# Description of a Network Attack Ontology Presented Formally

**Renier van Heerden**

University of the Western Cape and CSIR, South Africa

**Louise Leenen**

University of the Western Cape and CAIR

**Barry Irwin**

Noroff School of Technology and Digital Media

**Abstract** The identification of network attacks in real-time is becoming increasingly important. Most Artificial Intelligence (AI) applications use machine learning to do the classification of attack types but the advantage of an ontological approach is that automated reasoning is the underpinning theory rather than automated learning. Automated reasoners allow automated classification and this powerful feature is the basis for the developing of an early warning system for active network attacks. In this paper, the authors describe how to employ Semantic Technologies by building an ontology to identify network attack types in order to support the automated classification of current network attacks by recognising relevant properties which are then mapped to relevant attack scenarios depicted in the ontology. The ontology engineering guidelines provided by Noy and McGuinness (2001) were used to build the ontology. The classes and relationships of the ontology are described formally and implemented in Protégé, an ontology editor. A core class in the ontology is the Attack Scenario class that represents different types of network attacks, for example, a Denial of Service attack. The ontology is evaluated by showing two examples of real attacks that correctly classified by the presented ontology. The presented ontology is to be expanded in future work. The aim of this paper is not to present a complete network attack ontology, but rather to present a proof of the concept of how to formally describe such an ontology, with the view to providing a baseline for future development of details. Row examples are explored to demonstrate how specific instances of attacks are classified using the ontology.

Key Words: Network, Attack, Taxonomy, Ontology

## Introduction

When there are indications that a network is being attacked, it is essential to be able to classify the type of attack quickly so that measures can be employed to counter the attack efficiently. Balepin et al. (2003) noted almost two decades ago that the increasing speed of computer attacks results in a need for quick responses that match and can contain these evolutionary attacks. It is becoming increasingly important to employ technologies that are able to identify relevant relationships in big volumes of data in almost real time. Machine learning is a popular approach to classify network attacks but Semantic Technologies provide an alternative approach to address this problem in the form of ontologies. An ontology can be described as a technology that allows the representation of a formal, shared knowledge base of the core concepts of a specific domain while providing a means to store the meaning of the concepts and the relationships used in describing the domain. Gruber (1993) defines an ontology as: “a specification of a representational vocabulary for a shared domain of discourse - definitions of classes, relations, functions, and other objects...". Noy and McGuinness (2001) states that “Classes are the focal point of ontologies, and can be divided into sub-classes which represent more detailed concepts." Automated reasoners are used in conjunction with ontologies to make inferences.

A number of researchers have created ontologies for network and information security but the use of ontologies to classify network attack types is still an emerging field. Velasco and Rodriguez (2017) provides a thorough overview of network and information security ontologies and indicate which of the ontologies include attacks as an aspect. In Section 2 of this paper, the authors list the related works and describe in which aspect of the presented ontology these works are relative.

The authors define and implement an ontology which contains a taxonomy of different types of attack scenarios in this paper. The main contribution of this work is the formal description of the classes and their relations within the ontology by using set theory notation. The motivation for these detailed descriptions is so that practitioners from different disciplines are able to understand the notation without the need to learn the formal languages commonly used to build ontologies. The authors have noted that the learning curve in Semantic Technologies is often a reason for hesitance to adopt these emerging technologies.

The authors' intent is for the presented network attack ontology to be further expanded and support a future system for automated identification of a network attack in progress. This should be done by recognising features of an ongoing attack that are similar to the attack scenarios described in the ontology. Automated reasoners have been used to do the classification of attacks in progress into identified attack types, and it thus necessary to describe the contents of the ontology formally for a correct implementation.

The motivation for developing an ontology is to provide a knowledge base for common understanding and structuring of relevant information, and the content can be shared with humans and computers. In an ontology domain knowledge is separated from operational knowledge. Ontologies allows the capability to attach meaning to the concepts and relationship that describe the domain. The formal languages in which ontologies are presented have well-defined semantics which can be employed by powerful reasoning tools. The reasoning ability of a mature network attack ontology will enable intelligent automated classification of a current network attack.

In Section 2, the taxonomy on which the presented ontology is based, is discussed. This is followed in Section 3 by description of the Network Attack ontology. Examples of two real attacks (also referred to as individuals in an ontology) are discussed in Section 4. Section 5 contains the conclusion and considers future work by discussing how this work can be applied to identify network attack scenarios in the future.

## Taxonomy

A synopsis of the taxonomy, on which our Network Attack ontology is based, is given in this section. The detail of this taxonomy is contained in van Heerden et al. (2012a). The authors' decisions on the modelling of the class hierarchy are based on their experience, their preferences and other network attack ontologies. Brief discussions to clarify these decisions are given throughout this section. Only the Actor, Actor Location and Aggressor sub-classes are shown in detail (Figures 1, 2 and 3). For more detail on the remaining sub-classes refer to van Heerden et al. (2012a). The taxonomy consists of the following base classes:

- The Actor class represents an individual or a group that is actively engaged in an attack (Figure 1). This class represents the individual or group that executes the attack, not the entity who instigated the attack. This class was derived based on the work of the following researchers: Simmonds et al. (2004); Rounds and Pendgraft (2009); Taylor (2001); Magklaras and Furnell (2001); Spitzner (2000).
- The Actor Location class represents the actual )physical) location from where an attack is/was performed (Figure2). This can be the country or state where the attack was initiated, and this class is based on the definitions of Undercoffer et al. (2004).
- The Aggressor class represents the individual of group that is the mastermind behind the attack. This entity can be the actor or the entity that instructs the actor to attack a network (Figure 3). For example, a number of authors suggested that China, France, Japan, Russia, Israel, Germany and South Korea engage in economic espionage via the Internet and employing computer network attacks (Burstein (2009); Brenner and Crescenzi (2006); Joyal (1996); Kshetri (2005); and Kim (2018)). These states are regarded as the Aggressor in the archives.

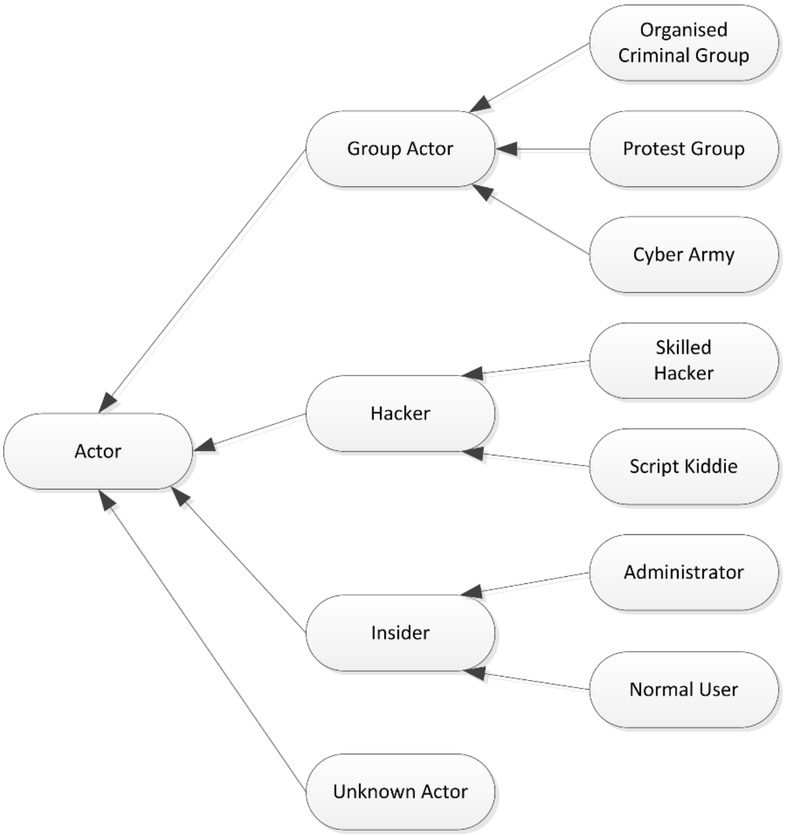

Figure 1 Actor Class

- The Asset class represents a non-personalised item that is being attacked; it allows the representation of different types of assets that can be attacked. Examples of such assets are information (stored data), a system, or network infrastructure.
- The Attack Goal class represents the objective that the Aggressor wants to achieve. The first few objectives correspond with the traditional CIA+ information security principles (Confidentiality, Availability, Integrity and Authentication). These objectives are similar to those mentioned by Simmonds et al. (2004) in his outcome class. The “Springboard for other attack” objective represents any instance where the network being attacked serves only as an intermediary node for attacks on another network.
- The Attack Mechanism class represents the approach (or the methodology that is used) of the attack. This class is related to vulnerability maps and attack vectors developed by the following researchers: Hansman (2003); Lee et al. (2003); Long (2007); Mookhey and Burghate (2004); Simmonds et al. (2004); Vasudevan and Yerraballi (2006).
- The Automation Level class represents the degree to which an attack can be automatically pre-programmed relative to the level of manual work that is required during the attack. The sub-classes are based on the taxonomy of Mirkovic and Reiher (2004).
- The Effect class represents the impact of an attack. Null means the target was not affected, Minor means the target can recover from the damage caused by the attack, and Major means the target cannot recover from the damage. Catastrophic means the damage is so severe that the target stops operating. An example of catastrophic damage is the declaration of bankruptcy. Similar classes were developed by Mirkovic and Reiher (2004).
- The Motivation class represents the incentive for the attack. Rounds and Pendgraft (2009); Gandhi et al. (2011); Pogrebna and Skilton (2019) developed similar motivation classes.
- The Phase class represents the temporal attack stages. These stages were identified by an evaluation of phases that appear commonly during attacks (Grant et al. (2007); Nachenberg (2012)).
- The Sabotage class represents the form of damage or type of loss that has been caused by an attack. Financial sabotage represents some monetary loss; Physical sabotage represents physical damage caused to hardware; and Virtual sabotage represents the loss of computer resources (examples are processing power, memory or bandwidth). The loss of Reputation is not a measurable and it is not tangible, but it may result in other further problems for a company in the future, and the damage is typically ongoing.

- The Scope class refers to the type of network that is being attacked. The type of networks are Corporate Network (networks that are controlled by private organisations), Government Network (networks that are controlled by government departments) and Private Network (a network that serves an individual in his/her private capacity).
- The Scope Size class represents the size of the network being attacked. Global Network represents the case when an attack affects a significant portion of the Internet or several countries. Large Network represents big corporations or large Government networks. It is difficult to precisely define the difference between a small, a medium or a large network, and the separation between these subclasses is subjective. The Single size present an attack on a single person or on a single computer.
- The Target class represents the physical devices that are targets in an attack, for example, Server, Desktop, Network Infrastructure or SCADA. This class is based on a similar class by Hansman (2003);Krebs (2009) methods to monetize value from Personal Computers.
- The Vulnerability class represents the weakness that is exploited in an attack. This class was based on a vulnerability map developed by Simmonds et al. (2004) and a vulnerability list from Undercoffer et al. (2004).
- The Attack Scenario class represents categories of different types of attacks. These scenarios were original presented and expanded by van Heerden et al. (2012a,b). The scenarios are: Denial of Service Scenario, Industrial Espionage, Web Defacement, Snooping for Secrets, Financial Theft, Amassing Computer Resources, Industrial Sabotage, Cyber Warfare and Runaway Malware.

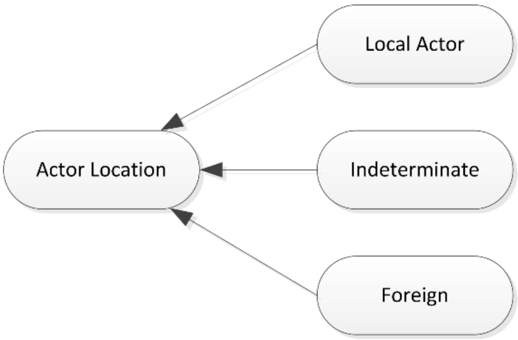

Figure 2 Actor Location Class

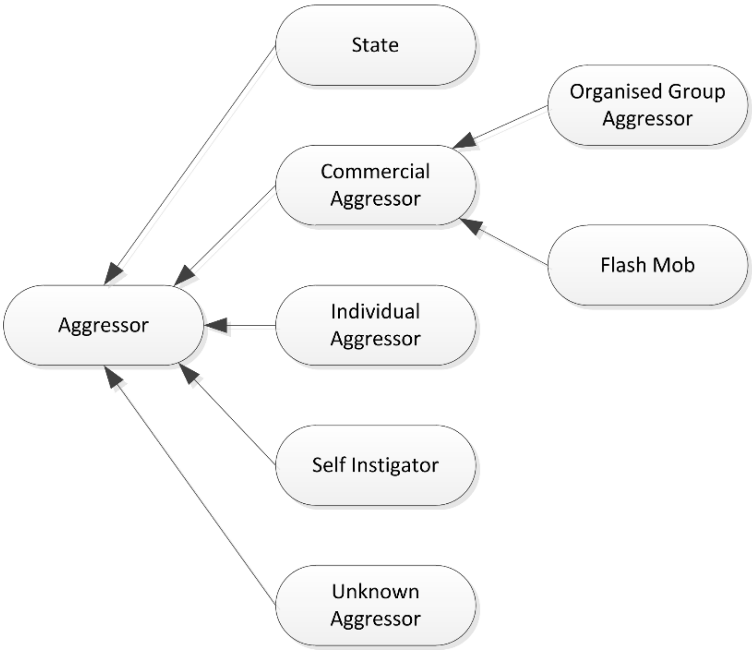

Figure 3 Aggressor Class

## Formal Description of Network Attack Ontology

In this Section each class and their relation in the ontology is mathematically defined. An ontology can be defined as a 4-tuple (Scharffe and de Bruijn (2005); Chaudhri et al. (1998); Zhai et al. (2009)):

O =< C, R, I, A > where

O is an ontology;

C is a set of concepts defined for the domain;

R is a set of binary semantic relations defined between concepts in C;

I is a set of Instances and where each instance can be one individual or one or more classes linked to by relations (Davies et al. (2006));

A is a set of axioms.

An axiom is a real fact or reasoning rule, while a concept is considered to be a class in an ontology. The definition also assumes there is an implicit assumption of a set, D, that represents the domain of interest. It follows that:

$$C \subseteq D \ (1)$$

$$R \subseteq D \times D \ (2)$$

The network ontology is defined in Statement 3:

$$NA \ =< \ CNA, RNA, INA, ANA \ > (3)$$

where NA defines an ontology related to network attack.

The set of concepts (or base classes), CNA, is described in Section 3.1. In Section 3.2 we define all the relations between the concepts, i.e. the set RNA. An example of an individual is discussed in Section 4. Axioms will be addressed in future work.

## Network Attack Concepts

The subsets of the set CNA are shown in Statement 4 and contains all the base classes of the taxonomy described in Section 2.

$$Actor, ActorLocation, Aggressor, Asset, AttackGoal, AttackMechanism,$$
$$AttackScenario, Automation\ Level, Effect, Motivation, Phase, Sabotage,$$
$$Scope, Scope\ Size, Target, Vulnerability \ \subseteq \ CNA \ (4)$$

The 16 subsets of CNA are defined in the following statements: 5, 9, 10, 12, 13, 14, 20, 21, 22, 23, 25, 27, 28, 29, 30 and 31. Some of these subsets are defined in more detail below. The class Actor and its sub-classes as displayed in Figure 1 are presented in statements 5 to 8.

$$GroupActor, HackerActor, InsiderActor, UnknownActor \ \subseteq \ Actor \ (5)$$

$$GroupActor, HackerActor, InsiderActor, UnknownActor \ \subseteq \ Actor \ (6)$$

$$ScriptKiddy, SkilledHacker \ \subseteq \ HackerActor \ (7)$$

$$Administrator, NormalU\ ser \ \subseteq \ Insider \ (8)$$

The class Actor Location and its sub-classes as displayed in Figure 2 are presented in statement 9.

$$Indeterminate, Local, Foreign \ \subseteq \ ActorLocation \ (9)$$

The class Aggressor and a sub-class Commercial are described in statements 10 and 11.

$$State, Commercial, Individual, SelfInstigator, Unknown \ \subseteq \ Aggressor \ (10)$$

$$FlashMob, OrganisedGroup \ \subseteq \ Commercial \ (11)$$

The classes Asset, AttackGoal and AttackMechanism are described in statement 12 to 14. Statements 15 to 19 give more detail regarding the sub-classes of Attack Mechanism.

$$Access, Data, Network, System \subseteq Asset \text{ (12)}$$

$$SpringboardforOtherAttack, StealData, Disrupt, ChangeData \subseteq AttackGoal \text{ (13)}$$

$$SpringboardforOtherAttack, StealData, Disrupt, ChangeData \subseteq AttackGoal \text{ (14)}$$

$$OpenInformation, Scanning \subseteq InformationGathering \text{ (15)}$$

$$SocialEngineering, SpearPhishing, BruteForce, BufferOverflow \subseteq AccessAttack \text{ (16)}$$

$$NetworkBased, Virus, WebApplication \subseteq DataManipulate \text{ (17)}$$

$$TrojanVirus, WormVirus, TraditionalVirus \subseteq Virus \text{ (18)}$$

$$CrossSiteScriptingWebApplication, SQLInjection \subseteq WebApplication \text{ (19)}$$

Statements 20 to 23 describe AttackScenario, AutomationLevel , Effect and Motivation. Statement 24 describe a sub-class of Motivation, namely Ethical.

$$\begin{gathered} DenialofService, IndustrialEspionage, \\ WebDefacement, SnoopingforSecrets, \\ FinancialTheft, AmassingComputerResources, \\ IndustrialSabotage, CyberWarfare, RunawayMalware \subseteq AttackScenario \text{ (20)} \end{gathered}$$

$$Automatic, SemiAutomatic, Automatic \; = \subseteq AutomationLevel \text{ (21)}$$

$$Null, Minor, Major, Catastrophic \subseteq Effect \text{ (22)}$$

$$Criminal, Financial, Fun, Ethical \subseteq Motivation \text{ (23)}$$

$$Espionage, Political, Vigilantism \subseteq Ethical \text{ (24)}$$

Statement 25 describes the Phase subset of CNA whilst Statement 26 gives more information on a subset of Phase, Attack.

$$TargetIdentification, Reconnaissance, Attack, PostAttack \subseteq Phase \text{ (25)}$$

$$RampUp, Damage, Residue \subseteq Attack \text{ (26)}$$

Statements 27 - 30 address the classes Sabotage, Scope, ScopeSize and Target.

$$\begin{gathered} OperationalLoss, FinancialLoss, PhysicalSabotage, ReputationalLoss, SecretLoss, Virtual \\ \subseteq Sabotage \text{ (27)} \end{gathered}$$

$$\begin{gathered} CriticalInformationInfrastructure, Corporate, Government, IndividualScope, Military, AllNetworks \\ \subseteq Scope \text{ (28)} \end{gathered}$$

$$Global, Large, Medium, Small, Single \subseteq \text{ScopeSize (29)}$$

$$NetworkInfrastructure, PC, IndustrialEquipment, Server \subseteq Target \text{ (30)}$$

The last subset of $C_{NA}$, Vulnerability, is described in statement 31 and its sub-classes in statements 32 - 34.

$$Config, Design, Implementation \subseteq Vulnerability \text{ (31)}$$

$$AccessRights, DefaultSetup \subseteq Config \text{ (32)}$$

$$OpenAccess, ProtocolError \subseteq Design \text{ (33)}$$

$$BufferOverflow, RaceCondition, EntryField, VariableTypeChecking \subseteq Implementation \text{ (34)}$$

## Relations

A major advantage of an ontology is that it can represent the meaning of concepts and relationships in a selected domain. A taxonomy is a hierarchical classification of concepts in the selected domain, but an ontology also includes the relationships linking the concepts. In this section, the authors describe the relationships between the classes in the ontology in the form of mathematical relations. Statement 35 defines the set RNA whilst statements 36 to 51 define the elements of RNA, i.e. the relations.

$$RNA = \{hasActor, hasActorLocation, hasAggressor, hasAsset,$$
$$hasAttackGoal, hasAttackMechanism, hasAutomationLevel,$$
$$hasEffect, hasMotivation, hasPhase, hasSabotage, hasScope,$$
$$hasScopeSize, hasTargethasVulner\grave{}ability, useVulnerability\} \text{ (35)}$$

$$hasActor \subseteq AttackScenario \times Actor \text{ (36)}$$

$$hasActorLocation \subseteq Actor \times ActorLocation \textit{ (37)}$$

$$hasAggressor \subseteq Actor \times Aggressor \textit{ (38)}$$

$$hasAsset \subseteq Target \times Asset \text{ (39)}$$

$$hasAttackGoal \subseteq Actor \times AttackGoal \text{ (40)}$$

$$hasAttackMechanism \subseteq AttackMechanism \times Phase \text{ (41)}$$

$$hasAutomationLevel \subseteq AutomationLevel \times AttackMechanism \text{ (42)}$$

$$hasEffect \subseteq Sabotage \times Effect \text{ (43)}$$

$$hasMotivation \subseteq Aggressor \times Motivation \text{ (44)}$$

$$hasPhase \subseteq Phase \times AttackScenario \text{ (45)}$$

$$hasSabotage \subseteq Asset \times Sabotage \text{ (46)}$$

$$hasScope \subseteq AttackScenario \times Scope \text{ (47)}$$

$$hasScopeSize \subseteq Scope \times ScopeSize \text{ (48)}$$

$$hasTarget \subseteq AttackScenario \times Target \text{ (49)}$$

$$hasVulnerability \subseteq Target \times Vulnerability \text{ (50)}$$

$$useVulnerability \subseteq AttackMechanism \times Vulnerability \text{ (51)}$$

## Constraints on Classes

In this section, the set AttackScenario (AS) is described (refer to Figure 4). The symbol ∃ is the first order existential quantifier: there exists at least one element. The symbol ∋ is used to express the words: such that. The symbol ∈ represents the classical set theory operator: element of. The symbol ∧ represents the logical operator: and. We now present (statement 52) a constrained definition of the set AttackScenario.

$$AS = \{x | (\exists z \in S\ cope \ni (x,z) \in hasS\ cope) \wedge (\exists v \in Actor \ni (x,v) \in hasActor) \wedge (\exists w \in Phase \ni (x,w) \in hasPhase) \wedge (\exists u \in T\ arget \ni (x,u) \in hasT\ arget)\} \text{ (52)}$$

In Statement 52, we further constrain the Attack Scenario set with every element x of the set AS , as depicted in Figure 5.

- there exists at least one element z which is a member of the set S cope, and is such that the ordered pair (x, z) participates in the relation hasS cope; and
- there exists at least one element v which is a member of the set Actor, and is such that the ordered pair (x, v) participates in the relation hasActor; and
- there exists at least one element w which is a member of the set Phase, and is such that the ordered pair (x, w) participates in the relation hasPhase; and
- there exists at least one element u which is a member of the set Target, and is such that the ordered pair (x, u) participates in the relation hasTarget.

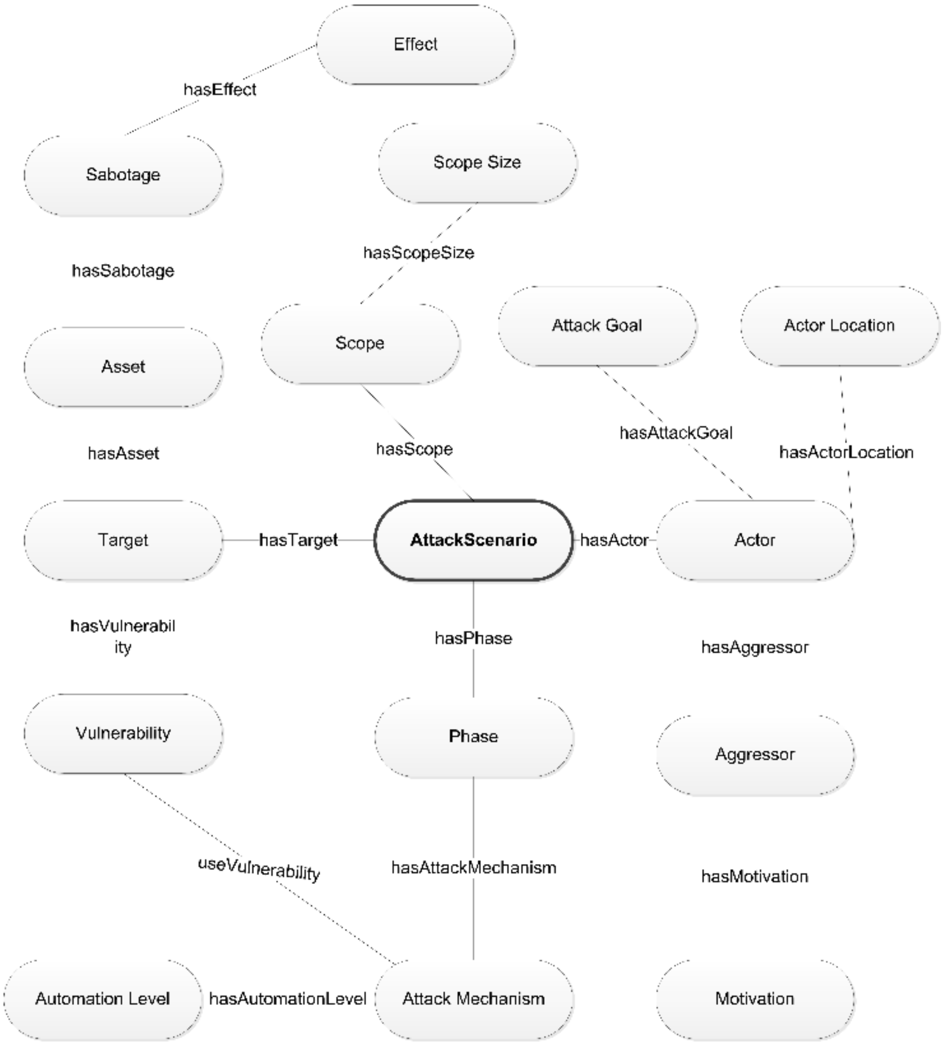

Figure 4 Attack Scenario

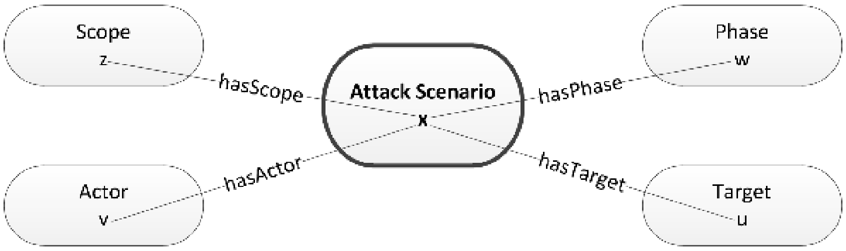

Figure 5 Illustration of Statement 52

Similarly, we define constrain the sets S cope, Actor, Aggressor, Phase, AttackMechanism, Target, Asset and Sabotage.

$$Scope = \{x \mid \exists y \in ScopeSize \ni (x,y) \in hasScopeSize\} \text{ (53)}$$

$$Actor = \{x \mid (\exists z \in AttackGoal \ni (x,z) \in hasAttackGoal) \wedge \exists v \in ActorLocation \ni (x,v) \in hasActorLocation) \wedge (\exists w \in Aggressor \ni (x,w) \in hasAggressor)\} \text{ (54)}$$

$$Aggressor = \{x \mid \exists y \in Motivation \ni (x,y) \in hasMotivation\} \text{ (55)}$$

$$Phase = \{x \mid \exists y \in AttackMechanism \ni (x,y) \in hasAttackMechanism\} \text{ (56)}$$

$$AttackMechanism = \{x \mid (\exists z \in AutomationLevel \ni (x,z) \in hasAutomationLevel) \wedge (\exists y \in Vulnerability \ni (x,y) \in useVulnerability)\} \text{ (57)}$$

$$Target = \{x | (\exists z \in Asset \ni (x,z) \in hasAsset) \wedge (\exists y \in Vulnerability \ni (x,y) \in hasVulnerability)\} \quad (58)$$

$$\mathrm{Asset} = \{\mathrm{x} | \exists \mathrm{y} \in \mathrm{Sabotage} \ni (\mathrm{x},\mathrm{y}) \in \mathrm{hasSabotage}\} \quad (59)$$

$$Sabotage = \{x | \exists y \in Effect \ni (x,y) \in hasEffect\} \quad (60)$$

## Denial of Service Scenario

In this section, we discuss one specific type of network attack, an element of the Attack Scenario class, namely a Denial of Service (DoS) attack in more detail. The aim of a DoS attack is to negatively affect the legitimate use of a computer network (Houle and Weaver (2001), Wang et al. (2018)). One of the most frequent methods that DoS attacks use, is to flood a single network node with network traffic. This flood of traffic will impair normal network operations. Distributed Denial of Service (DDoS) attacks disrupt networks by flooding it with traffic from a large number of different sources.

The DenialofService (DoS) scenario set is defined in Statements 61 to 76 (also refer to Figure 6). In Figure 6, the sub-classes that are specific to the DenialofService scenario are displayed in light grey. This demonstrates which sub-classes are used when the Denial of Service attack scenario is resented. For example, only the OperationalLoss sub-class is used from Sabotage class.

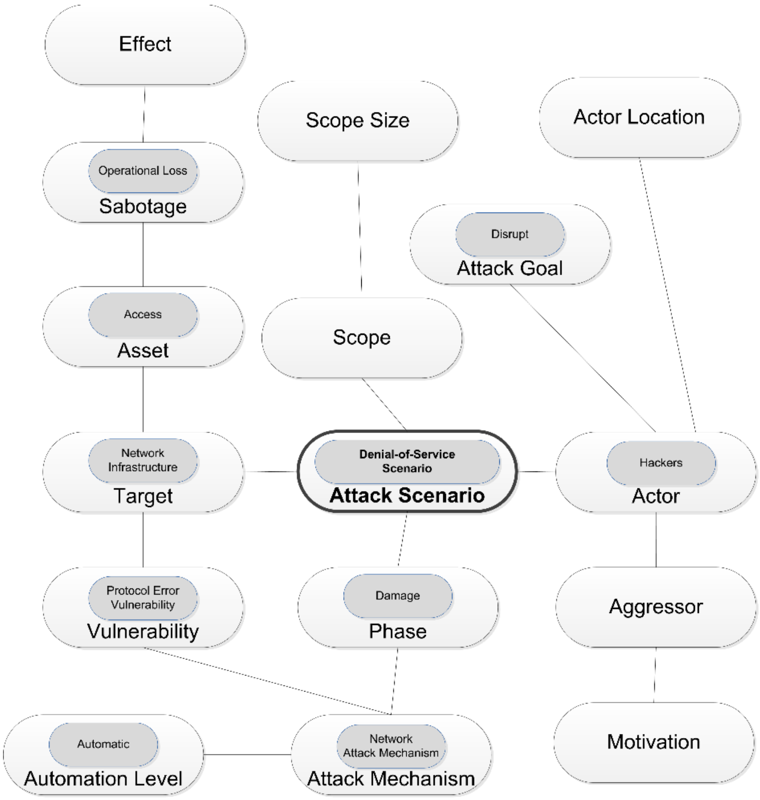

Figure 6 Denial of Service Attack Scenario

$$DoS \subseteq AttackScenario \quad (61)$$

$$DoS = \{x | (\exists v \in Hacker \ni (x,v) \in hasActor) \wedge (\exists w \in Damage \ni (x,w) \in hasPhase) \wedge (\exists u \in NetworkInfrastructure \ni (x,u) \in hasTarget)\} \quad (62)$$

$$HackerDoS \subseteq Hacker \subseteq Actor \quad (63)$$

$$DamageDoS \subseteq Damage \subseteq Phase \quad (64)$$

$$NetworkInfrastructureDoS \subseteq NetworkInfrastructure \subseteq Target \quad (65)$$

$$Hacker = \{x | \exists z \in Disrupt \ni (x,z) \in hasAttackGoal\} \ (66)$$

$$DisruptDoS \subseteq Disrupt \subseteq AttackGoal \ (67)$$

$$Damage = \{x | \exists y \in NetworkAttackMechanism \ni (x,y) \in hasAttackMechanism\}(68)$$

$$NetworkAttackMechanismDoS \subseteq NetworkAttackMechanism \subseteq AttackMechanism \ (69)$$

$$NetworkAttackMechanism = \{x | (\exists z \in Automati \ni (x,z) \in hasAutomationLevel) \wedge (\exists y \in ProtocolError \ni (x,y) \in useVulnerability) \ (70)$$

$$AutomaticDoS \subseteq Automatic \subseteq AutomationLevel \ (71)$$

$$\mathrm{ProtocolErrorDoS} \subseteq \mathrm{ProtocolError} \subseteq \mathrm{Vulnerability} \ (72)$$

$$NetworkInfrastructure = \{x | (\exists z \in Access \ni (x,z) \in hasAsset) \wedge (\exists y \in ProtocolError \ni (x,y) \in hasVulnerability)\} \ (73)$$

$$AccessDoS \subseteq Access \subseteq Asset \ (74)$$

$$Access = \{x | \exists y \in OperationalLoss \ni (x,y) \in hasS\ abotage \ (75)$$

$$OperationalLossDoS \subseteq OperationalLoss \subseteq Sabotage \ (76)$$

## Attack Scenario Examples: the SCO Attack and SpamHaus

In this section, we explore where a specific instance of a network attack (or individual) can belong within the Network Attack ontology. When the network attack ontology is available online as part of an early warning system, these classifications will be done automatically and in near real time.

In May 2003, a commercial Unix distributer SCO was attacked via a Distributed Denial-of-Service (DDoS) attack (Shankland (2003)). This attack used a number of different computers to make requests for connecting to the SCO web server at the same time. The SCO web server was not able to respond in time to all the requests for connections. The result was that the SCO web presence was impaired during the attack. A SCO representative confirmed that they did not know who was responsible for the attack and that nobody admitted to be responsible for the attack.

In a similar attack in December 2003, Moore and Shannon (2003) claimed that the SCO servers had to respond to more than 700 million attack packets over a period of 32 hours. This information was based on observations by the UCSD Network Telescope. The motive of the attack on the SCO web server was suspected to be anger at the SCO legal action case against IBM, regarding possible copyright of Linux code.

On 16 March 2013, a DDoS attacked was launched on the SpamHaus website (Hanford (2013)). The attack reached a flow high enough to threatened the core infrastructure of the Internet (Leyden (2013)). A Dutch hosting company, CyberBunker, is rumoured to be responsible for the attack to retaliate for being listed on the SpamHaus anti-spam list but the company denied this (Markoff and Perlroth (2013)). The two attacks, on SCO in 2003 and SpamHaus in 2013, employed the same approach, but differed in scale. In the decade between the attacks, the amount of data required to launch a successful DDoS attack has grown considerably.

These instances of Denial of Service attacks are shown in Figures 7. In these figures the Denial of Service Scenario sub-classes are shaded light grey. The sub-classes that are not required for the Denial of Service Scenario are shaded medium grey. The individuals are shaded dark grey. For the SCO individual the following two statements hold:

$$SCONetwork \in CorporateNetwork$$

$$SCOAttack \in DoSScenario \ (77)$$

For the SpamHaus individual the four following statements hold:

$$Netherlands \in ActorLocation$$

$$CyberBunker \in Aggressor$$

$$SpamHausNetwork \in CorporateNetwork$$

$$SpamHausAttack \in DoSScenario \ (78)$$

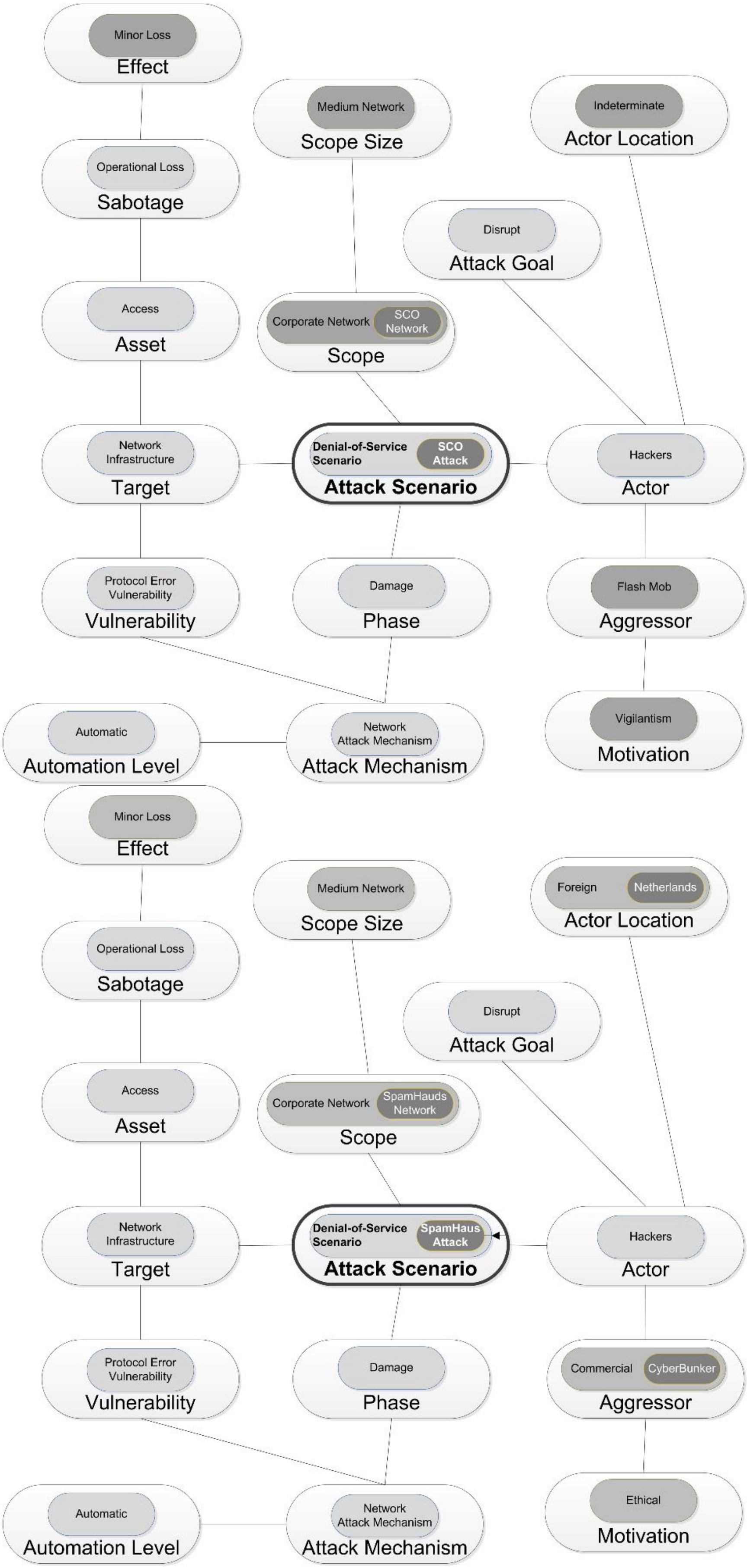

Figure 7 SCO Attack  and SpamHaus Individuals

For the SCO individual the aggressor was Flash Mob with Vigilantism motivation, and the scope is a Corporate Network of Medium size. The Actor Location was Indeterminate and the effect is Minor. For the Spamhaus individual the aggressor is Commercial (CyberBunker) with Ethical motivation. The scope of the attack was a Corporate Network of Medium size similar to SCO network. The Actor location was Foreign (Netherlands). The SCO and Spamhouse examples' remaining classes were within the Denial-of-Service Attack Scenario definition as shown in Figure 6.

## Conclusions and Future Work

In this paper we present an ontology for describing attacks executed against a computer network. The goal of the formal description is to enable future automated classification of network attacks. This ontology is based on a taxonomy previously published. Our ontology is intended to support the automated classification of ongoing network attacks in the future when the ontology is sufficiently mature. We formally describe the classes and relations in the ontology using set theory notation, and implemented it in Protégé, an ontology editor. The Denial-of-Service Attack Scenario is also described in detail. All the other mentioned Attack Scenarios can thus similarly be described. We show how examples of specific network attacks that occurred, such as the SCO attack, can be added as individuals to the ontology and then be correctly classified by the ontology an elements of Denial-of-Service Attack Scenario class.

Network sensors such as Intrusion Detection Systems (IDS) and Honeypots can be mapped into the ontology. A network sensor is any application or system that provide information about the network status. A sensor can be directly related to network attacks, such as an IDS or indirectly related such as a Network Telescope. Applications that provide abstract network information, such as bandwidth monitoring systems, can also act as a network sensors. By mapping the sensors, and what they measure, a determination of which scenarios can be measured by which sensors can be made. Some scenarios may even be proven to be un-measurable due to lack of a suitable sensor to directly detect their presence.

The final goal of this work is to optimally establish what needs to be measured to determine if a network is in the initial stages of an attack, and what type of attack is being launched. This knowledge will allow for the earliest possible response plan to be put in place to remediate, or contain the event.

## References

Balepin, I., Maltsev, S., Rowe, J. & Levitt, K. (2003). Using specification-based intrusion detection for automated response. In Recent Advances in Intrusion Detection (pp. 136-154). Springer.

Brenner, S.W. & Crescenzi, A.C. (2006). State-Sponsored Crime: The Futility of the Economic Espionage Act. Houston Journal of International Law, 28, 389.

Burstein, A.J. (2009). Trade Secrecy as an Instrument of National Security-Rethinking the Foundations of Economic Espionage. Arizona State Law Journal, 41, 933-1167.

Chaudhri, V.K., Farquhar, A., Fikes, R., Karp, P.D. et al. (1998). OKBC: A programmatic foundation for knowledge base interoperability. In Proceedings of the National Conference on Artificial Intelligence (pp. 600-607).

Davies, J., Studer, R. & Warren, P. (2006). Semantic web technologies (pp. 4,118-). John Wiley and Sons.

Gandhi, R., Sharma, A., Mahoney, W., Sousan, W. et al. (2011). Dimensions of Cyber-Attacks: Cultural, Social, Economic, and Political. Technology and Society Magazine, IEEE, 30(1), 28-38.

Grant, T., Venter, H. & Eloff, J. (2007). Simulating adversarial interactions between intruders and system administrators using OODA-RR. In Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT (SAICSIT) Research in Developing Countries (pp. 46-55). ACM.

Gruber, T.R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199-220.

Hanford, S. (2013). Chronology of a DDoS: SpamHaus. Cisco. Online. Accessed 2019/06/19. https://blogs.cisco.com/security/chronology-of-a-ddos-spamhaus.

Hansman, S. (2003). A Taxonomy of Network and Computer Attack Methodologies. Master's thesis, Department of Computer Science and Software Engineering, University of Canterbury, New Zealand. http://nzcsrsc08.canterbury.ac.nz/research/reports/HonsReps/2003/hons_0306.pdf.

Houle, K.J. & Weaver, G.M. (2001). Trends in Denial of Service Attack Technology. Technical report, CERT Coordination Center. Accessed 2019/07/01. https://resources.sei.cmu.edu/asset_files/WhitePaper/2001_019_001_52491.pdf.

Joyal, P.M. (1996). Industrial espionage today and information wars of tomorrow. In 19th National Information Systems Security Conference (pp. 139-151).

Kim, A.C. (2018). Prosecuting Chinese Spies: an Empirical Analysis of the Economic Espionage Act. Cardozo L. Rev., 40, 749.

Krebs, B. (2009). The Scrap Value of a Hacked PC. The Washington Post. Online. Accessed 2012/11/07. http://voices.washingtonpost.com/securityfix/2009/05/\the_scrap_value_of_a_hacked_pc.html.

Kshetri, N. (2005). Pattern of global cyber war and crime: A conceptual framework. Journal of International Management, 11(4), 541-562.

Lee, C.B., Roedel, C. & Silenok, E. (2003). Detection and characterization of port scan attacks. Technical report, University of California, Department of Computer Science and Engineering. Accessed 2013/01/01. http://www.cs.ucsd.edu/users/clbailey/PortScans.pdf.

Leyden, J. (2013). BIGGEST DDoS ATTACK IN HISTORY hammers Spamhaus. The Register. Online. Accessed 2019/05/18. https://www.theregister.co.uk/2013/03/27/spamhaus_ddos_megaflood/.

Long, J. (2007). Google Hacking for Penetration Testers, 2nd Edition (Vol. 2). Syngress Publishing.

Magklaras, G. & Furnell, S. (2001). Insider threat prediction tool: Evaluating the probability of IT misuse. Computers & Security, 21(1), 62-73.

Markoff, J. & Perlroth, N. (2013). Firm Is Accused of Sending Spam, and Fight Jams Internet. New York Times. Online. Accessed 2019/05/16. https://www.nytimes.com/2013/03/27/technology/internet/online-dispute-becomes-internet-snarling-attack.html.

Mirkovic, J. & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 34(2), 39-53.

Mookhey, K. & Burghate, N. (2004). Detection of SQL injection and Cross-Site Scripting attacks. Technical report INFOCUS 1768, Symantec. Accessed 2019/01/08. http://www.symantec.com/connect/articles/detection-sql-injection-and-cross-site-scripting-attacks

Moore, D. & Shannon, C. (2003). SCO Offline from Denial-of-Service Attack. Technical report. The Cooperative Association for Internet Data Analysis. Accessed 2018/09/02. http://www.caida.org/research/security/sco-dos/.

Nachenberg, C. (2012). Hacking. Technical report. BookRags. Accessed 2012/12/23. http://www.bookrags.com/research/hacking-csci-03/.

Noy, N.F. & McGuinness, D.L. (2001). Ontology development 101: A guide to creating your first ontology. Technical report KSL-01-05, SMI-2001-0880, Stanford knowledge systems laboratory and Stanford medical informatics technical report. Accessed 2013/01/01. http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf. Pogrebna,

Pogrebna G. & Skilton, M. (2019). A Sneak Peek into the Motivation of a Cybercriminal. In Navigating New Cyber Risks (pp.31-54). Springer.

Rounds, M. & Pendgraft, N. (2009). Diversity in network attacker motivation: A literature review. In 2009 International Conference on Computational Science and Engineering (pp. 319-323). IEEE.

Scharffe, F. & de Bruijn, J. (2005). A language to specify mappings between ontologies. In Proceedings of the 1st International Conference on Signal-Image Technology and Internet-Based Systems, (SITIS 2005), November 27 - December 1, 2005.Dicolor Press.

Shankland, S. (2003). Net attack crushes SCO Web site. Cnet. Online. Accessed 2012/09/02. https://www.cnet.com/news/ net-attack-crushes-sco-web-site/.

Simmonds, A., Sandilands, P. & van Ekert, L. (2004). An Ontology for Network Security Attacks. In S. Manandhar, J. Austin, U. Desai, Y. Oyanagi et al. (Eds.), Applied Computing (pp. 317-323). Berlin, Heidelberg: Springer Berlin Heidelberg.

Spitzner, L. (2000). Know your enemy. First Net Security. Online. Accessed 2018/12/18. http://old.honeynet.org/papers/ enemy/.

Taylor, P.A. (2001). Editorial: Hacktivism. The Semiotic Review of Books, 12(1), 1-4.

Undercoffer, J., Pinkston, J., Joshi, A. & Finin, T. (2004). A target-centric ontology for intrusion detection. In 18th International Joint Conference on Artificial Intelligence (pp. 9-15).

van Heerden, R.P., Burke, I.D. & Irwin, B. (2012a). Classifying Network Attack Scenarios Using an Ontology. In Proceedings of ICIW 2012 The 7th International Conference on Information-Warfare & Security (pp. 311-324). ACI.

van Heerden, R.P., Pieterse, H. & Irwin, B. (2012b). Mapping the Most Significant Computer Hacking Events to a Temporal Computer Attack Model. In Human Choice and Computers (HCC10) International Conference: ICT Critical Infrastructures and Society (pp. 226-236). Springer. IFIP.

Vasudevan, A. & Yerraballi, R. (2006). Spike: Engineering malware analysis tools using unobtrusive binary-instrumentation. In Proceedings of the 29th Australasian Computer Science Conference-Volume 48 (pp. 311-320). Australian Computer Society, Inc.

Velasco, D., & Rodriguez, G. (2017). Ontologies for Network Security and Future Challenges. ArXiv, abs/1704.02441.

Wang, A., Chang, W., Chen, S. & Mohaisen, A. (2018). Delving into internet DDoS attacks by botnets: Characterization and analysis. IEEE/ACM Transactions on Networking (TON), 26(6), 2843-2855.

Zhai, J., Chen, Y., Yu, Y., Liang, Y. et al. (2009). Fuzzy semantic retrieval for traffic information based on fuzzy ontology and RDF on the Semantic Web. Journal of Software, 4(7), 758-765.