International Journal of Computer & Information Technologies (IJOCIT)
www.ijocit.org & www.ijocit.org
ISSN = 2345-3877

Presenting a Metric-Based Model for Malware Detection and Classification

M. Sirwan Geramiparvar[1], Dr. Nasser Modiri[2]
M. Sc, Department of Electrical[1], Associate Professor in Department of Electrical[2]
Computer & IT, Zanjan Branch, Islamic Azad University[1, 2]
Zanjan, Iran[1, 2]
sirwan_gp@yahoo.com [1]

Abstract: Nowadays, malware is a known term in the cyber world, which has been created with the bad intents of spying, sabotage, changing, deleting information, and disordering. So the enormous direct and indirect costs carried by companies and organizations and its bad effects on their normal and commercial operation are undeniable. Until now, different approaches have been suggested for malware detection and classification. These approaches are divided into three groups of signature-based detections, behavior-based detection, and heuristics. Each one can be applied as static, dynamic (virtually simulation) or a mixture of both. Unfortunately, present methods aren't efficient anymore. In this article, based on the malware behaviors, nine metrics are introduced according which a method for their detection and a model for their classification are represented.

Keywords: malware, malware metrics, metric's classification, metric-based model, Fuzzy AHP.

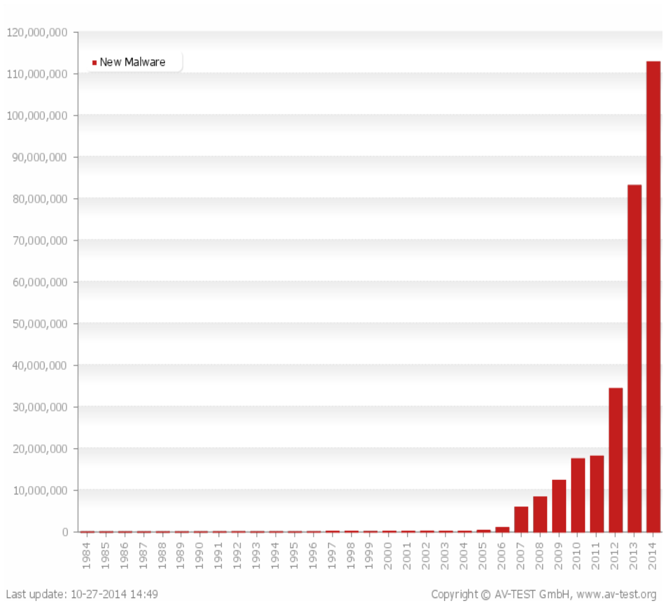
1. Introduction

Increasing extension of information systems and computer networks on one hand, and offices computerization and their dependence on storing financial, military, hygienic, and personal information in servers and their distribution throughout the computer networks to be easily accessible to the costumers, users, and personnel, on the other hand, have tempted the hackers and scammers not only to access the information unauthorized, but also to change, sabotage and hide them. According to the released statistics of Av-Test [1] organization, 450000 malwares are entering the cyber world every day. (Figure 1)

Every day we hear some news about cyber-attacks and cyber warfare in the field of information and communications security. These kinds of attacks are usually carried out with political aims in order to hurt economic infrastructure. They are employed without any physical weapons and of course are the most destructive ones. Malwares play an important role as weapons in cyber warfare and repelling them is considered as one of the biggest and

[1] http://www.av-test.org/en/statistics/malware/ [Online]

most crucial challenges in passive defense and cyber war footing. Mal is a Spanish prefix meaning bad.



Last update: 10-27-2014 14:49

Copyright © AV-TEST GmbH, www.av-test.org

Figure 1: New Malware Statistics, the distance between 1984 to 2014 years

Malware, short for malicious software, is used as a general term by computer experts. It means an invader program code that is hostile and annoying. Up to now, many malware classifications have been introduced in different articles and studies. CARO[2] naming system is usually used for naming and classification of malware in security products like anti viruses. Usual classifications are based on general features like duplication, distribution, and the way of penetration, according which malwares are divided into different groups like viruses, worms, Trojans, spywares, crimewares and.. There are three ways for malware classification and detection: Signature-based, Behavior-based, and Heuristics. In Signature-based approach or Characteristics-based, according to general and physical malware characteristics like file size and its operation, a signature is assigned and based on the similarities with this signature, its resemblances or correspondence can be determined. In Behavior-based Approach, Classification is done according to malware's behaviors and its behavioral patterns. In Heuristics approaches, some methods like Data Mining, Control Flow Graph, n-gram, and... Are used. All of these approaches are analyzed by the use of Static Analysis (like String Analysis, Hashing, and...) and Dynamic Analysis (like simulation, process and debugger monitoring and...) or a mixture of both. These methods have lost their efficiency to some extent and aren't suitable for new malwares. In this article, based on the malwares' behaviors and their attacks, studying dictionaries like CVE, CWE, and CAPEC and under the influence of existed models, 9 metrics are suggested for detecting malwares' vulnerability and deficiency. Also based on these metrics, one model for malwares' detection and classification has been introduced.

2. Related Works

In an article by Rafiqoleslam et al. [3], in different levels of the malware Life Cycle Analysis, Pattern Recognition algorithms and Static methods have been adopted. The framework combines the Static Features of Function Length and Printable String Information extracted from malware samples. In this research, about 1400 decrypted malware samples were applied to different classification algorithms. FLF (Function Length Frequency) and PSI (Printable String Information) techniques suggested by Tian et al. 2008 and 2009 have been employed in this research. General classification accuracy was 98 percent. In another research by Mr. Sarkardei and his colleague [6], an intelligent method based on the last section of the executable files has been

[2] http://www.caro.org/naming/scheme.html [Online]

presented in which 2 bytes in the last section of the file is considered as word. Classification methods employed in this research are Feature Weighting Method, used for text classification, and The Nearest Neighbor Method. For recognizing clean files from malicious ones, the differences between these two executable files were studied and based on these differences, a strategy was represented for recognizing clean files from malicious ones. For classification, K-Nearest Neighbors Method was used that is the most efficient and fastest method in text classification. In order to determine the distance between two vectors, the Euclidean Distance was used. In this experiment, 167 clean files and 167 malicious ones were studied. 1/3 of them were used for test and 2/3 was used for learning. According to the results obtained from Weighting Methods TFIDF Log TF, with 99/10 accuracy, FPR= 0, and TPR= 98/21, 111 files containing 56 malicious files and 55 clean files were the most efficient ones. According to another research by Arsenjani [5], an algorithm based on Machine Learning was introduced that could classify clean and malicious files with high accuracy. N-grams algorithm was used as the base method for feature extraction. Also, after studying more than 100 million extracted features, the best value for N was suggested. For this purpose a new algorithm for feature selection called iselection with high efficiency was presented. Also in order to decrease Error Rate, Majority Voting Architecture based on Naive Bayes Algorithm was used for samples classification that due to being independent, it would have high concurrency ability. By testing 18 million features in a set of test samples, they concluded that values less than 3 and more than 5 for N is not applicable. In another article written by Rahimi and his colleagues [4], a method based on Data Mining Techniques was suggested according to this article, API Calls in executable files can help us to gain a useful knowledge of executable files' aims and behaviors. In this study, 32000 destructive files from different kinds of malwares and 30000 files including windows operating system's files from different versions were tested. The obtained tracking rate in this experiment was 99,3 percent that was better than previous methods. In Mr. Lee and his colleagues' article [7], a set of malwares samples were analyzed. They also discussed the way that MAEC classification can help us to solve the problems with malwares. In their opinion, however, security systems like Intrusion Detection System, Honey Pots, and anti-viruses exist, the malwares dynamic nature and their attacks make it very difficult for us to detect and hinder them. Every day we see lots of Zero Day Attacks (some unknown attacks that are happening recently.) this research confirms that there is no systematic method for solving malware problems so there will be an eternal gap between malware attacks and counter strike softwares. Recently, a framework called MAEC (Malware Attribute Enumeration and Characterization) belonged to MITRE organization, has suggested a uniform classification structure for malwares. In this article, different kinds of malwares have been analyzed by the use of MAEC framework in order to be described and classified. The next study was done by two Korean researchers called Kim and Moon [8]. They believed that due to complexity of Malware Hiding Techniques, feature based detection methods are unable to cope with these techniques simultaneously. In this research, a method for detecting the hidden identity of malware by focusing on malwares scripts has been suggested. This system has a metric based Combinatorial Algorithm and Genetic Algorithm. Metric based methods use number vectors for representing features of each program. This code acts really well in plagiarism recognition. In this research, this method has been adapted for detecting malwares' scripts by the use of alternative tokens. In order to find subverted part of the program, Genetic Algorithm was employed. Of course, it should be mentioned that they represented the supplementary article about this model in 2013 [9]. Suggested system includes 14 modules. The first part, Decision Algorithm, determines whether the program is subverter or not. In the second module, Malicious Core Finder, Genetic Algorithm has been used for extracting the subverted part of the program, since it really look likes the malware. Next module is metric calculator that converts the programs to number vectors with different metrics. Finally Distance Calculator calculates the distance between vectors.

3. Available Models

In order to study the nature of the malware (ontology), there are different methods like Swimmer [14], MAEC[3], and some languages for expressing security events, like open-IOC [4], IODEF[5], and VERIS[6] that use XML schema. Also, there are 3 attack patterns and process models. Howard & Longstaff model [12], Gadelrab's first [10] and second models [11]. In Mr. Gadelrab and his colleagues' second model, all of the attacks steps are divided into 8 steps or phases. (Table 1) These phases consist of:

Table 1: 8 Steps of Gadelrab Malware Attacks

1. R: Reconnaissance
2. VB: Victim Browsing
3. EP: Execute Program
4. GA: Gain Access
5. IMC: Implant Malicious Code
6. CDI: Compromise Data Integrity
7. DOS: Denial of Service
8. HT: Hide Trace

He introduced the general schema of his model as the following figure. (Figure 2) By using this model, some malwares Flowchart and famous attacks like Code Red I, Code Red II, Trinoo, and…were drawn and studied. In another research, Saber and his colleagues [2], completed Gadelrab's model and presented it as the following figure, (Figure 3) that is a kind of state machine. Since the Reconnaissance phase (R) can be considered as the first step that influences other ones, the following table (Table 2) is introduced as the points table of attack steps inspired by the connection matrix of attack steps in Saber article, and based on a logical deduction:

Table 2: Scoring Attack Phases

	R	GA	DoS	VB	CDI	EP	IMC	HT	Sum
R	1	1	1	1	1	1	1	1	8
GA	0	1	1	1	1	1	1	0	6
VB	0	0	1	1	1	1	1	0	5
CDI	0	0	0	1	1	1	1	1	5
EP	0	0	1	1	1	0	1	1	5
IMC	0	0	0	1	1	1	0	1	4
HT	0	0	0	0	0	1	0	0	1
DoS	0	0	0	0	0	0	0	0	0

According to this table if during an attack or in a phase of attack, the malware or hacker gain access, it can also access to the other 6 phases, and this phase of attack is much more dangerous than a phase like Hide Trace.

[3] http://www. http://maec.mitre.org/ [Online]
[4] http://www.openioc.org/ [Online]
[5] http://xml.coverpages.org/iodef.html [Online]
[6] http://verisframework.wiki.zoho.com/ [Online]

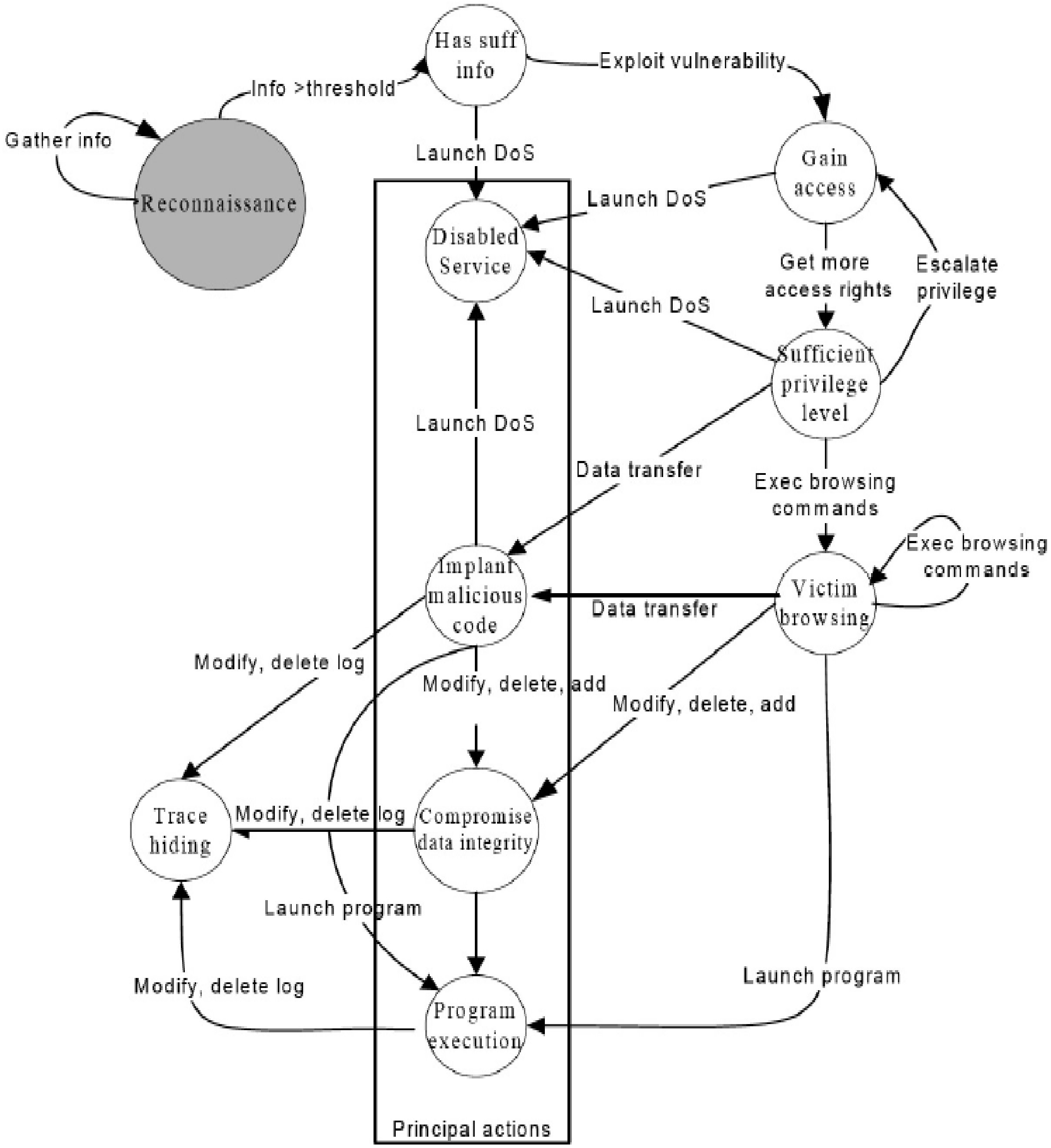


Figure 2: Second Model of Gadelrab's

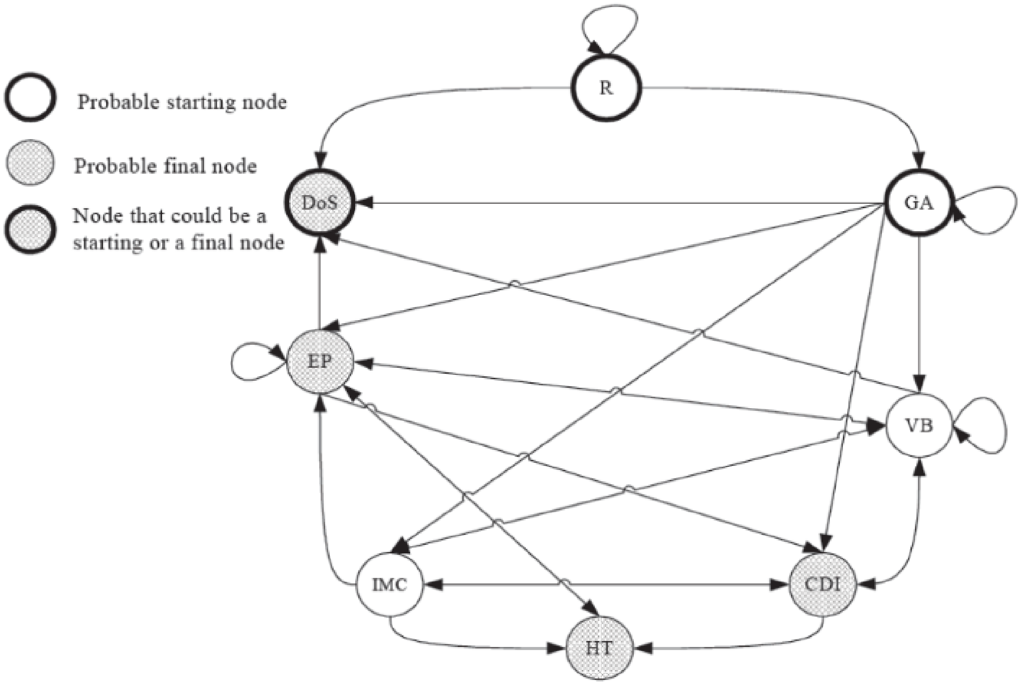


Figure 3: State Machine Model of Gadelrab's

4. Proposed Metrics

To suggest metrics, lots of studies carried out on cyber sources and different languages. Before the final introduction to the metrics, we need to require a brief familiarity with OWASP.

4.1. OWASP[7]

OWASP is international and nonprofit organization that works to create security, designing, implementations, expansion and testing software projects. All existing documents, tools, and check lists in the official site of this organization are free and have been improved to solve common security vulnerabilities in all softwares frameworks. Many thousands of active users in all over the world are working on this project with the aim of performance and softwares' enhancement. Once every few years, this organization makes a list of critical vulnerabilities in softwares and web services all around the world. This list is the base of security in web applications. In 2013, OWASP released the list of the top ten most dangerous vulnerabilities as indicated in Figure 4.

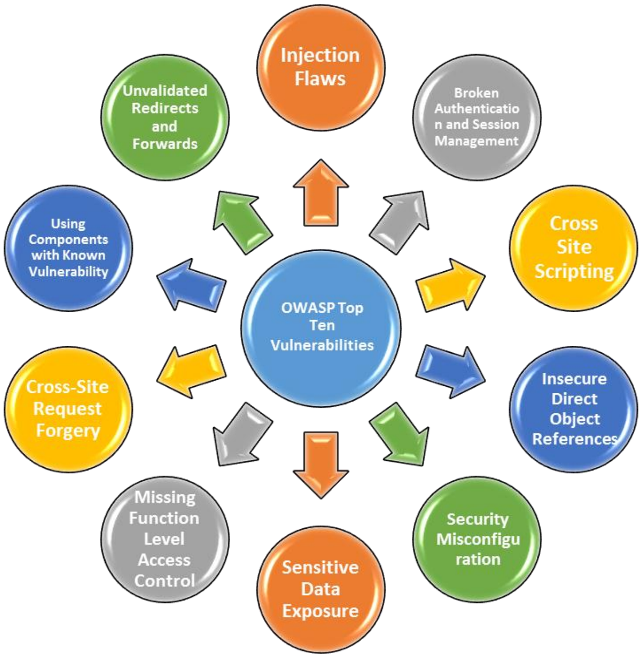

Injection Flaws
Broken Authentication and Session Management
Cross Site Scripting
Insecure Direct Object References
Security Misconfiguration
Sensitive Data Exposure
Missing Function Level Access Control
Cross-Site Request Forgery
Using Components with Known Vulnerability
Unvalidated Redirects and Forwards
OWASP Top Ten Vulnerabilities

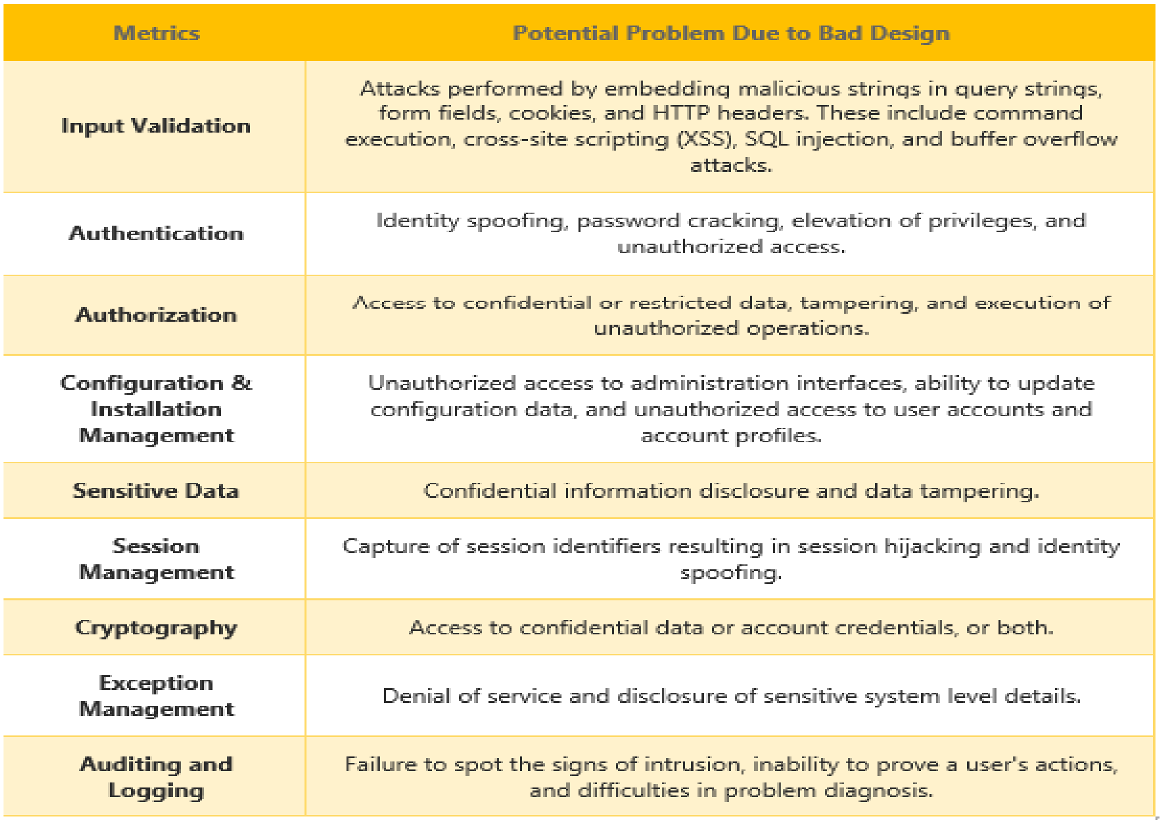
Figure 4: OWASP Top Ten Vulnerabilities 2013

[7] http://www.owasp.org [Online]

4.2. Metrics' Introduction

Regarding above mentioned vulnerabilities in web sites and security companies portals like acunetix, net-security, Panda security, ESET, Symantec, SANS, IBM, Cisco, CENZIC, and… that represent annual reliable statistics on high numbers of threats and security problems in their technical reports and white papers and according to the several points of similarities between CWEs like (SQL Injection, XSS, and…), the reviewed models (like Gadelrab's first and second models and other represented models in different researches), Cheat Sheet methods, inspired by OWASP, collecting some features and under the influence of practice and pattern models by Microsoft, 9 metrics for attacks and computer malwares' classification were extracted. Table 3 indicates these 9 metrics and explains which metric is applicable for each vulnerabilities and defections.

Table 3: Proposed Metrics



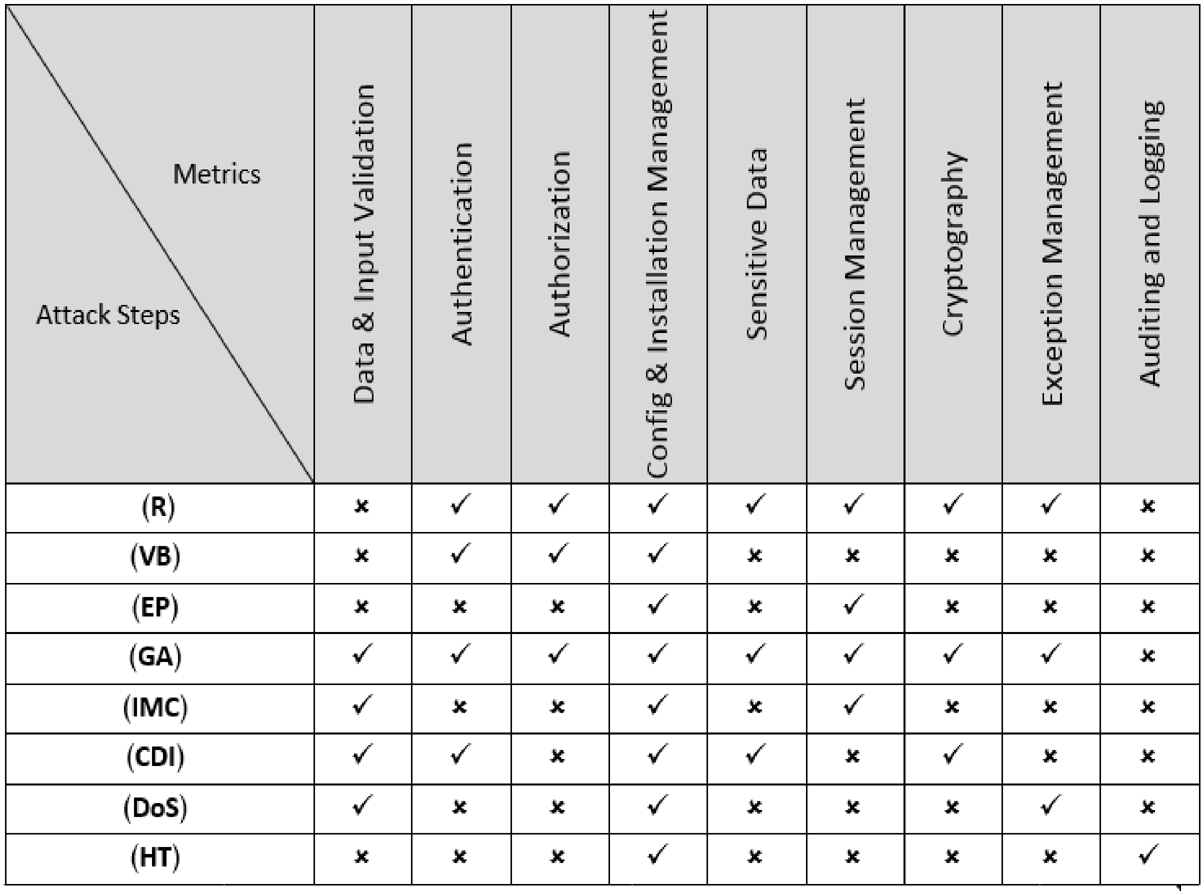
Metrics	Potential Problem Due to Bad Design
Input Validation	Attacks performed by embedding malicious strings in query strings, form fields, cookies, and HTTP headers. These include command execution, cross-site scripting (XSS), SQL injection, and buffer overflow attacks.
Authentication	Identity spoofing, password cracking, elevation of privileges, and unauthorized access.
Authorization	Access to confidential or restricted data, tampering, and execution of unauthorized operations.
Configuration & Installation Management	Unauthorized access to administration interfaces, ability to update configuration data, and unauthorized access to user accounts and account profiles.
Sensitive Data	Confidential information disclosure and data tampering.
Session Management	Capture of session identifiers resulting in session hijacking and identity spoofing.
Cryptography	Access to confidential data or account credentials, or both.
Exception Management	Denial of service and disclosure of sensitive system level details.
Auditing and Logging	Failure to spot the signs of intrusion, inability to prove a user's actions, and difficulties in problem diagnosis.

Each metric plays a role in creating one or more vulnerabilities and all malwares intrusion and their attacks occur as a result of a defection or lack of complete maintenance of each metric so in order to fight against malwares and repel their attacks we need to maintain these metrics otherwise we will face risk.

4.3. Mapping between metrics and attack steps

After having studied the role of each metric in creating vulnerabilities and attacks caused by the lack of their maintenance, and based on the adopted method in Gadelrab and his colleagues second model, we can represent a mapping for creating connection between attacks octamerous stages in the model and nine fold metrics as presented in the following table. (Tab. 4) It should be explained that the sign ✓ shows the relationship between metric and attack's step and it means that the lack of a maintaining a metric like Data Validation leads to Implant malicious Code and vice versa. Also the sign × shows the lack of any specific relationship or even no relationships at all. Table 4 shows above mentioned mapping.

Table 4: Mapping a Relation between Metrics and Attack Steps



Metrics / Attack Steps	Data & Input Validation	Authentication	Authorization	Config & Installation Management	Sensitive Data	Session Management	Cryptography	Exception Management	Auditing and Logging
(R)	×	✓	✓	✓	✓	✓	✓	✓	×
(VB)	×	✓	✓	✓	×	×	×	×	×
(EP)	×	×	×	✓	×	✓	×	×	×
(GA)	✓	✓	✓	✓	✓	✓	✓	✓	×
(IMC)	✓	×	×	✓	×	✓	×	×	×
(CDI)	✓	✓	×	✓	✓	×	✓	×	×
(DoS)	✓	×	×	✓	×	×	×	✓	×
(HT)	×	×	×	✓	×	×	×	×	✓

5. Metrics' Classification

To classify metrics, according to the mapping table and the effects of each metric on attack phases, they were weighted and to prioritize them. We used Fuzzy Analytic Hierarchy Process (FAHP). After analyzing and making calculations, we were able to reach to a kind of conclusion as indicated in the following table (Table 5):

According this table, compatibility factors are CR^m=0, 023 and CR^g=0,043 that represent the compatibility of factors. So the prioritization has been correct and based on the importance of each metric and their role in per attack's phases, we can design and carry out reciprocal operations and concerning their weight defend against them to a certain extent.

Table 5: Metrics' Prioritization with FAHP method

Metric	Fuzzy Weight	Normalized Weight
Configuration & Installation Management	0.465313	(0.265,0.354,0.515)
Session Management	0.111018	(0.066,0.114,0.149)
Authentication	0.111018	(0.066,0.114,0.149)
Input & Data Validation	0.111018	(0.066,0.114,0.149)
Cryptography	0.0461714	(0.056,0.071,0.108)
Sensitive Data	0.0461714	(0.056,0.071,0.108)
Exception & Error Management	0.0461714	(0.056,0.071,0.108)
Authorization	0.0461714	(0.056,0.071,0.108)
Auditing & Logging	0.0169469	(0.017,0.021,0.032)

5.1. Suggested Model

Concerning suggested metrics and representing Fuzzy Analytic Hierarchy Process, for the prioritization model applicable in security softwares products like PENTEST systems or IDS, and firewalls can be suggested. This model is a more efficient and secure method for detecting, avoiding, and preventing from computer malwares and cyber-attacks. It is inspired by Parmelee's suggested model [13] to some extent. In this model (Figure 5), at first suspected or malicious code enters the system and is analyzed by the help of an analyzer (like the comprehensive and integrated analyzer suggested by Rafiqoleslam, Tian et. al[1]) This analysis can be simulated in a dynamic environment and physical features and characteristics analysis can be accomplished by the static based methods. Then according to the amount of exploits that it derives from different vulnerabilities, it will be analyzed by suggested metrics. By the use of weighting the amount of vulnerabilities and defections exploited due to the lack of metric maintenance, and considering metrics priority, we can collect the information together with the Code analysis will be sent to decision making management system. In this system, based on the sent information from the previous stages and by the use of cognitive bases and dictionaries like MAEC, CVE, CWE… we can determine which class the malware belongs to exactly or which one it is the most similar to. Since the metrics weaknesses have been clear in the previous stage and security breach in metrics has been weighted, finding an efficient and effective method and making decision about it would be easier and faster. In the following figure the outline of the metric based model has been proposed.

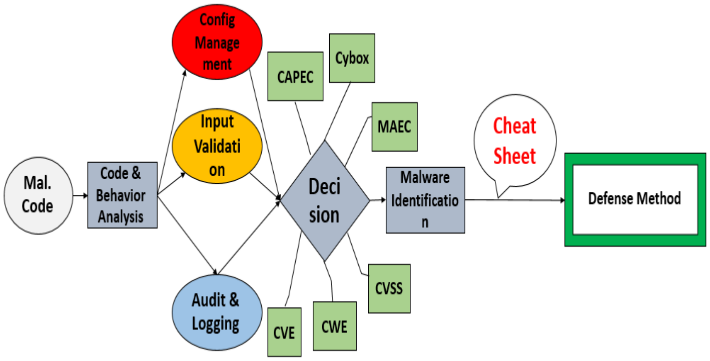


Figure 5: Suggested Metric-based model

6. Results and Discussion

We apply this model to a number of malicious programs like Operation Emmental, Stuxnet, XSS, and… we reached the conclusion that the model can be an appropriate language to identify and classify Malware. According to the presented model, the following points can be outlined. This research and suggested metrics for malware recognition and classification is employed for the first time. So it can be considered as a new method for malwares and cyber-attacks recognition and classification also more research can be done on it. Metrics can be used as a set of factors and lexical dictionaries (Syntax) in a new language by which the malware can be described. It has been introduced according to the latest suggested standards and models for malwares description (based on MITRE standard) and it employs dictionaries and reach databases like CWE, CVE, and CAPEC. Its advantage over Gadelrab model is that, at the time of analyzing attacks, his model can introduce different stages of attack and we just write them down, but in this model we can use the attacks stages threads along with the weaknesses which were due to the lack of metric maintenance as a combinational string that is a sign of signature (feature) and introduce it as a signature based method. It examines malware's behavior flow and operation flow semantically and it can improve significantly. Using cheat sheets facilitated confronting, counteraction and deception attacks and provides more security. It's implementable and compatible with all implementation controls of Information Security Management System (ISMS). According to metric's prioritization and determining their level of importance, we can estimate the exact costs needed for fighting against malwares and cyber-attacks and it is really affordable. This model's advantage over OWASP model is that in OWASP model based on the existing versions (as 2010 and 2013 versions) top 10 vulnerabilities are introduced and the solutions are represented. But in this model, based on 3 viewpoints; metrics, vulnerabilities, threats and discusses deficiencies and after identifying and recognizing behavior, and represents reciprocal action that is more complete than OWASP.

7. Future Works

Since this model and method are new, we can summarize our future works as following: the simulation of attack process based on the suggested model in a practical environment as in virtual machine, the simulation of confronting attack by cheat sheets considered in metrics, according to the amount of recent attacks and becoming combinational, representing a model for combinational attacks (presenting mathematical algorithm) with multi threads, speed evaluation, the model's efficiency, its comparison with other existing models and presenting a model and its evaluation for combinational or incomplete attacks.

8. References

[1] R. Islam et al. (2013) "Classification of malware based on integrated static and dynamic features". Journal of Network and Computer Applications, Vol. 36, pp. 646–656.

[2] M. Saber, T. Bouchentouf, A. Benazzi, M. Azizi. (2010) "Amelioration of Attack Classifications for Evaluating and Testing Intrusion Detection System". Journal of Computer Science, Vol. 6, Issue 7, pp. 716-722.

[3] R. Islam, R Tian, L Batten, S Versteeg. (2010) "Classification of Malware Based on String and Function Feature Selection". Second Cybercrime and Trustworthy Computing Workshop, (CTC-2010), 19-20 July, Ballarat, Australia.

[4] A. Sarkardei, A.A. Pouyan, H. Hassanpour. (2013) "Presenting an Intelligent Approach based on text classification due to unknown malicious code recognition". 11st. Conference on Intelligent Systems, (ICIS-2013), 28 Feb and 1 Mar, Iran.

[5] M. Arsanjani. (2012) "Malicious Code Detection by the use of Machine-learning classifiers based on Static Characteristics". 2nd. Conference on Information Technology, now and Future, Mashhad, Iran.

[6] M.R. Ghassemi, M. Amini laari. (2012) "Presenting a Malware Detection System based on Dynamic Analysis by the use of User Interaction". National Conference on Information Technology and Economic Jihad, 22-23 Feb, Kaazeroun, Iran.

[7] A. Lee, V. Varadharaju, U. Tupakula. (2012) "On Malware characterization and Attack Classification". Proceedings of the First Australasian Web Conference, (AWC-2012) CPRIT Vol.144 Adelaide, Australia.

[8] Jinhyun Kim, Byung-Ro Moon. (2012) "New Malware Detection System using Metric– based Method and Hybrid Genetic Algorithm". Proceeding of the 14th annual Conference Companion on Genetic and evolutionary Computation ACM, New York, NY, USA, pp. 1527- 1528.

[9] Jinhyun Kim, Byung-Ro Moon. (2013) "Disguised malware script detection system Using Hybrid Genetic Algorithm". Proceedings of the 28th Annual ACM Symposium on Applied Computing, (SAC '13).

[10] M. Gadelrab, et. al. (2007) "Defining categories to select representative attack test-cases". Proceedings of the ACM workshop on Quality of protection, ACM New York, NY, USA, pp. 40-42.

[11] M. Gadelrab, A. Abou El Kalam, Y. Deswarte. (2008) "Execution Patterns in Automatic Malware and Human-Centric Attacks". 7th IEEE International Symposium on Network Computing and Applications, (IEEE NCA07), 12-14 July, Cambridge, MA, USA.

[12] John D. Howard & Thomas A. Longstaff. (1998) "A Common Language for Computer Security Incidents", SANDIA Report, SAND 98-8667.

[13] Parmelee, M. (2010) "Toward an Ontology Architecture for Cyber-Security Standards", Semantic Technologies for Intelligence, Defense, and Security (STIDS), 22nd October. George Mason University, Fairfax, VA, USA.

[14] Swimmer, M. "towards an Ontology of Malware Classes. [Online]. Available:http://www.scribd.com/doc/24058261/Towards-an-Ontology-of-Malware-Classes [Accessed on 20 October 2014]