Malware fuzzy ontology for semantic web

Tala Tafazzoli[†] and Seyed Hadi Sadjadi[††],

[†,††]faculty members of ICT security department of Iran Telecommunication Research Center

Summary
The term malware, in the area related to computer science, is used to define malicious code which is designed and written to execute attacks on software systems. In this paper, after a quick review of malact ontology, malware ontology is presented. Malwares include viruses, worms, botnets, spywares, backdoors, trojan horses, rootkits and exploits. In this paper malwares are grouped based on four attributes. These attributes include: Objective, operational status, establishment status and communication status. Then by analyzing malware and their characteristics, we propose malware ontology. The ontology is used to represent the concepts and their relationships in network security. One of the usages of ontology is information sharing and reuse in semantic web. In this paper, by proposing malware ontology, we presented the semantic relation map between concepts of this area which is used in semantic based search engines in incident area and CERT portals. Because malwares have similar characteristics, there is no clear boundary between their concepts so fuzzy logic is used to represent malware relationships. Malware relationships are presented in five categories: very weak relations, weak relations, moderate relations, good relations and very good relations and weights are assigned to them. If search is done on any concept (nodes) in the graph, the amount of the relationship with other concepts is calculated and based on the search demand and level of relationship, search is done on other related concepts.

Key words:
Information security, artifact, malware, ontology, fuzzy logic.

1. Introduction

Today malware is a very important concept in information security and lots of information is produced in this area. Malware is software with malicious intent which has the potential to harm the machine on which it executes or the network over which it communicates. [2] Malwares include viruses, worms, botnets, spywares, backdoors, Trojan horses, rootkits and exploits. [5]
Existence of different malwares causes disturbance between the concepts and the relationships between them. Despite the activities done in this area, there is no distinct classification which differs between the concepts and explains them carefully. Activity investigations done in this area shows good attempts which is done in this area. Some of these activities are described below.
Andrew Simmonds [31] has defined network security attacks ontology. The work done in this paper displays a framework for defining an extensible ontology for network security attacks. Nicholas Weavor[6] defined a taxonomy of computer worms. In his work a preliminary classification of worms based on target discovery and selection strategies, worm carrier mechanisms, worm activation and possible payloads is described. Martin Karresand [1] has proposed software weapon taxonomy. He explained that there are different taxonomies of software weapons which contradict each other. He tried to solve this problem. David Dagon has proposed botnet taxonomy and has described different topological structures that botnets use to coordinate attacks. Luciana A. F. Martimiano[33] has described computer security incident ontology and proposed an ontology evaluation method. John D. Howard [32] has described a common language for computer security attacks.
Investigation of proposed methods shows that there are three different methods for ontology representation as follows: manual ontology generation, semi automatic and automatic ontology generation methods. Noy and McGuinness [35] has proposed a manual ontology generation method. Some of semi-automatic and automatic ontology generation methods are based on textual, dictionary, knowledge base, semistructured schema and relational schema data types. [36] Some of semi-automatic and automatic ontology generation methods are as follows. Lee et al [37] proposed an algorithm for fuzzy ontology generation and news summarization. Tho et al. [38] proposed a fuzzy ontology generation framework on uncertain data. This framework is based on fuzzy theory idea and Fuzzy Concept Analysis (FCA).
Exploitation and proposal of the ontology in this field of science seems to be useful. Particularly, ontology is part of semantic web and shows the information in such a way that is understandable by the machine. [26] Ontology is used as a standard way to represent knowledge in semantic web. [27] Ontology is the main part of search engines in semantic web. In situations which we face uncertainty in knowledge and there is no specific boundaries between concepts, it is not enough to use ontology for concept formalization and thus classical logical methods for uncertain information investigation is not enough. Using fuzzy logic in uncertain information increases the semantic web power. In fact fuzzy logic and ontology have integrated with each other to explain the fuzzy ontology new paradigm. [26]

Manuscript received July 5, 2008.
Manuscript revised July 20, 2008.

Because different types of malware have similar characteristics, and there is no specific boundary between them, fuzzy ontology relations are used and relationships in the ontology are weighted. Fuzzy logic prepares the situation in which membership values vary between zero and one or has values such as “little”, “many”, or “exactly”.
Fuzzy ontology is used in search engines, and if each concept is searched over, search is done on all the concepts which have certain relationships with that concept. In this paper preliminary malware concepts and the relationship between them are defined. As preliminary malware concepts are investigated, concepts such as malact, artifact and non-artifact parade. They are investigated in the second chapter.
This paper has three sections in addition to introduction. In section two key concepts about malware ontology are investigated. The malact ontology is defined in section 2-1. The fuzzy ontology components are defined in section 2-2. Malware fuzzy ontology is developed in section three. In section 3-1 malware types are investigated. Malware properties are defined in section 3-2. Axioms and relations are defined in section 3-3. Conclusion and related work is proposed in Section 4.

2. Key concepts of malware ontology

2.1 Malact ontology proposal

Malware characteristics and behavior are a subclass of malact superclass. Malacts have unsuitable or destructive effect on the system or network and are classified to two main groups:

a) Artifact: destructive or malicious code which is prepared and is ready for work by someone is called artifact. [17] Malware is an important subclass of malact superclass. Malware is a program with malicious intent that has the potential to hurt the machine which is executed on or has the potential to disturb the network which communicates on. Malwares include viruses, worms, botnets, spywares, backdoors, Trojan horses, rootkits and exploits.
b) Non-artifact: Unsuitable and disturber behavior on a computer system or network which is not produced by a pre-written code and is the result of a specific or managed procedure on the computer system or network is called non-artifact. Spam is a non-artifact.

The malact ontology is shown in figure one.

2.2 Fuzzy ontology components

An ontology is shown as a four tuple O=(C,P,R,A) and every fuzzy part is shown with index F such as O=(C,P,R_F,A), [25] where,

- C is a Concept
- P is a set of concept properties, where $p \in P$ is an instance of a 3-tuple p(c,v,f) where $c \in C$ is an ontology concept, v is the value of concept c and f is the restriction values of v.
- R_F is the set of relations between concepts. R_F is defined as a 5-tuple as $r_F(c_1,c_2,t,s_F,U)$ where $c_1,c_2 \in C$ are ontology concepts, t represents relation type, U is the universe of discourse and s_F models the relation strength and is a fuzzy concept in U that shows the strength of the relation between $<c_1,c_2>$.
- A is axioms

3. Key concepts of malware ontology

Now we represent the values of (C)Concepts, (P)Properties, (R_F)fuzzy Relations and (A)Axioms

3.1 Malware types

As it is said before, there is no general and distinct definition for different malware variants. [28] In the following list, initial malware concepts and their definitions are introduced. Some of these definitions are extracted from different references.
Worm: Worm is a program that self-propagates in the network and while uses security flaws or policies in services [6], causes malicious actions on the victim. In addition to self-propagation property, worms have self-replication and self-contained properties. Self-replication means that it copies itself and self-contained means that the worm executes without the need to attach to another program. [30]
Virus: Virus is a program that attaches itself to another program to propagate. The program that the virus attaches itself to is the victim program. [8] Virus causes a malicious action on the victim.
Botnet: Botnet is a platform for malicious parallel processing. [9] The term botnet is used to define a network of malicious hosts called bots and is controlled by a human operator named botmaster. Bots use vulnerable machines with methods which are used by other malware classes (such as software vulnerabilities, social engineering, …) and they use the command and control (C&C) channel. [10]

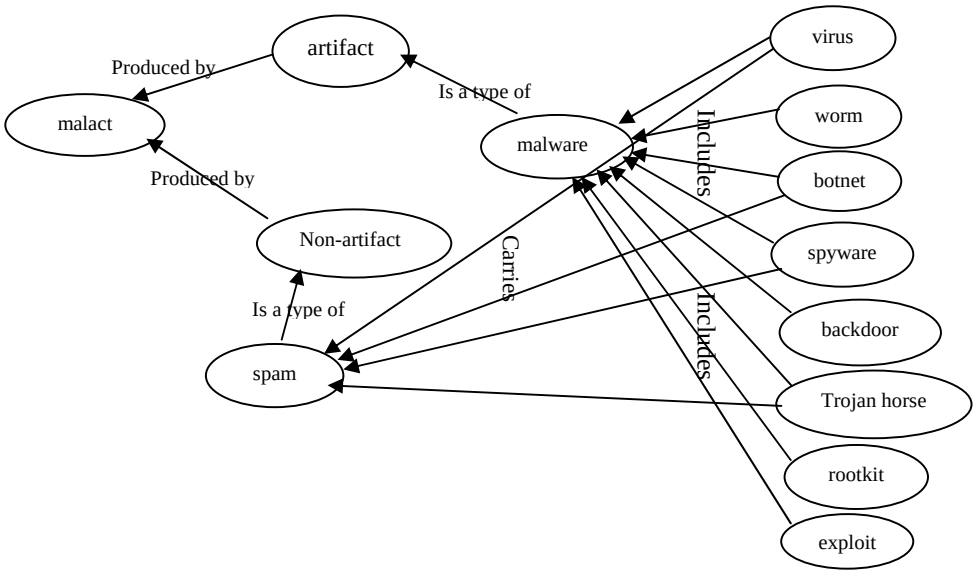

Fig. 1- malact ontology

Spyware: It is a software program which is placed on the victim machine and without the information and permission of the user sends personal information of the user to the third party. [11] Spyware stores the following information and sends them to the third party:

- Web search habits of the user
- Key strokes of the user
- User ID and password
- Important documents and passwords

Trojan horse: It is a software program which performs an unwanted or unknown action on the victim while it is known as a legitimate program. Trojan horse is a virus which does not replicate.

Rootkit: It is a program which is designed to hide the processes, files and activities of an attacker from the operating system and legitimate user and to access to the system. [18] Rootkits install a handler which omits the audit records and other records of the rootkit. [19] Rootkits have the following categories:

- Binary rootkits: These rootkits, replace binary files of the system such as ps, ls, netstat and these binaries replace the effects of the processes, files and open ports of rootkits.
- Kernel rootkits: These rootkits attach themselves to the kernel and change the system calls. They can change the kernel image, system call tables and other kernel components.
- Library rootkits: These rootkits replace the standard libraries of the system. Another way of their operation is that they can create a customized library which is added to the files.

Exploit: Exploit is a software, data or command which uses the vulnerabilities of the victim to create an unwanted or unpredicted behavior on the victim. [20] These behavior includes obtaining the control of the computer system, access control destruction, or denial of service.

Backdoor: backdoor is a software program which is installed by the attacker on a compromised system to facilitate the unauthorized further access of the attacker on the system. [29]

Based on the above definitions, the concept set is as follows. Because spam is not malware, it is not included in the concept set of this ontology.

C={botnet, exploit, rootkit, backdoor, spyware, Trojan horse, worm, virus}

3.2 Malware properties

Malwares have four types of characteristics. These characteristics are as follows and their values are introduced.

3.2.1 Malware operation from objective perspective

Malwares follow the following objectives. These are the values of the objective properties: [21]

- Unauthorized access: unauthorized access to information and resources including software and network services.
- Unauthorized use: access to the information or system or network by third party which do not have authorization for use.
- Disclosure: disclose system information or network to unauthorized person.
- Destruction: extinction of information or systems or networks or services
- Disconnection: discontinuity of services, information or network access

- Change: Unauthorized change of information, services, systems or network.
- Occupancy: unauthorized use of space, bandwidth, network or system resources

3.2.2 behavioral and technical characteristics of Malware from operational perspective

Operational characteristics of malware are as follows. These are the values of the operational characteristics property.

- Tangible: When some malwares are installed on the victim computer, they may cause disorder on the system and are called tangible because their existence is sensible.
- Intangible: Some malwares are intangible meaning that after installation on the victim computer, they don't cause any destruction on the victim computer and may cause operations such as information theft or duplication. These malwares are intangible and are not easily predictable and their existence is not sensible.
- Manual adjustment: In some malwares, the purulence target is determined by the attacker manually.
- Self-propagation: Some malwares determine the purulence target randomly and propagate from one computer to another with middleware, these are called self propagator.
- Single operation: If the target of the malware is only one computer, that malware is single operator.
- Network operation: If the malware has more than one victim, it has multiple operation.

3.2.3 Malwares from establishment method perspective

In this field malware architecture and victim placement is investigated. These are the establishment status values.

- Centeralized: The attack is initiated from a single point. Thus it has a single point of operation.
- Distributed: The attack is initiated from parallel sources, thus it has a distributed architecture.
- Local: If attack is done on the processor which the commands are executed, the malware has local property.
- Remote: If the attack is not done on the processor that the attack commands are executed on, the attack has remote property.

3.2.4 Malwares from the communications perspective

In this section, the type of malware communication with its creator and the management type of the malware are investigated. These are the values of the communication type property.

- Autonomic: Malware does not communicate with its creator.
- Dependent: Malware communicates with its creator.
- Central control: Malware is managed with a central control and receives commands from a central control.
- Without central control: Malwares does not communicate with a central control and receives commands from it.

3.2.5 Quadruple malware characteristics investigation

Now we investigate quadruple malware characteristics.

3.2.5.1 viruses

Because malwares cause disorder on the victim's computer, their purpose is destruction, disconnection and change on the victim computer. Because the changes that viruses exert on the systems are manifest and distinct changes, the viruses operational framework are tangible. Because viruses doesn't need human factors for copying and propagation, thus they have the self-propagation property. Because viruses aren't managed collectively, thus they are single operated. Because the target of the attack is the victim computer, they operate locally and viruses have central architecture. Viruses are autonomous and they don't have central control.

3.2.5.2 worms

Because worms cause disorder on the victim computer, their aim is causing destruction, disconnection and change. Because worm's effect on the victim is sensible, they are tangible and because they distribute on the network they have self propagation property. Sometimes worm's target is more than one computer and they operate on the network, thus they have network operation property. Worm's architecture is central and distributed and their target can be local and remote. Worms don't communicate with its originator and they don't have central control.

3.2.5.3 Trojan horse

Because Trojan horses cause disorder on the computer system, thus their aim is destruction, disconnection and change. Because the effect of the Trojan horse on the victim computer is sensible, they are tangible. Trojan

horse doesn't have self propagation property and is propagated by the attacker thus it propagates manually. Trojan horse infects only one computer thus it has single operation property. The architecture of the Trojan horse is central and its target is local. Trojan horse is copied by the attacker and thus is dependent. Trojan horse is not controlled by one controller and thus doesn't have central control.

3.2.5.4 Spyware

Because spyware steals secret information, thus its purpose is unauthorized access and because it doesn't disturb the victim machine, it is intangible. Spyware doesn't copy itself from one computer to another thus it is manual and because it infects only one computer, and doesn't operate in network, it has single operation property. Spyware architecture is centralized and isn't distributed and their target is local and doesn't target a remote computer. Spyware communicates with its creator and is not directed with its creator and doesn't receive commands from an attacker and its property is without central control.

3.2.5.5 backdoor

Because backdoor facilitates unauthorized access of the attacker to the system or network thus its target is unauthorized access. Because backdoor doesn't disturb the computer system thus it is intangible. Backdoor is created on the victim computer by the attacker thus it has manual property, it is not self-propagated and it operates on one computer thus it has single operation property. Backdoor has centralized architecture and isn't distributed, its target is local and it doesn't target a remote computer. Backdoor is created by the attacker on victim and thus it is dependent and is not controlled by another computer thus it is without central control.

3.2.5.6 rootkits

Rootkits change system files to hide unauthorized access of the attacker, thus unauthorized access and change are the goal of rootkits. Because rootkits doesn't change victim systems thus they are intangible. Rootkits doesn't have self-propagation property and they don't copy from one computer to another, thus they are manual and they have single operation property. Rootkits are centralized and they have local operation property because they attack the computer which executes the commands. Rootkits are attacker dependent and are without central control.

3.2.5.7 Exploits

Exploits facilitate unauthorized access on the victim machine. Because the changes they create on the victim are sensible they are tangible. They don't copy themselves from one computer to another, thus they are manual. Because the target of the attack is one computer thus they have single operation property. Exploits may cause denial of service attacks, thus they are distributed and their operation is local and remote. Exploits are attacker dependent and are without central control.

3.2.5.8 Botnets

Botnets are network of computers that initiate an attack on the victim. Thus they abuse computers. They are intangible and because computers are captured by the attacker they are manual and have network operation property. They are distributed and they have remote targets. They communicate with their creator thus they are dependent and have central control because they are directed by a computer.
Different types of malware and their characteristics are shown in table one.

3.2.6 Determining values related to malware characteristics

Malware characteristics is displayed with 3-tuple p(c,v,f). Now we determine the values of 3-tuple characteristics. For example, for virus concept, the objective attribute has the values {destruction, disconnection, change}. For determining the values that limit this characteristic, the weight related to each attribute (that is shown in section 3-3-1) and the number of attributes of each property is determined. [0-n] The limited value that is assigned to each property is determined by the equation (1).

$$F_r=w_p*i \text{ , } i\in[0,n] \qquad (1)$$

And i is the number of common attributes between two concepts. For the virus concept, the number of target attributes is three, because between each two concepts, the number of common values is one of the qualities of the set {0,1,2,3}, f_r for the virus concept is according to above equation and gets one of the values of the set {0,3,6,9}. Thus the 3-tuple for the virus attribute is as follows:
(virus,{destruction, disconnection, change}, {0,3,6,9})
Now we determine the 3-tuple attributes of other concepts.
P = {

Table 1- Malware characteristics

	Virus	Worm	Trojan horse	Spyware	Backdoor	Rootkit	Exploit	Botnet
Objective	Destruction Disconnection Change	Destruction Disconnection change	Destruction Disconnection change	Unauthorized-access	Unauthorized-access	Unauthorized-access Change	Unauthorized-use	Unauthorized-use
Operational method	Tangible Self-propagation Single-operattion	Tangible Self-propagation network-operattion	Tangible Manual Single-operation	Intangible Manual Single-operation	Intangible Manual Single-operation	Intangible Manual Single-operation	Tangible Manual Single-operation	Tangible Manual network-operation
Establishment method	Centralized Local	Centralized Distributed Local Remote	Centralized Local	Centralized Local	Centralized Local	Centralized Local	Distributed Remote	Distributed Remote
Communication method	Autonomic Without central-control	Autonomic Without-central-control	dependent Without central-control	dependent Without-central-control	dependent Without-central-control	dependent Without-central-control	dependent Without-central-control	dependent central-control

(virus, {destruction, disconnection, change}, {0,3,6,9}),
(virus, {tangible, self-propagation, single operation}, {0,5,10,15}),
(virus, {centralized, local}, {0,4}),
(virus, {autonomous, without central control}, {0,4}),
(worm, {destruction, disconnection, change}, {0,3,6,9}),
(worm, {tangible, self-propagation, network operation}, {0,5,10,15}),
(worm, {centralized, distributed, local, remote}, {0,4,8,10,12}),
(worm, {autonomous, without central control}, {0,4,8}),
(Trojan horse, {destruction, disconnection, change}, {0,3,6,9}),
(Trojan horse, {tangible, manual, single operation}, {0,5,10,15}),
(Trojan horse, {centralized, local}, {0,4,8}),
(Trojan horse, { dependant, without central control}, {0,4,8}),
(spyware, {unauthorized access}, {0,3}),
(spyware, {intangible, manual, single operation}, {0,5,10,15}),
(spyware, {centralized, local}, {0,4,8}),
(spyware, {dependant, without central control}, {0,4,8}),
(backdoor, {unauthorized access}, {0,3}),
(backdoor, {intangible, manual, single operation}, {0,5,10,15}),
(backdoor, {centralized, local}, {0,4,8}),
(backdoor, {dependant, without central control}, {0,4,8}),
(rootkit, {unauthorized access, change}, {0,3,6}),
(rootkit, {intangible, manual, single operation}, {0,5,10,15}),
(rootkit, {centralized, local}, {0,4,8}),
(rootkit, {dependant, without central control}, {0,4,8}),
(exploit, {unauthorized access}, {0,3}),
(exploit, {tangible, manual, single operation}, {0,5,10,15}),
(exploit, {distributed, remote}, {0,4,8}),
(exploit, {dependant, without central control}, {0,4,8}),
(botnet, {unauthorized use}, {0,3}),
(botnet, {intangible, manual, nework operation}, {0,5,10,15}),
(botnet, {distributed, remote}, {0,4,8}),
(botnet, {dependant, central control}, {0,4,8}),
}

3.3 Axioms and relations

3.3.1 Axioms

W_P, is the weight which each attribute has and is shown in the following set.
W_p= {(3,objective), (5,attribute), (4, establishment method), (4,communication method)}
The type of relation between two malwares is defined based on the following set:
T_F={too weak relation, weak relation, medium relation, good relation, very good relation}

3.3.2 Relations

As it is shown in section 2, relations are shown with 5-tuples $r_F(c_1,c_2,t,s_F,U)$. T_F is the kind of relations between two malwares which is defined in the axioms part. For defining the strength of the relation (S_F), the minimum and maximum values of S_F is defined first. These values are calculated with the following equations:
$S_{F\ min}$ = number of elements with minimum weight * minimum weight = 3*1
$S_{F\ max}$= Σ property weight * maximum number of attributes = w_p * max(n) = (3*6 + 5*3 + 2*4 + 2*4)=50

According to minimum and maximum permissible values of s_F, and because there are 5 types of relations between

two concepts, For calculating the strength of relations between two concepts, the space between $S_{F\ min}$ and $S_{F\ max}$ is divided into five sections and thus the set S_F that shows the strength of relations between concepts is as follows:
S_F={[1-10),[10-20),[20-30),[30-40),[40-50)}
The fuzzy relation between two concepts is calculated by the following equation:

$$R=\Sigma w_{p\,I} * n_i \quad , i\in[1\text{-}4] \qquad (1)$$

And n is the number of common characteristics in each attribute of two concepts.
U is the malware set.
The semantic relation between concepts which is calculated by the equation 1 is shown in the following set.
R = { (virus, worm, good relation, 35),
(virus, trojan horse, good relation, 31),
(virus, spyware, weak relation, 17),
(Virus, backdoor, weak relation, 17),
(Virus, rootkit, medium relation, 20),
(virus, exploit, medium relation, 20),
(virus, botnet, very weak relation, 0),
(worm, Trojan horse, medium relation, 26),
(worm, spyware, weak relation, 12),
(worm, backdoor, medium relation, 21),
(worm, rootkit, weak relation, 15),
(worm, exploit, weak relation, 13),
(worm, botnet, very weak relation, 9),
(Trojan horse, spyware, medium relation, 26),
(Trojan horse, backdoor, medium relation, 26),
(Trojan horse, rootkit, medium relation, 29),
(Trojan horse, botnet, very weak relation, 9),
(spyware, backdoor, good relation, 34),
(spyware, rootkit, good relation, 34),
(spyware, exploit, weak relation, 18),
(spyware, botnet, weak relation, 14),
(backdoor, rootkit, good relation, 35),
(backdoor, exploit, weak relation, 19),
Backdoor, botnet, weak relation, 15),
(rootkit, exploit, weak relation, 18),
(rootkit, botnet, weak relation, 14),
(exploit, botnet, medium relation, 20),
(Trojan horse, exploit, medium relation, 23)}
Malware fuzzy ontology according to relation type which is calculated in table 1 and sets S_F and T_F is shown in figure 2.
This fuzzy ontology is used in search engines. For example if the search looks for the word virus, the nodes which have very good, good and medium relation with virus are also searched and the results are shown.

4. Conclusions and further work

In this paper, meanwhile reviewing malact ontology, malware fuzzy ontology for use in semantic web is introduced. As it is explained in introduction, there are different taxonomies of worms, software weapons, botnets and ontologies of computer security incidents and network security attacks. This paper is the first malware ontology introduced. This ontology is generated manually and is based on the fuzzy ontology generation framework proposed in [25]. Information retrieval systems and search engines can use uncertain information. For displaying this uncertainty, fuzzy logic is used. According to received ontology, the relation between different types of malwares is very good, good, medium, weak and very weak and search on any malware may result in search on other types of malware which have good relation with that. Enhancement of vocabulary and addition of new nodes and relations are further works of this paper.

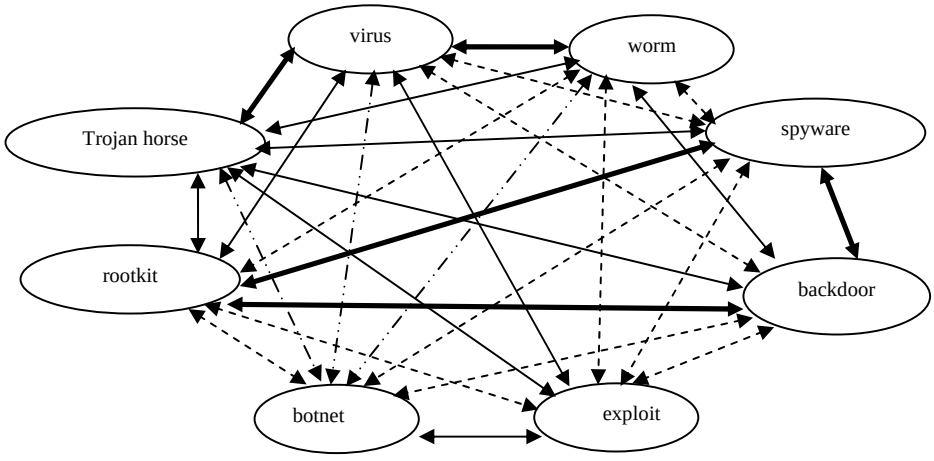

Fig 2- malware fuzzy ontology

References

[1] Martin Karresand , *A proposed taxonomy of software weapons*, Master's thesis in computer security, Linköping University, 2002.

[2] Mila Dalla Preda, Mihai Christodorescu and Somesh Jha, Saumya Debray, A Semantics-Based Approach to Malware Detection, , *ACM SIGPLAN-SIGACT symposium on principles of programming* languages, University of Verona, University of Wisconsin, University of Arizona, 2007.

[3] Nwokedi Idika, Aditya P. Mathur, A Survey of Malware Detection Techniques, Purdue University, 2007.

[4] J. Bergeron, M. Debbabi, J. Desharnais, M. M. Erhioui, Y. Lavoie and N. Tawbi, Static Detection of Malicious Code in Executable Programs, Universit ´e Laval, 2000.

[5] Ralf Benzmüller, Gertjan Vroon, Malware Annual Report 2005, G DATA Security, 2005.

[6] Nicholas Weavor, Vern Paxson, Stuart Staniford, Robert Cunningham, A Taxonomy of computer worms, UC Berkeley, ICSI, Silicon Defense, MIT Licoln laboratory, 2003.

[7] Sheng Bai, the classification and detection of computer worms, survey report, 2004.

[8] Vesselin Vladimirov Bontchev, *Methodology of Computer Anti-Virus Research*, PhD thesis, University of Hamburg, Germany, 1998.

[9] David Dagon, Guofei Gu, Cliff Zou, Julian Grizzard, Sanjeev Dwivedi, Wenke Lee, Richard Lipton, A taxonomy of botnets, Georgia Institute of technology, University of Central Florida, 2006.

[10] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, Andreas Terzis, A multifaceted approach to understanding the botnet phenomenon, Johns Hopkins University,*IMC'06*.

[11] Tzu-Yen Wang, Shi-Jinn Horng, Ming-Yang Su, Chin-Hsiung Wu, Peng-Chu Wang and Wei-Zen Su, A Surveillance Spyware Detection System Based on Data Mining Methods, *IEEE Congress on Evolutionary Computation*, 2006.

[12] Evan Cooke, Farnam Jahanian, Danny McPherson, The zombie roundup: understanding, detecting and disrupting botnets, University of Michigan, Arbor networks, 2006.

[13] William Stallings, *Cryprography and network security*, 2005.

[14] Marko Helenius, A system to support the analysis of antivirus products' virus detection capabilities, university of Tampere, 2002.

[15] Anselm Lambert, *Analysis of spam*, Master's thesis, University of Dublin, 2003.

[16] Chris Wysopal, Chris Eng, Static detection of application backdoors, Veracode Inc., Burlington, MA USA, 2005.

[17] Jason Milletart, Trends in internet attack technology and the role of artifact analysis, cert coordination center, 2005.

[18] Jesse D. Kornblum, Exploiting the rootkit paradox with windows memory analysis, *international journal of digital evidence*, 2006.

[19] Stephen M. Specht, Ruby B. Lee, Distributed Denial of service: taxonomies of attacks, tools and countermeasures, Princeton architecture laboratory for multimedia and security, technical report, 2003.

[20] http://en.wikipedia.org/wiki/Exploit_(computer_security)

[21] ITU-T recommendation X.805, Security architecture for systems providing end-to-end communications.

[22] George F. Luger, William A. Stubblefield, *Artificial intelligence, structures and strategies for complex problem solving*, Addison Wesley longman, 1998.

[23] James J. Buckley, Estandiar Eslami, *An introduction to fuzzy logic and fuzzy sets*, Physica Verlag, 2002.

[24]Jeffrey J. Farah, Robert B. Kelly, Utilizing Semantic networks to database and retrieve generalized stochastic colored petri nets, *IEEE Intelligent systems for space exploration*, Rensselaer polytechnic Institute, 1992.

[25]Jun Zhai, Lixin Shen, Zhou Zhou, Yan Liang, Fuzzy ontology model for knowledge management, Dalian Maritime University, 2007.

[26] Silvia Calegari, Elie Sanchez, A fuzzy ontology-approach to improve semantic information retrieval, University de Milano, 2007.

[27] T. Berners-Lee, Semantic web roadmap, W3C Design Issues, 1998.

[28] Martin Karresand, Seperating Trojan hourses, viruses, and worms – A proposed taxonomy of software weapons, *IEEE workshop on information assurance*, 2003.

[29] Yin Zhang, Vern Paxon, detecting backdoors, Cornell University, AT&T center for internet research at ICSI, 2000.

[30] SP 800-61, Computer security incident handling guide, NIST, 2004.

[31] Andrew Simmonds, Peter Sandilands, and Louis van Ekert, An ontology for network security attacks, *RAID 2003, LCNS 2820, Springer-Verlag*, 2003.

[32] John D. Howard, Thomas A. Longstaff, A common language for computer security

incidents, Sandia National Laboratories, Sandia Report, 1998.
[33] Luciana A. F. Martimiano, Edson Moreira, The evaluation process of a computer security incident ontology, University de Sao Paulo, 2005.
[34] David Dagon, Guofei Gu, Wenke Lee, A taxonomy of botnets, University of central Florida, 2006.
[35] Natalya F. Noy, Deborah L. McGuinness, Ontology development 101: A guide to creating your first ontology, Stanford University, 2000.
[36] Asuncion Gomez-Perez, David Manzano-Macho, A survey of ontology learning methods and techniques, Ontoweb: Ontology-based information exchange for knowledge management and electronic commerce, 2003.
[37] C.S.Lee, Z.W. Jian and L. K. Huang, A fuzzy ontology and its application to news summarization, *IEEE Transactions on Systems, Man and Cybernetics (Part B),* 35(5):859-880, 2005.
[38] Q. T. Tho, S. C. Hui, A. C. M. Fong and T. H. Cao, Automatic fuzzy ontology generation for semantic web. *IEEE Transactions on Knowlwdgw and Data Engineering*, 18(6):842-856, 2006.

Manager of information society security group in ITRC

Faculty member of Information society security group in ITRC