, security vulnerabilities are exploited. A vulnerability is a security exposure which results from flaws in a system or code. By providing an overview of attack characteristics such as vulnerabilities, a taxonomy can serve as a helpful tool in the security modeling process and in security assessment. Attack detection systems like intrustion detection systems can make use of taxonomies to identify a threat by the defined characteristics.

Before examining existing taxonomies, the main characteristics of a taxonomy have to be discussed. A taxonomy organizes classes in a hierarchical manner. The hierarchy is structured in multiple levels representing the depth of classification. The relationships between classes and subclasses are realized with the *is-a* relationship. In fact, this is the only relationship that can be drawn between classes in a taxonomy. Researchers have summerized a number of characteristics a taxonomy should satisfy in order to be sufficient. General requirements towards a taxonomy include the following:

Accepted: The structure of the taxonomy has to be intuitive and logical so that it can be easily understood and generally approved. It should build on previous, well-known research [15].
Comprehensible: The taxonomy should be understandable to both experts as well as those with less expertise. The concept has to be presented in a concise and clear form [19].
Determined: A clear definition and explanation for the developed classification is to be provided [17].
Exhaustive: A taxonomy is considered exhaustive or complete if all possibilities of attacks are accounted for [15].
Mutually exclusive: To achieve a mutual exclusive taxonomy every attack should be categorized into only one category. The developed categories must not overlap [15].
Repeatable: If the taxonomy is applied repeatedly, it has to result in the same classification [15].
Terms well defined: Only established security terminology should be used in the taxonomy. This is neccessary to avoid confusion and to build on previous, general knowledge [19].
Unambiguous: A precise definition of the categories is neccessary to prevent an ambiguous or unclear classification of an attack [15].
Useful: A taxonomy is useful when it is used to gain insights into a specific field of study [15].

According to Hansman [13] it is not possible or even neccessary for a taxonomy to fulfill all requirements at the same time. The degree on which a taxonomy aims to meet the requirements depends on the particular goal of the taxonomy. Authors have also identified a few more characteristics such as objectivity, appropriateness or primitivity [17, 1]. However, these characteristics are not taken into account in this survey since the presented taxonomies address only the above mentioned characteristics. The same characteristics will later on serve as criteria to conduct a comparative analysis between the presented taxonomies. In the following of this section, two selected attack taxonomies are presented. For a better understanding of the classification process with these taxonomies, they will be applied to a selected attack, the SQL slammer attack. The SQL slammer is a worm, which first appeared in 2003. It exploits a buffer overflow vulnerability in the Microsoft SQL Server. When the SQL server receives the request as a single large UDP packet the overrun in the server's buffer leads to the server overwriting its own stack with malicious code. Thereby, the worm code can then be executed. The worm then generates random IP addresses and send itself out to those addresses, allowing to spread rapidly to infect other hosts [7].

2.2 AVOIDIT

Simmons et al. [23] proposed in their paper a cyber-attack taxonomy called AVOIDIT. To classify an attack five classes were used: attack vector, operational impact, defense, informational impact and attack target. In their research they also address the issue of missing consideration of blended attacks in existing taxonomies. A blended attack is an attack which exploits different vulnerabilities at once [22]. So far, only little attention has been given to the possibility of blended attacks. Simmons et al. developed a tree structure for labeling attack vectors in their taxonomy. Their taxonomy is structured in five hierarchical levels. In the following, the classifiers of the first level are introduced. The complete taxonomy can be found in Figure 3 in the appendix of this paper.

Classification by attack vector: An attack vector describes the method or path by which an attacker reaches the target. This classifier defines the vulnerabilities of a system. The attack can use a single attack vector or a combination of several attack vectors. For example, an attacker can use the interaction with users to manipulate them in giving up their confidential information. Thus, social engineering is the way of performing an attack.
Classification by operational impact: This classifier includes the operational effects of an attack. Simmons et al. created a list containing mutual exclusive impacts. When an attacker successfully installs malware e.g. through a script or executable code (see *insufficient input validation* as attack vector), he can gain information about sensitive data.
Classification by informational impact: Besides operational impact the taxonomy also addresses informational impact. Informational impact contains potential ways to effect sensitive information through an attack. Possible impacts are distortion, disruption or disclosure of information.
Classification by attack target: The last classifier defines various attack targets. Possible instances are operating systems, networks, local computers or user information. An attack can also target a combination of instances. **Classification by defense:** The classification by defense contains numerous defense strategies which can be employed before or after an attack occurs. The defence strategies are subdivided in mitigation and remediation. Mitigation covers strategies to diminish damage before or during an attack.

Remediation involves procedures against existing vulnerabilities.

To better understand the process of classifying an attack with AVOIDIT we now demonstrate the usage with the SQL slammer attack. The SQL slammer is a worm launched via installed malware and spreads through the network (**Operational Impact**). It exploits misconfiguration, buffer overflow and denial of service vulnerabilities (**Attack Vector**). Primary targets are networks and applications (**Target**). Several damages can be caused when the worm is successfully installed (**Informational Impact**). It can change access to information (Disrupt), retrieve information (Discover) or modify data (Distort). Preventive and reactive methods are whitelists and patch systems (**Defense**).

2.3 Van Heerden's network attack taxonomy

Van Heerden et al. [27] developed an extensive taxonomy of computer network attacks using 12 classes, each containing multiple sub-classes. Their taxonomy consists of four hierarchical levels. Other than most taxonomies which cover attacks either from an attacker's or defender' point of view, van Heerden et al. included both views in their taxonomy. In the following, a description of the classes of the first level is given. The full taxonomy can be found in Figure 4 in the appendix.

Actor: The actor class describes the different entities which can execute an attack. Subclasses are commercial competitor, hacker, insider or protest groups.
Actor Location: The actor location refers to the country of origin of the attack. Attacks can be launched from local or foreign states. It is also possible that the specific location can not be determined or expand over multiple countries.
Aggressor: The aggressor represents the entity or group launching the attack. Aggressors can be individuals or groups, corporate entities or state aggressors. While the actor class describes the specific type of an attacker, the aggressor is an association with an actor.
Attack Goal: The attack goal specifies the attacker's objective. These can be the breach of security principles such as integrity or availability through changing, destroying or disrupting data. An attack can also work as a springboard for another attack.
Attack Mechanism: The attack mechanism defines the attack methodology. These can be access mechanism like hacking methods, e.g. brute force, phishing and buffer overflow. Data manipulation is another mechanism which uses data as an attack vector. They can be network based, e.g. denial of service or virus-based, e.g. trojans or worms. The collection of information for an attack is classified as information gathering.
Automation Level: This class describes the level of human interaction when launching an attack. A manual attack indicates that an attacker performs the methodology by hand. Automatic attacks only require a minimal amount of input by the attacker. Semi-automatic attacks are launched by tools which require user input.
Effects: Effects describe the severity of consequences caused by an attack. Minor effects are recoverable, whereas major effects are not. Effects are catastrophic when a target can no longer cease as an entity as a result of an attack. However, an attack does not neccessarily have to have an impact on a target. Then it is classified as null.

Table 1: Classification of the SQL Slammer with van Heerden's [27] taxonomy.

Attack Goal	**Attack Mechanism**	**Automation Level**	**Effect**
Disrupt, Change, Steal	Data-Manipulation: Virus-based: Worm	Automatic	Minor/ Major
Phase	**Scope**	**Target**	**Vulnerability**
Attack	Corporate, Governmental Network	Network, Software	Implementation: Buffer Overflow, Configuration: Default Setup

Motivation: The motivation for an attack differs from aggressor to aggressor. This class specifies incentives for an attack. A common motivation is the financial benefit. Other reasons are criminal or ethical aspects. An Aggressor can also launch attack simply for fun.
Phase: The phase class subdivides an attack into different stages. First, the attacker selects a target. Then the weaknesses of the target are identified. Finally, the attack is executed and post-attack activities are undertaken.
Scope: The scope determines the type of target and its size. Possible types are corporate, governmental or private networks. Corporate and governmetal targets can be subdivided into large or small networks.
Target: This class represents the physical entity targeted by the attack. Targets can be personal computers like laptops and tablets or network infrastructure devices like routers and switches. Servers are other possible targets of an attack.
Vulnerability: The vulnerability class describes the weaknesses exploited by an attacker. These can be deficient configurations regarding access rights or default setups or design issues in protocols or access control. Coding deficiencies are categorizied as implementation vulnerabilities.

The resulted classification of the SQL slammer using this taxonomy can be seen in Table 1. The colons represent the hierarchical structure through multiple subclasses. For example the class *Virus-based* is a subclass of the *Data-Manipulation* class and has itself the subclass *Worm.* Actor, Actor Location, Aggressor and Motivation are not listed in the table, since definite values are not available. The effects depend on the target and the severity of the attack. Therefore, effects can be of minor as well as major nature.

3. FROM TAXONOMIES TO ONTOLOGIES

Although taxonomies are useful means for classifications, they lack in several aspects. Taxonomies are often developed for specific domains which makes their extension as well as their consistency problematic. The reuse in other fields is often not possible. While taxonomies have mostly only hierarchical relationships, ontologies can also define custom semantic relationships. The formal and well-structured form of ontologies allow a better communication and reusability between organizations [26]. Additional advantages are named in [20]: Ontologies enable the seperation of domain knowledge from operational knowledge. The introduction of relationships provides the possibility to share knowledge with different fields. Ontology languages depict a common information representation and ease the process of information reuse.

According to Noy and McGuiness [20] an ontology consists of **concepts**, **attributes of classes** and **restrictions of slots**. The concept of a domain is described by classes. A class can have multiple subclasses which describe more specific concepts. Each class or subclass has instances. For example is *food* a superclass, *vegetable* and *fruit* subclasses and *apple* and *broccoli* are instances. The arrangement of the classes in a hierarchy builds the underlying taxonomy of the ontology. First-level classes are also referred as concepts. Attributes of classes are called slots. They describe behavioral and semantic properties of classes. Slots can therefore be described as relationships between classes. The class *human* for example can have the subclasses *woman* and *man*. Between those two subclasses a relationship can be defined, e.g. a man *is a husband* to a woman. Finally, ontologies need to define restrictions of slots, also called facets. Facets describe allowed values or types a slot can take. In the example above possible restrictions would be that a woman can have 0 or 1 husband, but not more.
The need of an ontology has been identified and there have been various attempts to create security ontologies [26, 14, 9]. In the following, two security ontologies are introduced in more detail.

3.1 Van Heerden's Ontology

In the previous section the taxonomy of van Heerden et al. was presented. This taxonomy is now used to create an ontology. The definition and arrangement of the classes are realized in their taxonomy. Furthermore, for their ontology they added an "Attack Scenario" class. This class is used to classify computer attacks and connects the other classes. It is subdivided in the classes denial of service, industrial espionage, web deface, spear phishing, password harvesting, snooping for secrets, financial theft, amassing computer resources, industrial sabotage and cyber warfare. Every attack scenario has a scope and goal. It consists of different attack phases and is assigned to an actor and an aggressor.
The next step in the ontology development process is to define the slots. Every class and subclass has a *is-a*-relationship. Classes can also have inter-relationships. Every *Actor* has at least one *Actor Location*. An *Aggressor* has always a motivation. An *Attack Mechanism* has exactly one *Target* and one *Automation Level*. A *Phase* has one *Effect* and a *Attack Mechanism*. A *Target* has a *Vulnerability* and a *Attack Scenario* has a *Attack Goal*, a *Phase* and at least one *Actor* and *Aggressor*. All relationships can be seen in Figure 1. A rectangle represents a class, the arrow the *has*-relationship between classes.

For the subclasses of the Attack Scenario class van Heerden et al. additionally defined attribute restrictions for their slots. With those restrictions attacks can be clearly seperated from each other. However, as they state themselves in their publication, their list of attack scenarios does not cover the full scope of possible attacks.

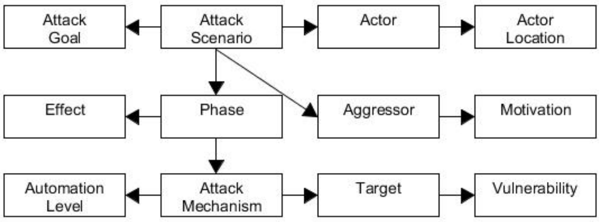


Figure 1: Van Heerden et al.'s [27] ontology

If the ontology is applied to the example of the SQL Slammer, following descriptions are defined: The Attack Scenario is SQL Slammer. It has the Phase attack. It has the Attack Mechanism worm. The SQL Slammer targets networks and software utilizing the vulnerabilities buffer overflow and default setup. Goal is to disrupt, change or steal data.

3.2 Ontology-based attack model

Gao et al. [10] developed an ontology-based attack model to assess the security of an information system from the angle of an attacker. Goal of the assessment process is the evaluation of attack effects. Thereby, the difference of system performance before and after an attack is calculated. The process consists of four phases. First, vulnerabilities of the system are identified using automated vulnerability tools. Such tools assess computer system, applications or network regarding their vulnerabilities and generate sets of scan results. In the second phase, the developed ontology is used to determine which attacks might occur due to the identified vulnerabilities. By quering the ontology, the possible effects are optained. This is the third phase. Finally, in the last phase the attack effect is calculated. In this paper a short overview of the classes is provided. For more detailed insight the reader is referred to their publication.

The ontology of [10] holds five classes: attack impact, attack vector, attack target, vulnerability and defense. **Attack Impact** consists of the security principles confidentiality, integrity, availability, authentication, authorization and auditing. All these principles are security properties of the target threatened by an attack. The **Attack Vector** describes here also the path by which am attack is launched. The **Target** class contains the possible targets hardware, software and humans. The **Vulnerability** addresses weaknesses and defects of the system. These can be for example design or implementation flaws. Finally, the **Defense** class describes countermeasures against attacks. The classes of their ontology show similarities to those used in the AVOIDIT taxonomy. Both adopted concepts from [13], [15] and [14].

Gao et al. [10] used relationships defined by Herzog [14] and extended his definitions with additional relationships. An attack *has* one or more attack vectors. It is *enabled by* a vulnerability. An attack *threatens* security properties defined in the attack impact. An attack vector *threatens* a target which *has* vulnerabilities. A target can also *reside* in another target. Defense strategies *protect* the target and the security properties. Finally, relationships between attack vectors are realized with the *ifSuccessfulLeadsToThreat*-relation. The ontology with all relations is shown in Figure 2.

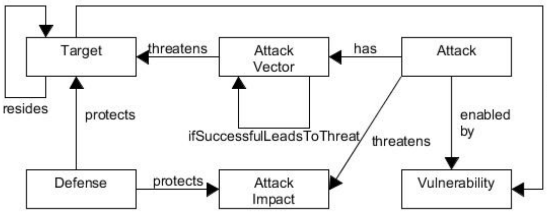


Figure 2: Gao et al.'s ontology [10]

Now the SQL Slammer is applied to the ontology. The SQL Slammer is a computer worm and *has* the attack vectors buffer overflow and denial of service. The attack *is enabled* by the vulnerabilities due to implementation flaws. Threatened targets are networks. If a Slammer attack succeeds he can cause further DoS attacks.

4. ANALYSIS AND RESULTS

After examining several classifications, a comparative analysis is conducted in this section. Thereby, the taxonomies and ontologies are compared with each other. Later on, the taxonomies are compared to the ontologies to determine the advantages of tansitioning from taxonomies to ontologies.

4.1 Taxonomies

The comparison of taxonomies is not a straightforward task. No general methods to compare attack taxonomies have been proposed so far. For the comparison of the presented taxonomies the criteria for developing successful taxonomies presented in Section 2.1 are applied.
First, a general comparison between the defined classes in the taxonomies of van Heerden et al.'s work and the AVOIDIT taxonomy is made.

Both taxonomies define a target class for possible targets. While van Heerden et al. provides a deeper hierarchical order of the target class using three subclasses, AVOIDIT uses a wider partion of targets with six subclasses. Van Heerden et al. provide moreover the scope class, which can be seen as an addition to the target class, giving more detail about the size and type of the target.
The classes attack goal in van Heerden et al.'s taxonomy and informational impact represent both the purpose of an attack. They share the same subclasses change, destroy and disrupt data. AVOIDIT provides beyond that two more subclasses disclosure and discover for acquiring information. Van Heerden et al. limits this to the single subclass steal data. The classes vulnerability and attack vector cover both security flaws and weaknesses that build the path to a successful attack. Van Heerden et al. provides excessive information about the attacker with additional classes, while AVOIDIT does not cover this aspect. This is due to fact, that in contrary to AVOIDIT, van Heerden et al.'s taxonomy not only addresses the defenders's point of view, but also the attackers's. Therefore, additional information about the attacker including location and motivation is neccessary. Furthermore, they provide additional information about the attack describing the different phases as well as the automation level of an attack. Other than van Heerden et al., AVOIDIT moreover provides defense techniques against attacks.

Table 2: Comparison of complied taxonomy requirements

Requirements	van Heerden et al.	Simmons et al.
Accepted	y	y
Comprehensible	y	y
Conforming	y	y
Determined	y	y
Exhaustive	n	n
Mutual Exclusive	y	y
Repeatable	y	y
Well Defined	n	y
Unambiguous	y	y
Useful	y	y

Now both taxonomies are evaluated against the criteria for a sufficient taxonomy. An overview of the comparison can be seen in Table 2. Van Heerden et al. state in their paper, that their taxonomy does not fulfill all criteria. Completeness could not be achieved due to the wide scope of existing attacks. Because their ontology uses a rather wide definition of network attacks instead of a detailed definition, also the requirement of well-defined terms was not achieved. According to the authors their developed taxonomy complies in the remaining requirements.

The AVOIDIT taxonomies meets all criteria for sufficient taxonomies according to their authors. However, since constantly new attacks and vulerabilities approach, their taxonomy is not considered exhaustive. Furthermore, the criteria for determinism is not mention in their publication. Since a detailed description about the development of their classes is provided, their taxonomy is considered determined.

Both authors name limitations of their developed taxonomy. Van Heerden et al.'s taxonomy does not cover all possible attack scenarios. AVOIDIT on the other hand, lacks in the amount of defense strategies. Both taxonomies do not discuss physical attacks.

4.2 Ontologies

In this section, the presented ontologies are analyzed. Thereby, first a general comparison is conducted analyzing differences and similarities between concepts, classes, slots and facets of each ontology. Based on the work of [5], we construct a table containting comparative metrics such as number of classes, average number of slots and average number of subclasses. Finally, we conclude by analyzing limitations and neccessary future work.

Ontologies consist of classes which are hierarchically ordered in a taxonomy. Van Heerden et al's taxonomy consists of overall 12 classes. In addition, for their ontology they added another class, the Attack Scenario, which makes their ontology consist of overall 13 classes. The ontology of Gao et al. contains six different classes.
Every taxonomy realizes the *is-a*-relationship between classes and subclasses. In an ontology, further relationships or slots can be defined. For their ontology, van Heerden et al. de-

Table 3: Comparison of general metrics between [27] and [10]

Metric	van Heerden et al.	Gao et al.
Number of concepts	13	6
Avg. number of subclasses/concept	3.6	8.2
Avg. depth of inheritance	2.8	2.2
Number of slots	10	9
Avg. number of slots/concept	1.8	2.6

fined the *has*-relationship to represent inter-relationships between classes. Gao et al.' taxonomy consists of a broader range of relationships.
To make the ontology complete van Heerden et al. define several restrictions of slots. These restrictions help to clearly distinct between attacks. Therefore, they define ten different attack scenarios with unique constraints. Gao et al. depict constraints for the three attacks SQL Slammer, Rootkit and the Mitnick attack.

Now the ontologies are compared using the metrics stated in [5]. The results are displayed in Table 3. The findings show, while van Heerden et al. use more concepts, Gao et al. have more subclasses per concept. The calculation of the average number of subclasses only includes subclasses until the second level. The average depth of inheritance describes the number of hierarchical levels for every concepts. Van Heerden et al. define their concepts in greater depth than Gao et al.. Regarding the relations between concepts, both define almost equal number of slots. However, Gao et al. have defined more slots per concept than van Heerden et al. in their ontology.

4.3 Taxonomies vs. Ontologies

So far, taxonomies have not directly been compared to ontologies. To determine the differences and similarities, the presented taxonomies and ontologies are compared with each other. Thereby, we compare the following aspects: purpose, usage, relationships and representation. The results will give further insights into the categorization process of an attack with the different concepts. Furthermore, we will conclude with advantages and disadvantages depending on the results of the comparison.

Purpose: Both taxonomies and ontologies follow the purpose to index attacks by classifying them by their characteristics. The AVOIDIT taxonomy is used to provide information regarding attack vectors, possible effects and defense strategies about an attack. Besides indexing attacks the ontologies describe a domain of knowledge. Van Heerden et al.'s taxonomy and ontology is supposed to clearly classify an attack from the view of the attacker and the target. As a future task they mention the refinement of their ontology to apply it for attack prediction. The main purpose of Gao et al.'s ontology is the evaluation of an attack effect.

Usage: The AVOIDIT taxonomy is applied to an issue resolution system (IRS). The IRS is a system which contains and manages a list of issues and countermeasures for those issues. It teaches the defender about potential risks of cyber attacks. The list is organized according to the taxonomy. Their taxonomy does not provide any information if an attack was successful, but classifies the attack vectors to foresee possible effects and identify appropriate defense strategies. Van Heerden et al. do not state any specific information on where their taxonomy and ontology is applied. In their future work they mention the usage of their ontology in intrusion detection systems. However, the determination of concepts such as motivation or attack goal by a computer system seems to be problematic. Gao et al. use their ontology in an ontology-based framework. The framework calculates the attack effect by comparing the system performance before and after the attack.

Relationships: Due to the hierarchical structure a taxonomy can only provide a parent-child-relationship. Ontologies, however, can not only describe a domain in a hierarchy but also define additional relations between classes and different concepts. These relationships are of a semantical or behavioral character. The AVOIDIT taxonomy for example allows the relations *is-a* between the target and its subclasses. The ontology by Gao et al. additionally adds the relation *resides* between different targets. This provides relationships between different concepts. Therefore, a taxonomy can be seen as a tree, whereas an ontology functions more like a web. This concludes that taxonomies are often restricted to the usage in a specific domain. Ontologies on the other hand allow the communication to other concepts and systems. This also points to another restriction of taxonomies, namely that knowledge is in most cases not hierarchical.

Representation: Taxonomies are mostly represented graphically in a tree-like structure. Ontologies can be represented either in a formal text format or graphically. Through machine interpretable definitions of the concepts computer applications are capable of interpreting the ontology. Gao et al.'s ontology is build using the language OWL. OWL is based on XML and is endorsed by the World Web Consortium (W3C). Van Heerden et al. make no further statements regarding the language they used for their ontology. Both presented taxonomies use a tree structure for their realization. The advantage of ontologies over taxonomies in this aspect is that the use of machine interpretable definitions makes reusability and knowledge sharing between different software systems easier.

The purpose of a taxonomy is to provide useful means to classify characteristics of attacks and thereby provide a better description of attacks. This classification helps to identify vulnerabilities, predict potential attacks and possible effects. Taxonomies are mostly used for risk management with identification, assessment and prioritization of risks as well as evaluation of systems. Taxonomies do not determine if an attack was successful. The AVOIDIT taxonomy is used in an issue resolution system. It classifies the attack vector information and foresees possible effects on the system. In summary, taxonomies are primarily used to represent security knowledge and determine defense mechanisms prior an attack.

Ontologies, unlike taxonomies, use semantic relations between attacks. Machine interpretable syntax allows comprehensive use in software systems such as Intrusion Detection Systems. Monitoring components collect data such as traffic, requests or packets and an alerting system provides response on the attempted attack and countermeasures. Gao et al. use their ontology for security assessment. Thereby, first vulnerabilities of the system are detected. Then possible attacks are queried. Based on the resulting attacks risks and neccessary defense methods are determined.

5. RELATED WORK

The use of taxonomies has become a key technique for the categorization and formal description of attacks. They reach from general attack taxonomies to specific field related taxonomies. Until today numerous surveys were conducted analyzing existing taxonomies to use them in defense methods against network attacks. Igure and Williams [16] conducted an extensive survey on cyber adversaries and attacks, discussing taxonomies from the early 1970s to 2006. By analyzing the efficiency of these taxonomies regarding the use in security assessment, Igure and Williams define requirements for taxonomies used in a security assessment process. Another extensive survey was presented by Meyers et al. [21]. In their paper, publications from 1985 to 2006 are covered. Further surveys were carried out in [29], [25] and [13].

Although many surveys were conducted on existing taxonomies, only few research was done regarding attack ontologies. Blanco et al. [3] carried out a systematic survey on existing security ontologies, evaluating and comparing concepts, relations and attributes using a framework. Souag et al. [24] conducted a general survey on existing security ontologies. Furthermore, the examined ontologies were analyzed regarding security aspects such as vulnerabilities, threats and countermeasures and evaluated for the use in security requirements engineering. Evesti et al. [8] examine a number of security ontologies, comparing their applicability for run-time security monitoring.

6. CONCLUSION

In this paper, a survey on existing attack taxonomies and ontologies was conducted. Furthermore, an analyis comparing differences between those concepts was carried out.

While many taxonomies for specific fields exist, there has been an increased attempt to develop a common, standardized attack taxonomy for the scientific community. However, depending on their goal and purpose the taxonomies still differ in their realization. Furthermore, the development of most examined work still resides in the early stages since they do not completely cover all attack possibilities. Therefore, the neccessity to combine existing taxonomies was identified.

Like research before has already shown [26] the limitations of attack taxonomies make the advancement to ontologies a neccessary task. The development of ontologies has been identified as an important branch of research. As a result of this work, it is concluded that the existing taxonomies and ontologies are not far enough developed for general usage and extension. Existing concepts need to be combined to create a flexible ontology that easily enables reuse and knowledge sharing between different systems.

7. REFERENCES

[1] E. G. Amoroso: *Fundamentals of Computer Security Technology* Prentice-Hall PTR, 1994

[2] M. Bishop, D. Bailey: *A critical analysis of vulnerability taxonomies*, California University Davis, Department of Computer Science, 1996

[3] C. Blanco, J. Lasheras, R. Valencia-Garcia, E. FernÃąndez-Medina, A. Toval, M. Piattini: *A systematic review and comparison of security ontologies*, In Availability, Reliability and Security, pages 813-820, ARES 08. Third International Conference on, 2008

[4] E. Blomqvist, A. Ohgren, K. Sandkuhl: *The analytic hierarchy process and multicriterion decision making* In: Enterprise Information Systems, pages 221-240, Springer Berlin Heidelberg, 2008

[5] E.Blomqvist, A. Ohgren, K. Sandkuhl: *Ontology Construction in anEnterprise Context: Comparing and Evaluating Two Approaches* In: Proceedings of the Eighth International Conference on Enterprise Information Systems: Databases and Information Systems Integration,Paphos, Cyprus, 2006

[6] K. F. P. Chan, M. Olivier, R.P. van Heerden: *A Taxonomy of Web Service Attacks*, In: Proceedings of the 8th International Conference on Information Warfare and Security: ICIW 2013, page 34, Academic Conferences Limited, 2013

[7] T. M. Chen, J. M. Robert(2004): *Worm epidemics in high-speed networks.* Computer, 37(6), pages 48-53, 2004

[8] A. Evesti, E. Ovaska, R. Savola: *From security modelling to run-time security monitoring*, Security in Model-Driven Architecture, page 33, 2009

[9] S. Fenz, A. Ekelhart: *Formalizing information security knowledge* In: Proceedings of the 4th international Symposium on information, Computer, and Communications Security, ACM, pages 183-194, 2009

[10] J. B. Gao, B. W. Zhang, X. H. Chen, Z. Luo: *Ontology-based model of network and computer attacks for security assessment* Journal of Shanghai Jiaotong University (Science), 18. Jg., pages 554-562, 2013

[11] T. R. Gruber: *A translation approach to portable ontology specifications*,Knowledge acquisition, 5. Jg., Nr. 2, pages 199-220, 1993

[12] N. Gruschka, M. Jensen: *Attack surfaces: A taxonomy for attacks on cloud services*, In: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, pages 276-279, 2010

[13] Hansman, Simon, and R. Hunt: *A taxonomy of network and computer attack methodologies*, Department of Computer Science and Software Engineering, University of Canterbury, 7, New Zealand, 2003

[14] A. Herzog, N. Shahmehri, C. Duma: *An ontology of information security* International Journal of Information Security and Privacy (IJISP), 1. Jg., Nr. 4, Spages. 1-23, 2007

[15] J. D. Howard: *An Analysis Of Security Incidents On The Internet 1989-1995* PhD thesis, Carnegie Mellon University, 1997.

[16] V. Igure, R. Williams: *Taxonomies of attacks and*

vulnerabilities in computer systems, Communications Surveys & Tutorials, IEEE, 10, 1, pages 6-19, 2008

[17] I. V. Krsul: *Software Vulnerability Analysis* PhD thesis, Purdue University, 1998

[18] C. E. Landwehr, A. R. Bull, , J. P. McDermott, W. S. Choi: *A taxonomy of computer program security flaws, with examples*, Naval Research Lab Washington DC, 1993

[19] U. Lindqvist, E. Jonsson: *How to Systematically Classify Computer Security Intrusions* IEEE Security and Privacy, pages 154âĂŞ163, 1997

[20] N. F. Noy, D. L. McGuinness: *Ontology development 101: A guide to creating your first ontology* Stanford University, Stanford, CA, 2001

[21] C. Meyers, S. Powers, D. Faissol: *Taxonomies of cyber adversaries and attacks: a survey of incidents and approaches*, Lawrence Livermore National Laboratory (April 2009), 7, 2009

[22] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver: *Inside the slammer worm*In IEEE Security and Privacy, volume 1, 2003.

[23] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, Q. Wu: *AVOIDIT: A cyber attack taxonomy*, Annual Symposium on Information Assurance, 2014

[24] A. Souag, C. Salinesi, I. Comyn-Wattiau: *Ontologies for security requirements: A literature survey and classification*, In: Advanced Information Systems Engineering Workshops. Springer Berlin Heidelberg, pages 61-69, 2012

[25] M. Uma, G. Padmavathi: *A Survey on Various Cyber Attacks and their Classification*, IJ Network Security, 15(5), pages 390-396, 2013

[26] J. Undercoffer, A. Joshi, J. Pinkston: *Modeling computer attacks: An ontology for intrusion detection*, In: Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, pages 113-135, 2003

[27] R. P. van Heerden, B. Irwin, I. D. Burke: *Classifying network attack scenarios using an Ontology*, In: Proceedings of the 7th International Conference on Information Warfare and Security. Academic Conferences Limited, pages 331-324, 2012

[28] L. G. Vargas, J.J. Doughe: *The analytic hierarchy process and multicriterion decision making* American Journal of Mathematical and Management Sciences, , 19(1), pages 59-92, 1982

[29] J. Wei: *Survey of network and computer attack taxonomy*, Proceedings of the 2012 IEEE Symposium on Robotics and Applications (ISRA), IEEE, USA, 2012

APPENDIX

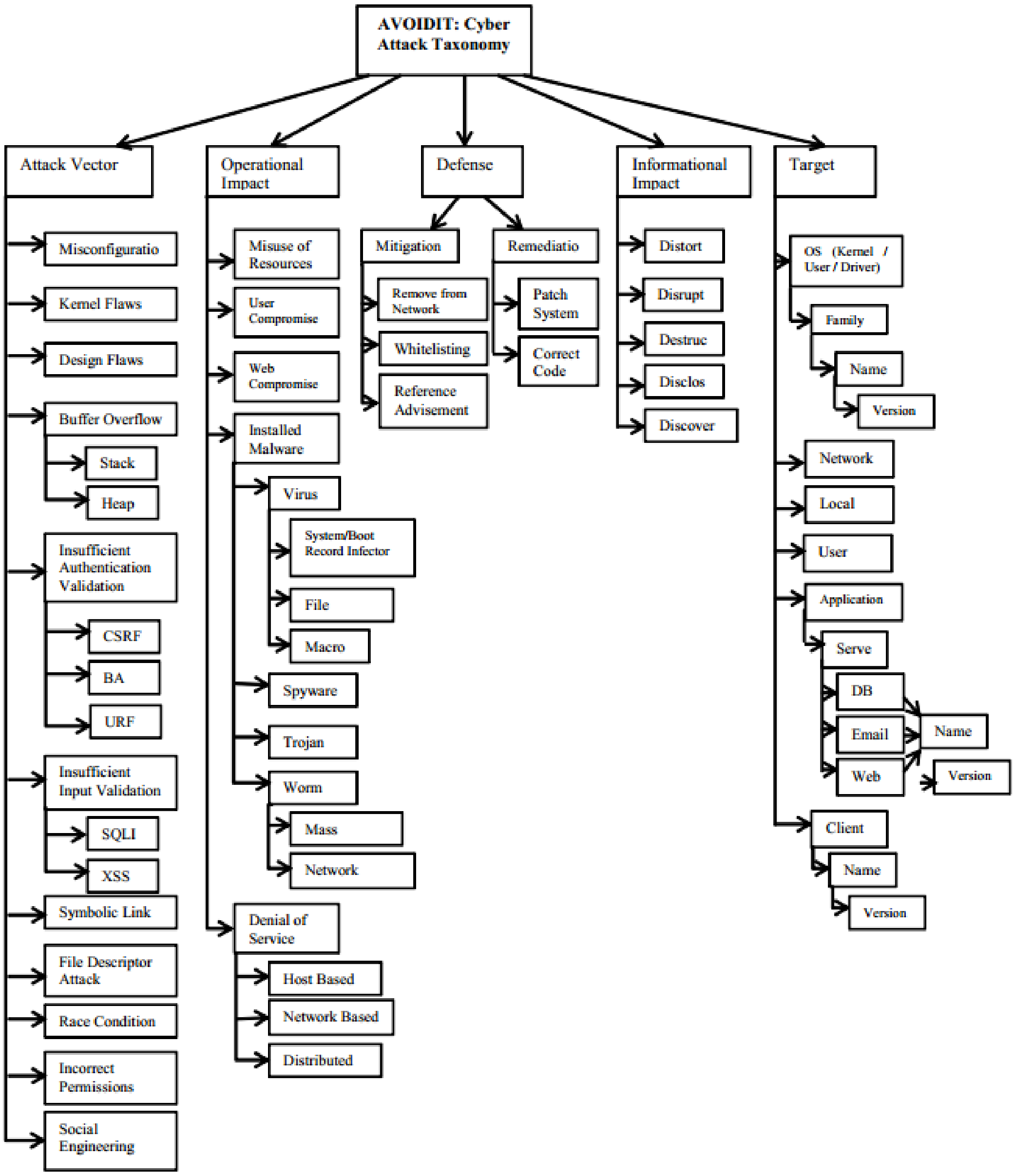
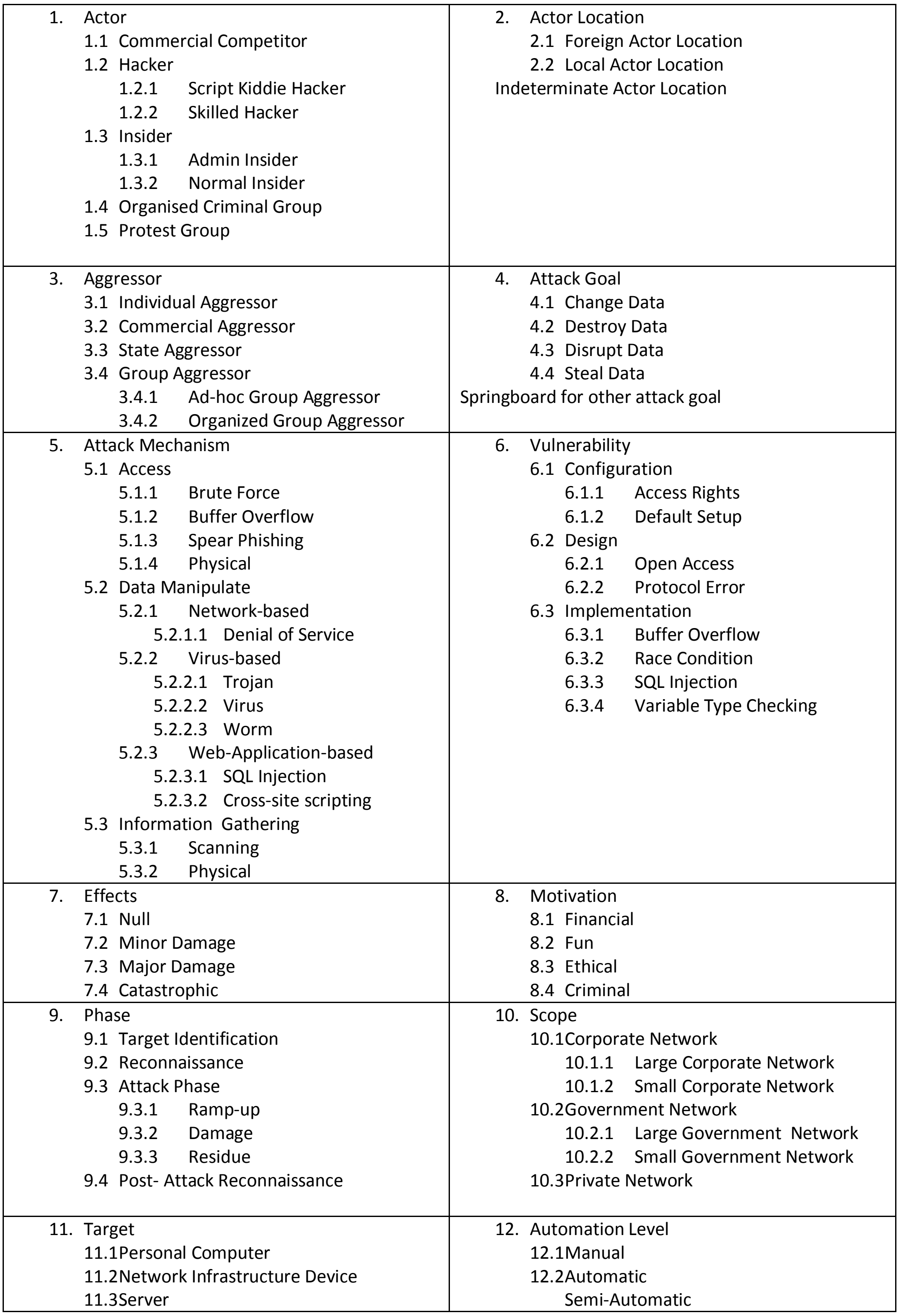


Figure 3: AVOIDIT taxonomy [23]



1. Actor 1.1 Commercial Competitor 1.2 Hacker 1.2.1 Script Kiddie Hacker 1.2.2 Skilled Hacker 1.3 Insider 1.3.1 Admin Insider 1.3.2 Normal Insider 1.4 Organised Criminal Group 1.5 Protest Group	2. Actor Location 2.1 Foreign Actor Location 2.2 Local Actor Location Indeterminate Actor Location
3. Aggressor 3.1 Individual Aggressor 3.2 Commercial Aggressor 3.3 State Aggressor 3.4 Group Aggressor 3.4.1 Ad-hoc Group Aggressor 3.4.2 Organized Group Aggressor	4. Attack Goal 4.1 Change Data 4.2 Destroy Data 4.3 Disrupt Data 4.4 Steal Data Springboard for other attack goal
5. Attack Mechanism 5.1 Access 5.1.1 Brute Force 5.1.2 Buffer Overflow 5.1.3 Spear Phishing 5.1.4 Physical 5.2 Data Manipulate 5.2.1 Network-based 5.2.1.1 Denial of Service 5.2.2 Virus-based 5.2.2.1 Trojan 5.2.2.2 Virus 5.2.2.3 Worm 5.2.3 Web-Application-based 5.2.3.1 SQL Injection 5.2.3.2 Cross-site scripting 5.3 Information Gathering 5.3.1 Scanning 5.3.2 Physical	6. Vulnerability 6.1 Configuration 6.1.1 Access Rights 6.1.2 Default Setup 6.2 Design 6.2.1 Open Access 6.2.2 Protocol Error 6.3 Implementation 6.3.1 Buffer Overflow 6.3.2 Race Condition 6.3.3 SQL Injection 6.3.4 Variable Type Checking
7. Effects 7.1 Null 7.2 Minor Damage 7.3 Major Damage 7.4 Catastrophic	8. Motivation 8.1 Financial 8.2 Fun 8.3 Ethical 8.4 Criminal
9. Phase 9.1 Target Identification 9.2 Reconnaissance 9.3 Attack Phase 9.3.1 Ramp-up 9.3.2 Damage 9.3.3 Residue 9.4 Post- Attack Reconnaissance	10. Scope 10.1Corporate Network 10.1.1 Large Corporate Network 10.1.2 Small Corporate Network 10.2Government Network 10.2.1 Large Government Network 10.2.2 Small Government Network 10.3Private Network
11. Target 11.1Personal Computer 11.2Network Infrastructure Device 11.3Server	12. Automation Level 12.1Manual 12.2Automatic Semi-Automatic

Figure 4: Van Heerden et al.'s taxonomy [27]