R4 Model For Malware Detection And Prevention Using Case Based Reasoning

Venmaa Devi P, [a] Karpagam G R [b]

[a]Student,PSG College Of Technology, India, [b] Professor,PSG College of Technology,India

Abstract: Nowadays, with the rapid development of technology, the attacks on networks and computer have been tremendously increasing. One of the serious threats is malware which makes the computer networks more vulnerable to attack from hackers. There exist various methods to detect malware like Signature based algorithms. But these methods may not prove efficient because of the polymorphism property of the malware and these detection mechanisms are static. In this paper, efforts have been taken to design a R4 model(Retrieval,Reuse,Revise,Retain) for case-based reasoning of malware detection and prevention.

Keywords—malware, Signature based algorithms, R4 model, Case based reasoning.

1 Introduction

With the advent of technology, vulnerability of computer networks has become a great issue. One of the serious threats is malware. Malware is short for malicious soft*ware*, meaning software that can be used to compromise computer functions, steal data, bypass access controls, or otherwise cause harm to the host computer.

1.1 Types of Malware

Malware is a broad term that refers to a variety of malicious programs.

1.1.1 Viruses and Worms

The best-known types of malware, viruses and worms, are known because they spread, rather than any other behavior. The term computer virus is used for a program that has infected some executable software and, when run, causes the virus to spread to other executables. On the other hand, a worm is a program that actively transmits itself over a network to infect other computers. Worms may also take malicious actions.

1.1.2 Trojans

A Trojan horse is any program that invites the user to run it, concealing a harmful or malicious payload. The payload may take effect immediately and can lead to many undesirable effects, such as deleting the user's files or further installing malicious or undesirable software.

1.1.3 Rootkits

Originally, a rootkit was a set of tools installed by a human attacker on a UNIX system, allowing the attacker to gain administrator (root) access. Today, the term rootkit is used more generally for concealment routines in a malicious program.

1.1.4 Backdoors

A backdoor is a method of bypassing normal authentication procedures. Once a system has been compromised (by one of the above methods, or in some other way), one or more backdoors may be installed to allow easier access in the future.

1.1.5 Spyware

Spyware is a type of malicious software that can be installed on computers, and which collects small pieces of information about users without their knowledge. The presence of spyware is typically hidden from the user and can be difficult to detect.

1.1.6 Loggers

Keystroke logging (often called key logging) is the action of tracking (or logging) the keys struck on a keyboard, typically

in a covert manner so that the person using the keyboard is unaware that their actions are being monitored.

1.1.7 Adware

Adware, or advertising-supported software, is any software package which automatically plays, displays, or downloads advertisements to a computer. These advertisements can be in the form of a pop-up. The object of the Adware is to generate revenue for its author.

1.2 Anti-Malware Mechanism

Anti- malware mechanisms involve three types.

- Prevention
- Detection
- Removal of virus

Prevention algorithm focuses on preventing the malware to infect the system. This can be done by dllinjection while auto load of processes at the boot time which occupies process' memory space.

Removal algorithm involves removing the malware from the system after it has been detected. Detection algorithm focuses on detecting the virus by extracting the executable files and storing in database and performing Machine Learning algorithms to classify them as malicious or benign files.

2 Malware Impact on Sensitive Scenarios

Consider a scenario in which a confidential email is to be sent. Or when a person is using Net banking he enters secret information such as passwords, pin ,etc. These data are vulnerable as they can be sent to the hackers via a trapdoor such as Trojan horse which is installed in the victim's system as an executable file.

Figure.1 Malware attack Scenario

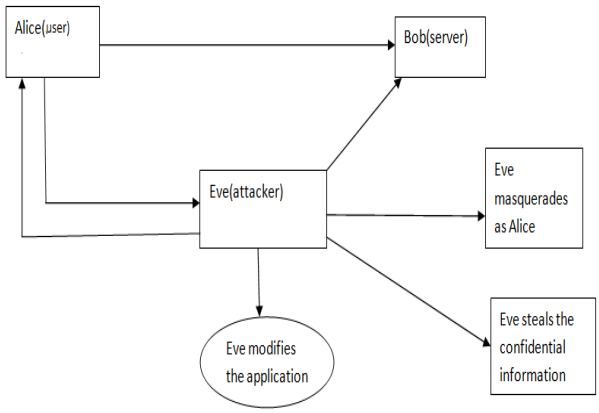
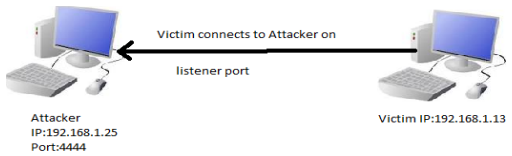


Fig.1 represents the possible threats by the attacker Eve on victim Alice.

Figure.2 Attack scenario



This is how the attacker listens to the victim through port 4444.

3 Related Works

The most widely used method for malware detection is signature-based method. Signatures are short strings of bytes that are unique to programs. The signature-based method uses a simple pattern matching approach to detect malicious code which has high accuracy. However, the signature is sensitive to slight changes in malicious code. Signature based approach cannot detect modified or previously unseen malware.
To address these machine learning approaches are used. Based on different feature representations, different kinds of classification methods, such as Artificial Neural Network (ANNs), Support Vector Machine (SVM), Naıve Bayes (NB), and Decision Tree (DT), are used for model construction to detect malware. Most of these methods are built on shallow learning architectures. [5][6][7][8]
Yu-Feng et al proposed Trojan horse detection based on system behavior using machine learning method. These machine learning methods comprise of using KNN, Naïve Bayes, decision tree and feature selection. This involves collecting data samples and storing in database and performing machine learning operations. [2]
Chen Qin-Zhang et al, also proposed a method of classification algorithms for Trojan horse detection based on behavior which involves fuzzy classification which includes data formalization, design of classification algorithm which classifies sets of Trojans based on their behavior. [3]

Igor Popov proposed an approach for malware detectionusing deep learning techniques (Convolutional Neural Network) for classification of malware. Hu and tan proposed two approaches to analyze the robustness of machine learning based malware detection algorithms. [9]
Berkat proposed a case-based reasoning approach for detecting computer virus where a new virus detected willbe automatically added to the database. Case-Based-Reasoning involves learning from experience, since where learning is done by retaining a concrete problem-solving experience than to generalize from it. [10]

4 Objectives

Signature based approaches are static and cannot detect modified or previously unseen malware. Hence dynamic approaches based on the resources consumed and processes' behavior is discussed in this paper. The principal goal is to

segregate a suspicious process or program out of several others, based on the behavior.

4.1 Detection based on Task Manager

Task Manager is probably the most well-known tool for monitoring processes. Processes are extracted from the task manager dynamically as shown in fig.4. The intercepted processes are analyzed by the anti-malware. Certain processes which do not have verified digital signature are suspicious. In some cases, process which hides in auto start location are considered suspicious. They often hide behind Rundll32 and DLLHost. Malwares get attached toauto start processes through dll injection. Processes which are launched later are also considered suspicious.

Resources consumed by the process are also monitored. It is considered skeptical if certain process uses out of bound resources. The task manager displays the CPU usage, memory, network and disk usage as shown in fig.3. If the usage goes beyond threshold limit, malware is suspected, and the user is prevented from further action. It also displays the performance of the system. If opening certain processes slows down the performance, malware is suspected, as the role of malware often includes performance degradation.

The methodologies mentioned above are a little time consuming and costly. Hence, they are not carried out every time. They take place only if the user performs certain actions which involve confidential data.

Figure 3 CPU and memory Usage

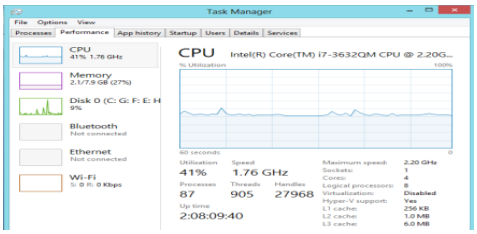
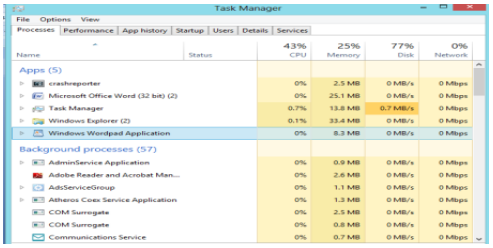
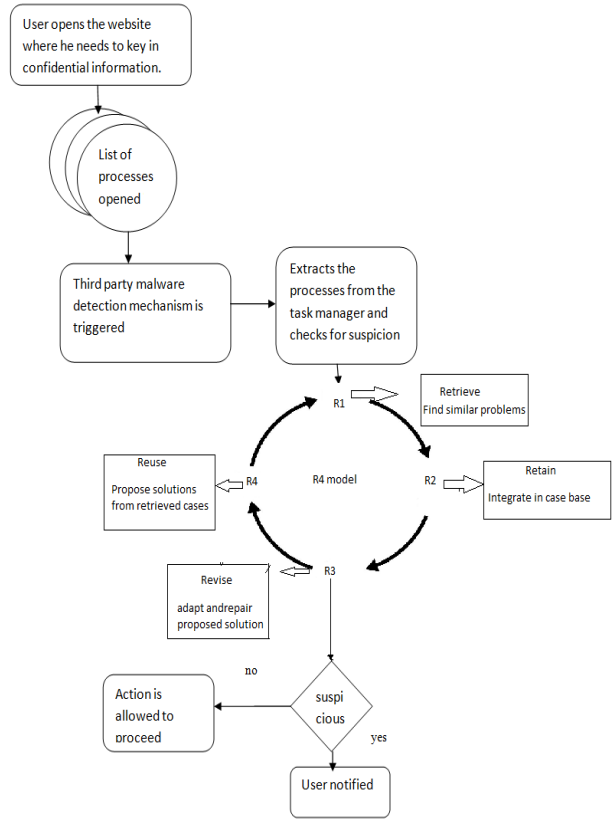


Figure 4 Processes which are currently active



2 CONCEPTUAL ARCHITECTURE

Figure 5 Conceptual Architecture



Case based Reasoning is the process of solving new problems based on similar previous problems. This is based on a four-step process, -Retrieve, Retain, Revise, reuse as shown in Fig.5

CBR has been adopted because

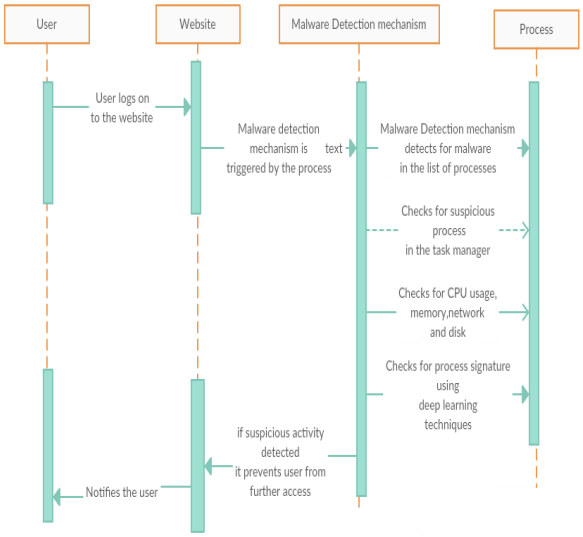
1. Detection of malware using CBR does not require explicit domain model and so elicitation becomes a task of gathering case histories about malwares.
2. Implementation is reduced to identifying significant features that describe the case i.e. rules describing the possibilities of malware, which is easier than creating an explicit model.
3. CBR systems can learn by acquiring new knowledge as cases about the malware.

4. Acquiring knowledge and applying database techniques makes the maintenance of large volumes of information easier.

3 PROTOTYPE IMPLEMENTATION

As shown in Fig.6, the third-party malware detection mechanism acts as the proxy for the malware. The malware is unable to obtain confidential information as the anti-malware system blocks the user from further using the website. It intercepts the data from the user and acts as a proxy for the malware. Hence malware is not only dynamically detected but also prevented from undesirable activities.

Figure 6 Prototype Implementation



4 CASE BASED REASONING SCENARIOS

A case-based reasoning involves a four-step process.

1. Retrieval
2. Reuse
3. Revise
4. Retain

This section represents the scenarios where there is a large possibility of downloading the malware into the system.

SCENARIO 1:

EMAIL:

Consider a situation in which a user is receiving a mail which is the source of a malware which gets automatically downloaded into the system. The user is not aware of the fact that he has downloaded a malware.

Parameters to be considered:

P1:EH. Email Header- Email headers show the route an email has taken to arrive at its destination. They also contain other information about the email, such as the sender and recipient, the message ID, date and time of transmission, subject and several other email characteristics. Spam accounts tend to hide the identity of the origin of the mail.

EH_{s}- Email Header –spam
EH_{NS}- Email Header-not spam
SI-Sender Identity-If the sender of the mail is from the address book of the recipient or not.
CO-if the sender belongs to a community or a recognized organization or an individual.
P2: MS- The size of the message sent including the size of attachments
P3: RS-Reported as spam- The if the sender of the mail had been reported as spam.
P4: MT-Missing to field
P5: ET- Certain file extensions are malicious
P6:CR- CC has more number of recipients.
RULESIf one or more of the following cases fail, then an alert is notified to the recipient. After the necessary verification, the recipient can accept or reject the mail.

Case 1: EH is forged or not recognized as a valid sender
Case 2: SI is not from the address book of the recipient
Case 3: sender does not belong to a recognized community or organization
Case 4: The average mail size of the recipient is calculated and if the size of the mail exceeds the average. (E.g. MS is not between 10 KB and 12.5 MB)
Case 5: One of the file extensions mentioned in Table 1 is the extension of the downloaded file
Case 6: RS=true
Case 7: MT=true
Case 8: CR>15

Table 1 File Extensions

File name extension	File types
.adp	Access Project (Microsoft)
.app	Executable Application
.hpj	Project file used to create Windows Help File
.hta	Hypertext Application
.inf	Information or Setup File

Table 2 Sample Data for Email

	From	To	Message Size	Attachment Extension	No of recipients in cc field	Sender from address book	Email Header verified	Detection Strategy	Prevention Strategy
1	xxx@spam.com	recipient	13 MB	.exe	3	No	No	1.Message size is greater than average size 2. File extension is .exe	Alert message is sent. User can either accept the mail or reject considering it as non-fatal
2	xxx@yyy.com	recipient	10 MB	.pdf	2	Yes	Yes	Not detected as malware	-----
3.	aaa@bbb.com	recipient	100 KB	.doc	16	No	Yes	1.Number of recipients in CC is more than 15	Alert message is sent. User can either accept the mail or reject considering it as non-fatal

Retrieval:

Given a specific case, retrieve the existing relevant cases, arrive at a solution based on solutions of similar past problems.

Table 3 Retrieval sample data

Spam	Not spam
Case 1: EH is forged or not recognized as a valid sender Case 2: SI is not from the address book of the recipient Case 3: sender does not belong to a recognized community or organization	Case 1: EH is recognized as a valid sender Case 2: SI is from the address book of the recipient Case 3: sender belong to a recognized community or organization

Revise:

Having mapped the previous similar solution to the new case if necessary

Table 5 Revise sample data

Case id	Spam	Revision strategy
C04	Yes	MS is not between 10 KB and 12.5 MB
C05	Yes	RS=true

Retain:

If the solution either mapped to the similar previous case or revised, store the resulting experience in database.

CASE 2:

SEARCHING THE WEB

In this scenario, malware may be downloaded while web surfing by automatically redirecting to the download page.

Parameters to be considered:

P1: AR-Automatically redirected to the virus page

Reuse:

Map the solution from the similar previous case by adopting solution strategy that will fit the new case.

Table 4 Reuse sample data

Case id	Spam	Reason
C01	Yes	EH is forged or not recognized as a valid sendercommunity or organization
C02	Yes	MT=true
C03	Yes	CR>15

P2: FE-File Extensions of the downloaded file

P3: DN- Verify the Domain Name of the website

Rules

Case 1: If AR=true

Case 2: If the file extensions are one of the extensions mentioned in Table 1

Case 3: If the DN is not registered.

SCENARIO 3

POP UPS/ADS

This is a scenario where ads with malicious contents may pop up while chatting or while surfing the web.

Parameters to be considered

P1: CSP-Content Security Policy- Setting up a Content-Security-Policy with reporting will actively detect and prevent unintended access to the site.

P2: PS-Pop up sites; verify the pop-up site

P3: LT-Legitimacy; verify its legitimacy.

Rules

Case 1: CSP violated

Case 2: PS not verified

Case 3: LT not verified

Table 6 Sample Data for searching the web

SNO	Website	Automatically redirected	File Extension of the downloaded file	Domain Name Registered	Detection Strategy	Prevention Strategy
1	xxx.com	Yes	..pdf	Yes	Automatically redirected.	Alert is popped up to the user who decides on what to do.
2.	yyy.com	No	.pdf	No	Domain name not registered	Alert is popped up to the user who decides on what to do
3	zzz.com	No	.exe	Yes	File extension is .exe	Alert is popped up to the user .

Table 7 Sample Data for pop ups

SNO	Verified by CSP	Pop up website verified	Legitimacy verified	Detection Strategy	Prevention Strategy
1	Yes	No	Yes	The website is not verified	Alert pop up is sent to user who decides whether the content is fatal or non-fatal
2	No	Yes	Yes	The content Service provider finds that the content is injected by a third party	Alert pop up is sent to user who decides whether the content is fatal or non-fatal
3	Yes	Yes	No	The legitimacy of the pop-up content cannot be verified	Alert pop up is sent to user who decides whether the content is fatal or non-fatal

5 ASSESMENT OF R4 MODEL

Confusion Matrix

True positives (TP): These refer to the tuples that were correctly labeled as malware.

True negatives (TN): These are the tuples that were correctly labeled as not a malware.

False positives (FP): These are the tuples that were incorrectly labeled as malware.

False negatives (FN): These are the tuples that were mislabeled as not a malware.

Figure 7 Confusion Matrix

n=100	malware	not a malware	
malware	TP=74	TN=16	90
not a malware	FN=6	FP=4	10
	80	20	

True Positive, TP=74

True Negative, TN=16

False Negative, FN=6

False Positive, FP=4

Precision=TP/(TP+FP)=74/(74+4)=0.8222

Recall=TP/(TP+FN)=74/(74+6)=0.9250

F1 score= 2TP/(2TP+FP+FN)=0.8706

6 CONCLUSION AND FUTURE WORK

The strength of the methodology lies in the fact that the task manager can track all kinds of processes. Also, it computes the CPU usage and other performance measures accurately which is the main factor in detecting and prevention of malware. The mentioned methodology is costly and hence a cheaper and more effective solution can be suggested in the future.The future work may involve methodologies which dynamically monitor the malware using cheaper solutions.

References

[1] Z. Shumei and J. Yanru, "The Model of Trojan Horse Detection System Based on Behavior Analysis," in *Multimedia Technology (ICMT), 2010 International Conference on*, 2010, pp. 1-4.

[2] L. Yu-Feng, Z. Li-Wei, L. Jian, Q. Sheng, and N. Zhi-Qiang, "Detecting Trojan horses based on system behavior using machine learning method," in *Machine Learning and Cybernetics (ICMLC),2010 International Conference on*, 2010, pp. 855-860.

[3] C. Qin-Zhang, C. Rong, and G. Yu-Jie, "Classification Algorithms of Trojan Horse Detection Based on Behavior," in *Multimedia Information Networking and Security, 2009. MINES '09. International Conference on*, 2009, pp. 510-513.

[4] Q. Jie, Y. Huijuan, S. Qun, and Y. Fuliang, "A Trojan Horse Detection Technology Based on Behavior Analysis," in *Wireless Communications Networking and Mobile Computing (WiCOM),2010 6th International Conference on*, 2010, pp. 1-4.

[5] W. NaiQi, Q. Yanming, and C. Guiqing, "A Novel Approach to Trojan Horse Detection by Process Tracing," in *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, 2006, pp. 721-726.

[6] Nari, S. and Ghorbani, "Automated Malware Classification Based on Network Behavior." *Proceedings of International Conference onComputing, Networking and Communications* (*ICNC*), San Diego,28-31 January 2013, 642-647.

[7]Firdausi, I., Lim, C. and Erwin, "Analysis of Machine Learning Techniques Used in Behavior Based Malware Detection," *Proceedings of 2nd International Conference on Advances in Computing, Control and Telecommunication Technologies* (*ACT*),Jakarta, 2-3 December 2010, 201-203.

[8] Smita Ranveerand SwapnajaHiray, "SVM Based Effective Malware Detection System", International Journal of Computer Science and Information Technologies, Vol. 6 (4), 2015, 3361-3365

[9] Igor Popov, "Malware Detection using machine learning based on word2vec embeddings of machine code instructions", Data Science and Engineering (SSDSE), 2017 Siberian Symposium on 12-13 April 2017

[10] AbdellatifBerkat, Using Case-Based Reasoning (CBR) for detecting computer virus , IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No. 2, July 2011