A SURVEY ON MALWARE PROPAGATION, ANALYSIS, AND DETECTION

[1]Mohsen Damshenas, [2*]Ali Dehghantanha, [3]Ramlan Mahmoud
[1, 2, 3] *Faculty of Computer Science and Information Technology, University Putra Malaysia*
** Corresponding Author*
damshenas@gmx.com, {alid, ramlan}@upm.edu.my

ABSTRACT

Lately, a new kind of war takes place between the security community and malicious software developers, the security specialists use all possible techniques, methods and strategies to stop and remove the threats while the malware developers utilize new types of malwares that bypass implemented security features. In this study we closely looked into malware, to understand the definition, types, propagation of malware, and detecting/defending mechanisms in order to contribute to the process of protection and security enhancement.

Keywords
Malware, Propagation, Detection, Honeypot, Obfuscation, Privacy

1. INTRODUCTION

With the escalating growth of communication and information systems, a new term and acronym invaded the digital world called as malware. It is a general term, which stands for malicious software and has many shapes (codes, scripts, active content and others). It has been designed to achieve some targets such as, collecting sensitive data, accessing private computer systems, even sometimes harming the systems. The malware can reach the systems in different ways and through multiple media; the most common way is the downloading process from the internet, once the malware finds its way to the systems, based on the functions of the malware the drama will begin. In some cases, the malware will not totally harm the system, instead affect the performance and creates overload process; in case of spying, the malware hides itself in the system, which cannot be detected by the anti-virus software, these hidden malware send critical information about the computer to the source. Based on the above challenges, it is critical to carry out an in-depth analysis to understand the malware for better detection and removal chance. This paper is organized as follows: Section two has covered the recent state of the malware security and threats through results obtained from different journals. Section three discusses about the types of malware, section four presents the malware analysis techniques. Section five studies the propagation of malware in different applications and environment, and finally section six explains malware detection techniques.

2. AND NOW

Technology has become an element key for today’s life style where both business and research worlds completely rely on the technology and its applications. However like the other side of the coin, these developments have also opened the doors for the hacking and attacking community, and within a few years the malware has become a major security threat, affecting computers and networks widely [1].

Initially, the hackers and attackers started invading others computers just for fun they did not have any serious intention to look for any great gains, until online commerce gained its popularity especially in banking, financial transactions etc, which made the hacker to get financial gains [2], this has motivated the attackers, to work more and more to keep the machines infected as longer as possible, to get more financial gains and more valued information and data [2], consequently a big challenge has emerged in terms of protecting the information and business systems and a kind of arm races have started between security products and attackers community [3].

The malware historical timeline shows that it has a lot of changes and phases since it has been discovered and detected in hosts and networks , starting from virus which is a self-replicating malware but not self-transporting [2], moving to worm, which is a self-replicating and self-transporting [4], and going more for other malware types and families . With the rapidly increasing complexity and interconnection of emerging information systems, the number of malware attacks is also increasing piercingly. While, there are a noticeable development in defense technologies and security techniques, there is also a similar development in sophisticated hacking techniques and appearance of new security vulnerabilities from day to day [5]. Due to the sequence of malware propagation, we can now clearly feel the impact of malware on various computer network infrastructures, technologies and services such as, file-sharing [6], online social networking [7], [8], Bluetooth[9], [10] and wireless Networks [11]. Many techniques have been developed and used to detect malware and prevent its propagation like sandboxing [12] and virtual environment [13] and some time the malware environment has been simulated to make it easy to detect by using FRAM model [14]. The enhancement and improvement process for security should be powerful and simultaneously move in two directions; protecting the systems from the well-known malware threats and seeking for innovative ideas and insightful analysis for handling the malware issues.

2.1. Malware Issues

Many studies, surveys, experiments, brainstorming, statistical analysis and modeling methods have been done to gain deeper knowledge and valuable information about malware [15], because the attackers are continually developing their abilities, attacking skills and techniques. In order to make the tracking and detection processes difficult, and to pose new challenges to inspectors, all these studies and works are not sufficient enough to cover the rapid increase in malware evolution. Based on our understanding Virus Bulletin (1988) was the first dedicated Journal to study the malware [2], while, now there are a lot of Journals available that are dedicated to the security issues, especially malware issues. This paper has been presented to gain understanding about the various issues related to malware. We have used much recourse to form different papers and journals, the details of the recourses that we used, will be shown in data collection part in more details.

2.2. Limitations of the Study

The publications related to this paper are more common in university libraries than in the offices of chief security officers and companies specialized in information security service such as, Norton and McAfee. Another point related to the publications of this study is, how the publications are distributed in many topics related to malware, and this will not help to dig enough for solutions and defense mechanisms, against malware attacks. It may help to clarify the picture of malware issues, but not enough for enhancement process and additional contributions. The authors were looking to the malware from different angles and viewpoints, which are great, but will confuse the readers. On the other hand the numbers of statistics provided and details analyzed are also few, to adequately sustain very significant research value. In this case, where most of the papers are too specific in their corresponding research field and purpose, it is difficult to generalize the specimen into statistical data with higher accuracy. We have also realized that most papers are from IEEE publications, and thus also acknowledged this as a form of limitation on availability of more related research publications in other sources.

2.3. Data Collection

Access, review and analyze articles covered by journals and institutes, specialists in scientific research as follows: Science Direct (service provided by the ELSEVIER Publishing company) Journal, AMC (Digital Library published by the Association for Computing Machinery), Institute of Electrical and Electronics Engineers (IEEE) and SANS Institute. The main purpose is to identify the malware issues currently being addressed by malware detectors, to gain a complete picture about malicious software. This paper is presented, based on publications and articles from 2006 until January 2012. The reason behind choosing the above journals and institutes as sources for this study is to combine and gather the academic and business fields, also these journals and institutes are primarily focusing on information security field.

2.3.1. Topics covered in the Study

The method we have followed to collect data was based on journals reviewing and analyzing. Then we have moved to gather the similar topics and ideas, and group them in specific structure as required, for instance, we categorized all topics related to malware and its propagation in different networks and environments such as, LANs, Bluetooth, and Wireless Networks under one main category called malware propagation. Another category is malware detection techniques, where we have gathered all techniques such as, anomaly-based detection, specification-based detection and signature-based detection. We have applied this method on the remaining topics covered in our resources, but we categorized all the topics that do not belong to any main category, as separate category named as, other.

During the categorization process some topics fitted into more than one category, while other topics did not overlap. Some of the overlapped topics were categorized into a category called as Other, for example in the case of virtualization, it has been categorized once as malware detection technique, and once as environment for malware propagation, the reason behind this was, going deeply into the details of keywords and abstract on virtualization papers gave us clear picture how to categorize it in the right manner. So we simply consider the keywords and abstract of the paper as the base of categorization process and if still unclear, we investigated the discussion and conclusion part to differentiate the topics.

2.3.2. Results obtained from the resources of the study.

In this part of the paper, we have highlighted the headlines and topics covered by all resources of the research. Based on the approach of the categorization, Fig. 1 illustrates the statistical distribution of the papers under the main topics of the study. Classification categories connected with each other smoothly, to integrating as one dynamic environment of attacking side that develops new strategies, to avoid detection process from the anti-side. The story of malware classification starts by developing small malware pieces to attack (.exe) files and destroy them, and now there are many types of malware, with different attacking strategies and techniques.

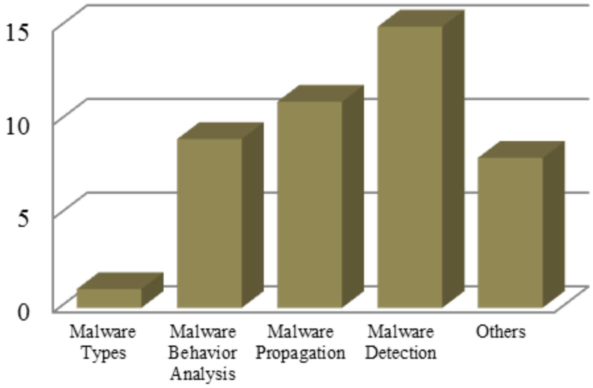


Figure 1: Main topics of the Study

Therefore, the need to develop strategies and modules to detect malware and disinfect their harmful effects is one of the most important fields of computer science. Nowadays, the malware developers are working in groups and are more adaptive, for example integrations happen between malware types, to generate new forms for specific tasks. On the other hand the attackers, using the suitable malware type, depend on the purpose and nature of attacking i.e. using spyware against android applications. Malware development does not work randomly or depend on chance, analytical strategies will be carried out by malware on their targets. The analytical data include systems structures, network connections and controllers, application features, programming languages and many other data. After analyzing the proposed data from the attackers then they start to develop new features for malware, and design new strategies to manage and plan the activities of attacking process.

The attacking strategies are developed continuously and effectively designed, to ensure the accuracy in attacking. For example, some strategies depend on two malware types, the first type used to weaken the target security level, then the second type attack the target directly. Parallel with attacking strategies, the anti-malware strategies are developed in response for detecting malware, thus we have found most of the research papers discussing about malware detection. The new ideas of detection strategies focus on developing anti malware strategies, to cover the malware development that may happen. In other words, the researchers have provided analytical studies for forecasting the malware activities. The main topics were divided into subtopics to go deep and highlight the most important terms and keywords related to malware. Appendix 1 shows the highest frequencies for the topics in malware detection and analysis.

3. TYPES OF MALWARE

Recently, the number of information security threats caused by malware has rapidly increased, which leads to urgently studying the threats and accordingly categorizing them, to simplify the process of discovering and handling them, in order to detect them and find appropriate solutions. Malware has been categorized into seventeen different types [16], in this section we have listed and discussed the main and most common categories as follows:

Virus: is a computer program that has the ability to harm and self-replicating in order to infect host; viruses are linked or attached to a software utility (e.g. PDF document). Launching the infected PDF document could then activate the virus, and a sequence of events may occur based on the function of the virus [4], [17].

Worm: another kind of harmful programs is worm, which can replicate itself and invisibly transfer through networking. The effects of worms differ from viruses as the former need help from any file, to work and mainly its effect is on networking bandwidth or sending junk emails. One example of worms is Conficker [6], [17].

Spyware: this may occur, when users download free or trial software. In this kind, the users are observed by spies; hence their passwords, account numbers and every other personal detail become vulnerable [4].

Adware: this kind usually happens, while downloading free games or it is combined and embedded with advertisements, so when we watch advertisements this embedded code is installed to our PCs. This kind aims to observe the user's activities, when using networking [4].

Trojan: this kind gives power to remote hijackers, to use your system as they wish. They may get your passwords, observe your systems or damage the system files [4].

Botnet: this kind of malware controls your systems remotely and sends spam or spyware. Most of botnets are zombie and wait for command of the party who runs it, where there are two types of botnet such as, simple or hierarchical [18].

4. MALWARE PROPAGATION

Many studies and researches focused on studying the malware propagation in the digital world, communications and computer networks, some of the modeling and experimental procedures have been followed to study the effect of malware and the way it propagates in these fields, in addition to this, the studies cover some concepts and techniques related to malware detection. The malware propagation concept refers to the electronic method, by which, malware is transmitted to an information system, platform or device it seeks to infect for example the malware can propagate through PDF files and access the host unless the user disable the JavaScript in PDF reader [19].

4.1. Through Operating System

Malware is attacking the operating systems such as Mac, Android, Windows and Linux, but not in the same level and strength because some operating systems have more defense mechanisms which don't allow the malware to achieve its design purpose [20]–[22]. In the following lines some attacks followed by malware against OSs will be highlighted to show how the operating systems act accordingly. Every year a large number of new OSs malware with stronger propagation and strategies are created.

The malware follows dynamic and adaptive propagation to attack the OS architecture such as, attack the security levels in OSs to open security threat. Another malware method propagates the OSs to infect the executable file and create virtual tasks which will slow down the OS performance [21]. Propagation of malware differs, based on OS, for example, the malware work on (.plist) Macintosh system files [22], but in Android it comes as spyware which attack the source code of Android OS [20].

4.2. Through Wireless Networks

In [23], [24], the Authors have introduced the mobile and smartphone applications and some security issues related to wireless networks. In the above studies, the Bluetooth technology has been introduced in specific project named as Blue Bag that includes a covert attack and scanning device, which demonstrates how attackers can infect and reach a wide range of mobiles and devices running a Bluetooth Technology, they have found some weaknesses in Bluetooth technology, which may allow attackers to reach the devices. In [9] the authors have explained some specific attacks that can affect the wireless communication and Bluetooth such as:

BlueSnarf: it uses the (Object exchange) push service and the attacker can access without any authentication and recently in the upgraded version of this kind of attack, the attacker can get a full access including read and writes access.

Bluejacking: occurs by sending a short tricky text message into authenticated dialog, and the users will be using the access codes of the tricky message, which allows the attacker to take control of the device.

BlueBug: the attacker will be able to use phone services, which include incoming and outgoing calls, sending and receiving SMS, etc. all through accessing the cell phone.

Blue Bump: it goes through the weakness of Bluetooth in the way it handles link keys, and it can lead to getting the data or abusing the mobile services such as internet, WAP and GPRS.

Blue Smack: it simply guides to service denial.

HeloMoto: it is a combination of BlueBug and BlueSnarf effect.

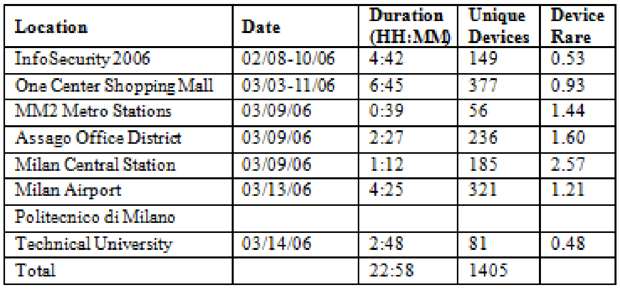
Blue Dump: the attacker will involve himself in the pairing process through Bluetooth after dumping the stored link key.

Car Whisperer: the default configuration of some devices makes the PIN code fixed for pairing and exchanging, which will make it easy for the attackers, to abuse the devices and take control of the devices accordingly once they get the PIN, which is not changeable.

Blue Chop: the attacker will get the chance to disconnect and terminate the established connection, especially when the master of the connection is supporting multiple connections.

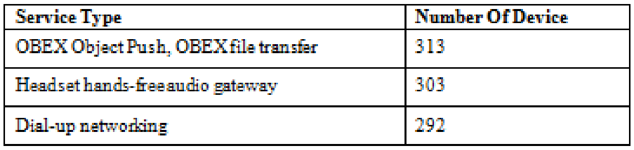
The hardware and software structure of the Blue Bag project have been illustrated with specification details and the survey results have been summarized as shown in the following table:

Table 1: Summary of the surveying results



Location	Date	Duration (HH:MM)	Unique Devices	Device Rare
InfoSecurity 2006	02/08-10/06	4:42	149	0.53
One Center Shopping Mall	03/03-11/06	6:45	377	0.93
MM2 Metro Stations	03/09/06	0:39	56	1.44
Assago Office District	03/09/06	2:27	236	1.60
Milan Central Station	03/09/06	1:12	185	2.57
Milan Airport	03/13/06	4:25	321	1.21
Politecnico di Milano				
Technical University	03/14/06	2:48	81	0.48
Total		22:58	1405	

Table 2: Services offered by mobile devices during the Survey



Service Type	Number Of Device
OBEX Object Push, OBEX file transfer	313
Headset hands-free audio gateway	303
Dial-up networking	292

The authors of this study have come out with some points and results after conducting the survey as follows:

1) Bluetooth technology is involved in many devices cell and Smartphones, PCs Notebooks, GPS Printers, palm pilots and others, which means more possibility for malware to propagate.

2) Visibility time is an important factor in the possibility of being attacked, longer time more possibility, and unfortunately some users are not aware about this point and hence keep Bluetooth on discoverable and visible mode in need and without need.

3) Social Engineering factor: 7.5% of the owners are simply careless in terms of the received files and they tend to accept the unknown files from unknown sources.

4) The survey shows that, small percentage of people are aware about the risks that they may face, when they use the new technology devices, such as, smart phones, and how this can affect their work and organizations, where the data value is high and critical, if they save it on their devices, and it is possible that, they can carry the risk with them to their organizations and work

environment, where the attackers are simply using them to reach the network or the CEO.

5) MMS messages are another way to propagate malware in addition to Bluetooth connections.

6) The Survey shows that, the technology is growing fast as against the techniques of handling security issues related to the new technology, and there is a real gap between technology and security enhancement process, which may affect the reliability and stability of the technology.

In [25], the Authors have presented a deterministic detailed analytical model, which characterizes the propagation dynamics of Bluetooth worms. The model takes into consideration, the impact of mobility patterns on the Bluetooth worm propagation and the behavior of the Bluetooth protocol. A lot of modeling processes have been done in this paper, to characterize and analyze the behavior of Bluetooth worm propagation and all related aspects as follows:

Modeling the inquiry phase, which represents the time that the infective device starts its inquiry, then considering the following points in modeling process: the number of neighbors, neighbor discovery probability, and number of inquiry responses and the duration of inquiry phase. The Next step of modeling is modeling the neighbor processing phase, where the infected device will be numbered as device 0 and all discovered neighbors from 1 to R (t) and consider the following points: the step of establishing a connection, the step of probing for infection possibility, the step of replicating the worm code and total time spent on processing all the neighbors discovered. Fig. 2 shows the flow chart of the infection cycle of Bluetooth worm.

Now as the communication channel, packet loss and data throughput are important points to work on.

In [25] the authors have specified one model for this, as modeling the Packet loss Probability and the Data Throughput. At the end of the modeling work the authors modeled the Infection Curve, by using the logistic equation with the variable pairwise infection rate, finally the authors came to the point of predicting the propagation curve of Bluetooth worms in a large population such as, Los Angeles city, supposing that all people in the city are using and carrying a vulnerable Bluetooth-enabled mobile phones and walking in the city.

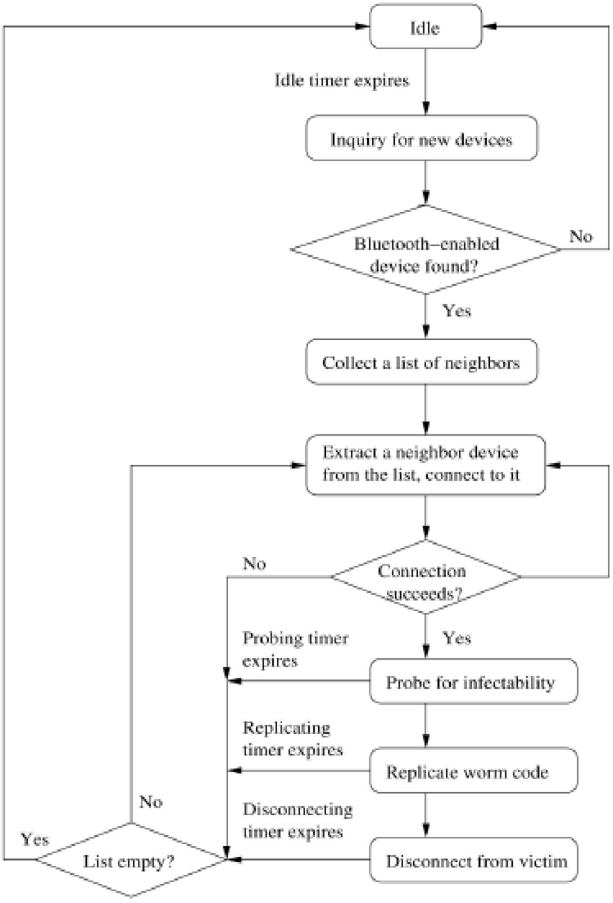


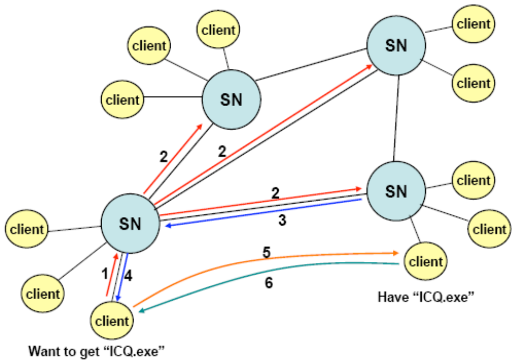
Figure 2: The Infection Cycle of Bluetooth Worm

As a result and based on the done work in this section of malware propagation, it is evident that modeling process and surveying can give a clear picture about the reality of Bluetooth malware and worm propagation, since studying the cycle of the malware and worms and their behavior will definitely guide to enhance the security level in Bluetooth technology and provide more protection to the devices, where Bluetooth and wireless technology are involved.

4.3. Through File Sharing

File sharing has become a very common application for Peer-to-Peer networking, which allows the users to share a huge number of digitally stored information, one of the most common file sharing networks is Kazaa, which has been developed in 2001, based on the Fast Track Protocol, Kazaa was subsequently under license as a legal music subscription service, but as of August 2012, the Kazaa website is not offering a music service anymore [6].

Having few number of defense mechanisms is the reason behind the vulnerability of the Peer-to-Peer file sharing networks to many security attacks ; according to this hundreds of viruses have used the P2P as a propagation vector , the authors have described how KaZaA works and shares files and explained the concept of supernode and indexing process for the hosts, where the connection between hosts is encrypted with a key exchanged at the beginning of the session , also they have discussed about Krawler (A KaZaA Crawler), which has two main components : the dispatcher, which maintains a list of super nodes and the fetcher, which is responsible for communicating with the Dispatcher , Updating process and Sending Queries. Fig. 3 below is an example of the search sequence in KaZaA.



1. client sends "ICQ" query to its parent SN
2. If parent SN can not find file, then forward query to connected SNs
3. SN that knows client which has "ICQ.exe" answers to query
4. Parent SN relays answer to the client
5. Connect to the client which has "ICQ.exe"
6. Download "ICQ.exe"

Figure 3: Example of KaZaA Search Sequence

After explaining the concepts of KaZaA, the authors have moved to the core of the research, where they have studied the propagation of malware in the P2P and KaZaA and came out with some results that supports the process of enhancing security and protection mechanisms as follows: the propagation of viruses in P2P is unlike the operation for worms, it does not send copies to the hosts by itself, instead the viruses are engaged in the process of file exchange, where the viruses start propagation. One more additional step is making multiple copies of the viral file with different names, to increase the chance of downloading and infection.

In the last stage of this study by the authors, the mechanism of Krawler has been explained in details, where the Krawler runs on three machines and was able to investigate more than 60,000 files in an hour on average, the target of this was to collect a large number of popular executable files in KaZaA network and the percentage of malicious programs. The authors then have studied the signature of the viruses and used the hashing mechanism, to confirm, whether the downloadable file is a match with the original file or if it is malicious software. The results have been taken from two datasets (Feb-06 and May-06), in both datasets same set of query strings has been used for crawling, where the Krawler issues 24 quires to each supernode (in series) and gathers the responses that may come from any peering supernodes. After explaining the results the research discussed the malware distribution and the percentage of infections for the hosts and focused on the point of viral naming mechanism and using the popular file names, to work under cover, the analysis results show that 15% of the total number of downloadable executable file samples have a viral code, and 71% of infection cases in the clients and Hosts were because of SdDrop worm and its variants and Tanked viruses.

4.4. Through Social Networking

During the last few years, online social networks have become very popular and grew tremendously as they act as platform of real-world relationship; its popularity comes from the feature of virtual interaction techniques [8].

According to [7], OSN provides the users with many services such as, sharing photos, clips, files and applications in addition to chat and call services, and the last two years have proved that, the OSNs are not only websites for communication and fun, but can contribute in the process of changing the culture and lifestyle. On the other hand in terms of security, OSNs can be considered as a perfect environment for malware and security threats. Based on the studies and researches, the attacks and threats can be categorized into four wide categories:

a) Privacy Breach Attacks: three primary parties interact with one another in an OSN: breaches from service providers, which represent the companies such as Facebook, Twitter and so on, breaches from other users and accounts owners, and breaches from third-party applications, which are involved in many stages of OSNs. On the other hand the threats related to the privacy issues can be classified as: browsing user activities, disclosing the user's identity, cyber-stalking, cyber-bullying, harassing, and slandering. The most ideal privacy level is users sharing information to only their friends or a group of users; this rule is, however, vulnerable to cybercriminals, who pose themselves as a friend using a fake name and image to gain access to all information shared by the targeted users, Nowadays, everyday hundreds of millions of users connect to OSNs from different places and using different media and devices, where control on the protection and security exists. Additionally, most current OSNs do not provide a secure communication layer and as a result of these vulnerabilities, there is a risk of sniffing tools capturing the data.

b) Viral Marketing: this refers to the techniques of marketing, including OSNs and other technologies, in OSNs viral marketing can be considered as an unwanted and good environment for malware, one of the most common examples is the spam in OSNs, in addition to the process of phishing attacks, which is considered as social engineering technique [26].

c) Network Structural Attacks: such as Sybil Attacks and here some defense mechanisms are provided such as, trusted certificates, resource testing and recurring costs.

d) Malware Attacks: one of the most common attacks is the attack of a worm known as Koobface worm.

4.5. Through Virtualized Systems

Virtualization technique is quickly becoming a standard technique for business. The technology lets one computer or server run multiple operating systems, or multiple sessions of an operating system at the same time, which lets users run many applications and functions on a single computer or server, instead of running them on different machines as in the old technology.

The biggest challenge faced by organizations now is, how to secure the virtualized system, which are vulnerable to the same type of threats as real systems. Virtualized systems cannot always be secured by the same technique as real systems, because each virtualized system on the same machine may face different threats and need different security levels, and we need additional security techniques, to secure the channels between the virtualized systems on the same machine.

In [13], the author has studied several virtual system security problems. He has traced the virtual system history from security point of view and has identified that virtualization creates new security challenges for organizations, and the administrator must ensure that every single virtualized system follows the rules and policies of the organization such as, limiting access to some data and applications. An important example of new problem created by using virtualization is that, the network-based security system does not usually trace the communication between two virtual machines installed on the same server. The author has also presented the importance of security zones, to enhance the level of security for virtual systems. In security zones the host server divides the virtualized systems into zones; each zone has its security level, depending on the requirements of virtualized systems.

In [27], the authors have focused on detection and mitigation techniques for the most famous VME product nowadays, VMware. They have presented two methods used by malware to detect VMware. The first method is related to VMware communication channel. The communication between host and guest operating systems occurs via a custom communications channel hard-coded into all products of VMware. The guest and host operating systems work together during this channels for a range of functions, including enhanced GUI performance, support for data moving in and out of the host clipboard, and files dragging and dropping from guest and host and vice versa. The authors have discovered a sample program with a small piece of code that checked for the presence of this type of communications channels. The second method to detect VMware exist is the Red Pill techniques. The physical memory is shared by the operating system of guest, which is virtualized by software run by the operating system of the host, a VME usually introduces some differences in the location of memory global items mapping. Like the locations of (IDT) the Interrupt Descriptor Table, and (LDT) the Local Descriptor Table to map the host and guest operating systems. The malware can detect VMware by looking at the new memory location. Red Pill was the first released tool that used this technique.

4.6. Through Email Communications

There are many ways to attack emails, which affects the sending emails (email backscatter) i.e. spam emails using viruses or worms. For that, we need to inform the sender about the real reasons for not receiving email from the other side. The attackers intercept the email, and delete the sender's address, therefore the email gets spammed and the receiving process fails, thus the sender receives a failing note message and he/she cannot determine the real reason of failure.

The email spam propagation can be analyzed by many factors such as the period of time between sending the email and sending back the failing report for the sender, another factor is the returned message which does not contain a real failing reason i.e. the system is down at this time please try later again [28].

5. MALWARE DETECTION TECHNIQUES

Since malware has different types, behaviors and different level of risk, the same detection methods and mechanisms cannot be used in all cases. It is impractical to have just one security software to efficiently handle the malwares. Hence having different detection methods for different environments becomes unavoidable. This study had focused on the most common and powerful techniques such as honeypot, honeynet, virtualization (partial and full), sandboxing and behavior operation sets. A massive experiment had been done by Taiwan malware analysis net (TWMAN), it was based on virtualization concept and client-server model, the experiment added a great value to the field of malware detection since it was able to detect many malwares which were not detectable by normal detection methods, going forward, we can clearly see that the detection process needs more computer processing power and advance techniques to make sure that the nature and behavior of malware are clear and covered from all the angles and views.

5.1. Anomaly-Based

Anomaly-based detection looks for unexpected or abnormal behavior indicators, which indicate the presence of malware. In more detail, anomaly based detection creates a baseline of expected operation. After this baseline has been created, any different form of baseline is recognized as malware. We have identified that the anomaly based detection technique uses the previous knowledge of what is known as normal to find out what is malicious. A special type of anomaly based detection techniques is specification based detection. A specification based detection uses set of rules to determine what is considered as normal, with the purpose of making a decision about the maliciousness of the program that breaches the rule set. The basic limitation of the specification based system technique is the difficulty to correctly determine the program or system behavior [4].

5.2. Honeypots

The traditional methods for detecting and preventing malware, like using anti-virus can only detect the malware with the same features. In this method, security vendors build pattern files, which contain the features of malware that have been already collected and analyzed. However, it is not possible to detect malware with different features and characteristics, especially with increasing the variation in the ratio of malware [29].

To solve this problem, [30] proposed honeypots techniques, to investigate and analyze the distribution of malware to websites. Honeypots can collect malware attacks which particularly target web applications' vulnerabilities. There are two types of web honeypots, high-interaction and low-interactions. Low-interactions type does not have actual web vulnerabilities, but simulate applications and OS performance, while high-interaction type has actual vulnerabilities which already installed to honeypot. The authors have used two techniques to investigate the ratio of anti-virus software detection. They have chosen six server protection software, from different security vendors. Fig. 4 illustrates all six anti-virus software updated by last pattern files, and the malware collected from September 2009 to January 2010 by the web honeypot. Table 3 shows the information attack information that was collected by web honeypots.

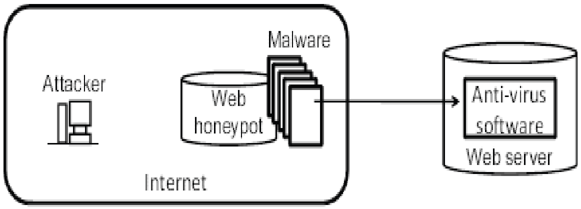
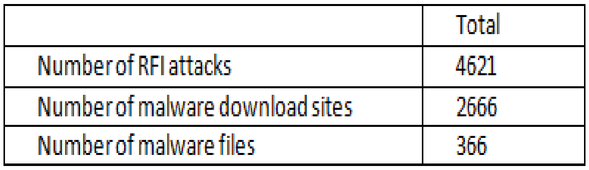


Figure 4: First investigation

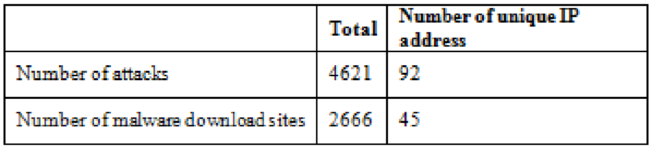
Table 3: Honeypots attack information



	Total
Number of RFI attacks	4621
Number of malware download sites	2666
Number of malware files	366

In the first investigation, the malware detection ratio of the six anti-virus tools was immediately checked and the average detection value was just 3.13%. In the second investigation the malware detection ratio of the six anti-virus tools was checked after 4 months and the average detection value was 39.8%. From these results it is evident that we cannot prevent the malware from infecting our computers despite using anti-virus software. The good note was the appearance frequencies of IP addresses for the source of attack and malware download sites. Table 4 shows IP frequencies for the source of the attack and the malware download sites.

Table 4: IP frequencies for attack and malware download sites



	Total	Number of unique IP address
Number of attacks	4621	92
Number of malware download sites	2666	45

From this result, we have identified that, only 92 malware attacks have unique IP address, and also only 45 unique malware download sites were used for attacks, which means 98% of malware information has reappeared. The traffic patterns such as a source IP attacker address and the other information that was collected by honeypots are very useful to detect and investigate malware.

In [31], the authors have defined the honeypot as a trap to detect or deflect unauthorized access to the system. A honeynet is a network that contains more than one honeypots. The honeynet aims to invite attacker, then its activities and features can be considered and analyzed to increase network security. The honeypot/honeynet typically has real services and applications thus it appears to the attackers as a normal network and valuable object. Fig. 5 shows an example of honeynet structure.

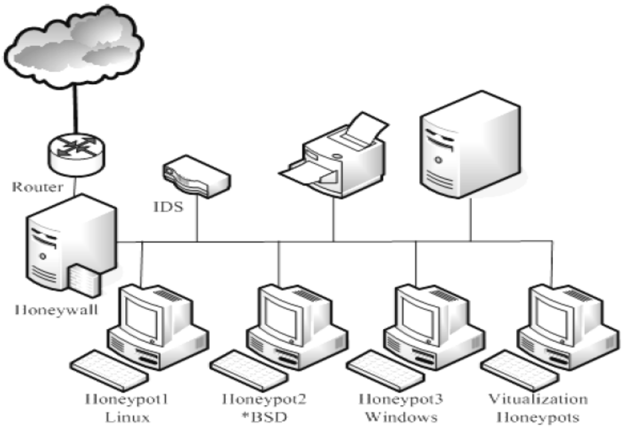


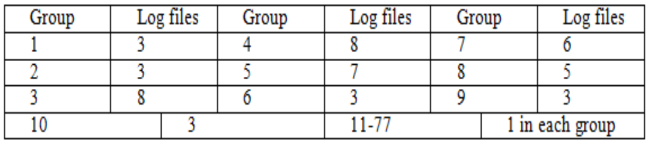
Figure 5: Honeynet structure

The effective design of a honeynet is called as multi-agents system. This system uses 3 kinds of agents. The first and second agents work in honeynet, while the third consider and examine the production network. The first and second agents collect malware information and try to recognize them using anti-virus software. The third agent uses the collected data to delete the malware from the production network or at least restrict its operation.

In [32], the authors discuss a client honeypots and the benefit of applying automated state machine to the client honeypots. The client honeypots visit and access the suspect websites with the purpose of detecting and bringing the malware information. Malicious websites may cause many activities to occur in a victim's system and each activity is done in different stages. The state machine is used to characterize the actions performed by the malicious websites into predefined states. Then these states are used to summarize the interactions with the malicious websites using the same structure of the state machine. The states are then applied to a clustering algorithm to assembly similar malicious websites with the aim of understanding how to develop the software to get better response to these attacks. The outputs of the clustering algorithm are classified to build up similar state groups that describe the malicious actions performed on the victim's system. The advantage of using this procedure is to build behavior families (each family has the same malicious characteristics) which will lead to develop common ways to deal with such exploits.

In [32], proposed an experiment for using automated state machine to detect the malicious websites. They used Capture-HPC as a client honeypot to scan the sites (which called URLs) provided to find the log files. Then these log files will be converted to the state machine file structures using client honeypot state machine (CHSM) tool. At the next stage the clustering algorithms will be applied to the generated state transition to discover the similarities in different attacks and group these attacks using these similarities. The experiment was done using 116 Capture-HPC log files. They got these log files by scanning different 116 websites. The clustering result is illustrated in table 5.

Table 5: Web based groups of exploiting experiment



Group	Log files	Group	Log files	Group	Log files
1	3	4	8	7	6
2	3	5	7	8	5
3	8	6	3	9	3
10	3		11-77		1 in each group

By grouping similar files together the authors have reduced the time of analyzing malicious websites activities, they got 77 main groups rather than 116, which mean that they have reduced 0.336 of needed analysis time.

In [11] the authors propose a Bluetooth honeypot technique and call it bluebat. By using bluebat the authors aims to provide new means to understand both existing and emerging threats that target wireless and Bluetooth networks (PANs).

5.3. Sandboxing

Previously, we had shown that the malware can exploit VME to propagate between VM hosts. In this section we will present how the VME can be used to detect malware and prevent its propagation.

There are many methods to prevent malware from detecting VME. In [27] [33], the authors have discussed two mainly useful methods to prevent the most popular VME detection techniques used by malicious attacker from detecting the VME and mitigate malware effect. The first method is undocumented VMware options. VMware VMX configuration files contain many parameters that can be changed by the administrator of VMware to set the guest machine. Some of these configuration files are well-known and documented. After many experiments, a lot of undocumented configuration files were and the amazing result was that changing some parameters in this undocumented configuration files can prevent or control the behaviors that allow malware to detect VMware. For example Jerry.c can be prevented from being detected by VMware by setting VMX file parameters as in the following program snippet:

- Isolation.tools.getPtrLoc
- Ation.disable="TRUE"
- Isolation.tools.setPtrLoc
- Ation.disable="TRUE"
- Isolation.tools.setVersio
- n.disable="TRUE"
- Isolation.tools.setPtrVersio
- n.disable="TRUE"

The changes in VMX configuration files can prevent many of current detection techniques, but the functionality and ease-of-use of guest machine will be affected, such as copy-paste via clipboard and drag-drop; because of this undesirable effect, the search for alternate techniques to prevent VMware detection is carried out. The alternate techniques are called as altering the magic value. In this method the VMware binary executable file is patched to disable or change the magic value of VMX that is related to the communication channel.

In [18], the authors have presented the important rule of sandboxing, partial virtualization, and full virtualization in combating malware. In April 2010 a study by Cyveillance showed that, current antivirus programs are not effective in discovering the threat. They have examined 13 of the most well-known antivirus products, and found that the average of malware that was detected on one day after the malware became known is only 19%, and also, the average detection rate for all 13 products only reached to 61.7% on average after 30 days. Antivirus programs are still an essential part of computer security, but it is very clear that they do not have the enough ability against a threat that continuously produces thousands of new malware day after day.

To deal with the big gap left by antivirus programs, new classes of computer security products that use sandboxing application and virtualization have been developed. High-profile applications that currently use sandboxing include Google Chrome browser, Adobe Reader X, and Internet Explorer in Protecting Mode. Separating untested code from the system using some type of a sandbox can considerably mitigate the malware by preventing malicious behavior from affecting the other computer programs.

Full virtualization gives a high management level of the vulnerable application, without changing requirements to the application itself. If any part of the application is affected by malware, the attacker can only gain access to the guest resources, programs, environment's data, and OS, but not the original host's. Simple hardware virtualization does not give a secure solution; one must satisfy the some points for a secure confinement solution like network and host isolation, real-time detection that controls unseen attacks, and fast complete clean state recovery when malware is detected. Appendix 2 shows sandboxing and virtualization, to address the malware and compares them in terms of protection level and ease of deployment.

In [34], the authors have designed an experimental model to the analyze malware behavior in real environment, as the authors have observed many differences between real environment and the virtual environment. There are many anti-VM applications to prevent analysis and discover malware in a VM environment. This experimental model represents the implementation of the Taiwan Malware Analysis Net (TWMAN), which represents a real operational environment for analysis and report malware behavior. Fig. 6 shows the flowchart for TWMAN model. TWMAN is a client-server model and configured to automatically run the analysis. The Linux operating system is installed on the server, while Microsoft Windows is installed on the client. The client downloads malware from the repository of Linux server and collects the information about changes in registers and files like image of dump memory, then the client has to restart and save the infect image of windows as an image file in Linux server.

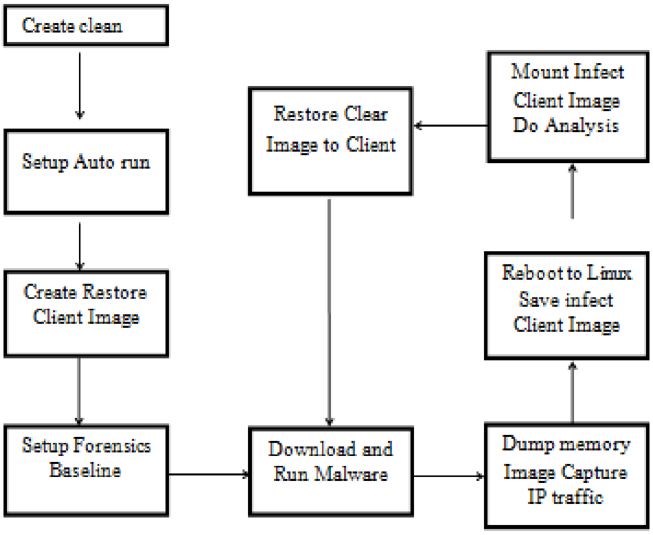

Create clean
Setup Auto run
Create Restore Client Image
Setup Forensics Baseline
Download and Run Malware
Dump memory Image Capture IP traffic
Reboot to Linux Save infect Client Image
Mount Infect Client Image Do Analysis
Restore Clear Image to Client

Figure 6: flowchart of TWMAN model

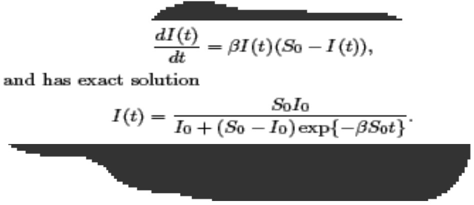
This procedure is repeated 4840 times and then the result was analyzed and reported. TWMAN can detect a lot of malware behavior which cannot be detected by VM environment and sandbox environment.

5.4. Mathematical Models

In [10], differential equation and stochastic variance have been used to express the behavior of internet worm and modeling process, the model focuses on times of next infection (TNI), which is used to explain and clarify the variance. The paper contributed in this field by validating the infection times of the (TNI) with respect to oriented scanning model, based on the structure of Red Code, and to experimentally evaluate the variance using commonly used metrics for the process of detecting worms. Based on the results of the experiments, it shows that the level of variance is tremendously high; this variance should be taken in consideration.

The element key for the research is the process of modeling worm behavior; the authors have explained different detection mechanisms and clarified the expected results for each mechanism. The following are the contributions, which have been presented by them: First, doing the validating process for a Markov chain model of the infections based on specific scanning model. Second, use the validated mode to calculate and compute the standard deviation and the mean of diverse metrics, which is related to the process of detecting the worm to show and clarify the occasional tremendous variability. The modeling process has been done in the following sequence:

Firstly: Modeling and worm detection using the following equation:



[redacted]

$$\frac{dI(t)}{dt} = \beta I(t)(S_0 - I(t)),$$

and has exact solution

$$I(t) = \frac{S_0 I_0}{I_0 + (S_0 - I_0)\exp\{-\beta S_0 t\}}.$$

[redacted]

Figure 7: An equation proposed for detection of worm behavior

Where I (t) is the state, which is the number of infected hosts at the time (t) , S0 is the number of original susceptible hosts, and I0 is the number of initial infections and β is the pairwise interaction rate between a given infected host and a given susceptible host.

Secondly: Direct Simulation Model: (DS) Model, in this model the worm will be using 100 threads on an infected machine or client, each thread at random will sample one IP address and send TCP SYN to selected target and wait either till receiving the TCP SYN ACK or getting out of the time and no response, after that the handshaking process will be completed and the thread will send an infection packet to the target, then move to work with another target and so on.

Thirdly: Time of Next Infection: next step involves in developing a different model, which considers the behavior of the secluded thread in the DS model, the concept of inner and outer cycles have been used as a way to think about its actions. The Inner cycle; is the sequence of targets on which the thread times out will end once the target responds to the request, and the outer cycle is a sequence of inner cycles, which ends with the first successful infection process to the host. The TNI model has a computational advantage in comparison with DS model, where, the events in the simulation process for the TNI occur only in infections.

Fourthly: Worm Detection Metrics: many different mechanisms and techniques have been designed in order to detect worms, some of them are based on the observation of the scanning behavior of the worms [15], and some others are based on the content [1], [5].

6. MALWARE ANALYSIS

Analyzing a malware is the inspection of the malware from its signature or behavior, to discover the attributes and functionalities of the malware; and to find out the source, target range, propagation approach and defense mechanisms of the malware. The result of these inspections helps increasing the security of the end users by providing better security through products like anti-viruses, intrusion detection systems and firewalls. Antivirus software usually maintains a virus signatures repository, which contains the binary patterns characteristic for the malicious codes. This software checks the files that are assumed to be infected for the existence of a virus signature. This detection method worked effectively until creator of malware started writing polymorphic and metamorphic code. These modifications of malware code enabled them to avoid detection by using encryption techniques, to prevent signature based detection. Security products and virus scanners look for the sequence of characteristic bytes (signature), to recognize the malicious code. The detector is determined by the detection techniques. A good quality malware technique should be able to recognize malicious codes that are embedded and hidden in the original program, and should be able to detect new unknown malware. Most of commercial antivirus software does not have the required flexibility, to detect new attacks, because the writers of malware always create new obfuscation techniques, to cheat the detection software, so that the malware can avoid detections. A Malware detector is defined as shown in the following function. The domain of detector is the set of programs ‘P’, and the range is the set of {malicious, benign} [5].

```
D(p)=‟malicious″ if p contains malicious code, or ‟benign″ otherwise.
```

The program is scanned by the detector to check if the program is benign or malicious. The test aims to find out false negative, false positive, or hit ratio. The malware detector uses the malware signature to detect the malware. The machine code binary pattern of a virus is called a signature. Antivirus programs compare their virus signature database with the files on the hard drive; removable devices (including disk boot sectors), RAM, and the data propagate to the systems through the network. Security vendors update the signature database repeatedly and make it available to customer users via their websites. The result of detection function can be classified into one of the following three categories [4]:

a) False positive: this results when a virus scanner incorrectly detects existence of the virus in a non-infected file. False positives occur when the signature used to detect the virus is not exclusively for this virus, because the signature appears in legal or non-infected software.

b) False negative: this results when a virus scanner cannot detect a virus existence in an infected file. The antivirus scanner may not succeed in detection of the virus because this virus is very new and its signature is not yet available, or it may fail to detect it because the configuration settings for that virus is dynamic and very complex and the ability of the detector is less than the robust of the virus.

c) Hit ratio: this results when a malware detector takes the correct decision and detects the malware as the malware signature matches with the stored signature.

6.1. Malware Behavior

Behavior based detection techniques study and analyzes the behavior of suspected or known malicious code, such as destination and source addresses of this code, and the way in which, the code was attached. Behavior based detection technique differs from the other scanning techniques as it considers the action performed by the malware, rather than the binary pattern. The programs with different binary content but having same behavior are collected. These types of detection techniques help in detecting the malware, which keeps on generating new signature versions, because they will always use the recourses of the system in the same manner. The behavior detector collects the data, interprets the data, and then applies the matching algorithm [1], [35].

In [1], the authors have proposed new techniques to extract and detect malware behavior. They have analyzed the behavior of 236 popular malwares. About 67% of malware produces sub-process when executed. Some malicious behaviors appear after malware execution, like thread injection and self-delete. These malicious behaviors are called as Malicious Behavior Feature (MBF). The authors have present the term Behavior Operation Set (BOS), which defined by file actions (e.g. read, rename), process actions (e.g. terminate, create), network action (e.g. TCP, UDP), and registry actions (e.g. open key, query value). These four operations were used to extract and investigate the behavior. The authors did two tests. In the first test 328 of non-affected files were tested, the result showed that only 7 of them were falsely detected as malware, then the error rate is 2.13 and the accuracy rate is 97.87%. In the second test, the authors tested suspected file and detected new malware, by observing the common behaviors with popular malwares.

In [14], the FRAM model had been proposed to make a malware forensic repository for the purpose of malware analysis. FRAM is mixed from open source tools and commercial tools which integrated together to propose an automated system. This automated system aims to reduce the time needed to handle the new malware and increase the rate of success reverse engineering. In [29], the authors proposed MalTRAK technique which is a framework for tracking and removing either unknown or known malware. In MalTRAK the users can run any program without asking for any policy or rules, but MalTRAK guarantee that the user can recover the clean state if the infectious state were found. MalTRAK can satisfy this by storing many logical views during the program run time. The draw back with the MalTRAK model that the extra overhead in disk space and run time, but using this model we have a very good recovery result to clean state in case of infection.

In [36], the authors tried to prove how the segmentation and partitioning of malware into the disconnect process can let this malware to propagate through the system. The authors showed that the separated malware pieces can reassemble together and maliciously infect without any detection form more than 40 anti-malware programs; thus the authors suggest that the malware detector must take into consideration the multi process malware behavior to reassemble different malware pieces depending on this properties and behaviors. In [37], the authors show the importance of integration between different security technologies like intrusion detection, antivirus software, and firewalls. They proposed a detection technique which depends on cloud computing and called it uCLAVS. They showed how the integration between many search engines raised the malware detection ratio to 97% while the higher detection ratio of any single engine before the integration was only 80% uCLAVS.

6.2. Malware Signature

Normal antivirus software look for signatures, which are a sequence of bytes in the malware code to state that if the program scanned, is malicious or not. Essentially, there are three types of malware: basic, polymorphic, and metamorphic malware. In basic malware, the malware developer changes the entry point of the program. Polymorphic viruses alter themselves, while leaving the original code unchanged. A polymorphic virus contains an encrypted malicious code beside the decryption part. This virus is enabled by a polymorphic engine, which is included in the body of the virus. The polymorphic engine generates new versions every time it is run; thus it is very difficult to detect this type of virus by signature based detection techniques. Metamorphic malware use advanced obfuscation techniques, to reprogram itself therefore the children and parent signatures are very different. It is not possible to detect this type of malware without disassemble the virus file [4], [38], [39].
There are many problems associated with the signature based detection technique. The biggest problem is that, the signature generation is a very complex process and requires a strong code analysis algorithm. The second problem is that the signatures are distributed as fast as possible. The third problem is that, new signatures can easily bypass the detectors, and the final problem is that, the size of signatures repository is increasing day by day.

6.3. Obfuscation and Normalization

It is a technique used by software developers and writers targeting to hide the details of their products so that the reverse engineers can't find the correct code, it has been used as an advantage by the malware writers to achieve the same goal, obfuscation can be achieved by different operations and easily can make changes in the signature of malware in order to make the process of detecting the malware very difficult. Fig. 8 shows the obfuscation process [4], [40]. Given a program P and a transformation function T generates program P' such that the following properties hold true:

- P' is difficult to reverse engineer.
- P' holds the functionality of P.
- P' performs comparable to P

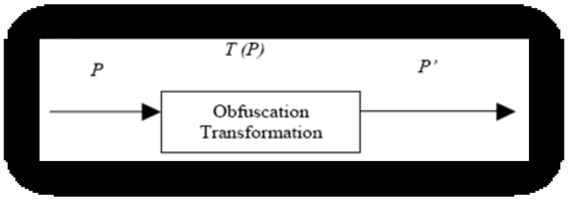


Figure 8: The obfuscation process

The obfuscation techniques can be done in different methods, starting from inserting some (no operation) instructions and inserting (push-pop) x, which known as dead-code because nothing will be achieved and accomplished and inserting some instructions for branching unconditionally, moving to inserting process for some registers and substituting instructions, all of these methods will guide to obfuscate the code of the malware and make the process of detection difficult to malware scanners. Malware normalization can be identified as a process and mechanism to detect the obfuscated copies of malware and increasing the rate of catching the malware by the detector, the output of the normalization will be the original signature of the malware which has been obfuscated and accordingly the signature will be compared to the signatures to verify it, then it will be saved in the list of known signatures in order to decrease the time of scanning and detecting next times.

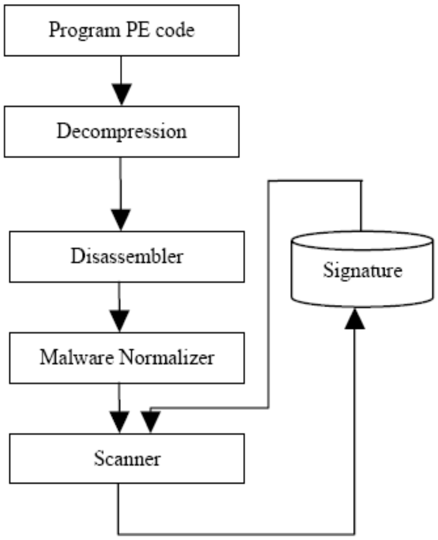


Figure 9: Normalization process flow chart

The normalization process can be done through some steps as follows: decompressing the binary code of malware, then disassembled it and pass it to the normalizer to eliminate the obfuscation and get the original code, finally passing the normalized code to malware detector in order to get out the signature, compare it with the available list and get the matched one.

Since the signatures of malware are long and take many comparisons times to detect them there was a need for additional procedures such as API Procedure to reduce the time of normalization and detection process. The key to enhance the process of malware detection based on signatures is via developing better disassembler and better algorithm for analyzing the similarity [4].

7. CONCLUSION AND FUTURE WORK

The malware developer tries to write new techniques and strategies to hide the malicious code and infect the targets. On the other hand, the detectors analyze malware behaviors continuously and try to resist these techniques and strategies hence, we need to allow detection development techniques to lead malware updating through very well analytical process for malware activities and behaviors to fix any possible targeted threats. A new simulation must be designed to contain real system samples, to analyze the malware behaviors against these samples after elaborate malware updating. The objectives of this simulation are to avoid systems threats before being infected by real malware.

ACKNOWLEDGEMENT:

This work has been supported by the Research University Grant Scheme (RUGS) University Putra Malaysia (UPM) project 9371600 (0502122147RU). The views and opinions expressed in this article are those of authors alone and not the organizations with which authors are or have been associated or supported.

REFERENCES

[1] L. Wu, R. Ping, L. Ke, and D. Hai-xin, “Behavior-Based Malware Analysis and Detection,” in *Complexity and Data Mining (IWCDM), 2011 First International Workshop on*, 2011, pp. 39–42.

[2] R. Ford and W. H. Allen, “Malware Shall Greatly Increase...,” *Secur. Priv. IEEE*, vol. 7, no. 6, pp. 69–71, 2009.

[3] J. R. Crandall, R. Ensafi, S. Forrest, J. Ladau, and B. Shebaro, “The ecology of Malware,” in *Proceedings of the 2008 workshop on New security paradigms*, 2009, pp. 99–106.

[4] P. Vinod, R. Jaipur, V. Laxmi, and M. Gaur, “Survey on malware detection methods,” in *Proceedings of the 3rd Hackers’ Workshop on Computer and*

Internet Security (IITKHACK'09), 2009, pp. 74–79.

[5] J.-H. Park, M. Kim, B.-N. Noh, and J. B. Joshi, "A Similarity based Technique for Detecting Malicious Executable files for Computer Forensics," in *Information Reuse and Integration, 2006 IEEE International Conference on*, 2006, pp. 188–193.

[6] S. Shin, J. Jung, and H. Balakrishnan, "Malware prevalence in the KaZaA file-sharing network," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 333–338.

[7] H. Gao, J. Hu, T. Huang, J. Wang, and Y. Chen, "Security issues in online social networks," *Internet Comput. IEEE*, vol. 15, no. 4, pp. 56–63, 2011.

[8] S. Mohtasebi and A. Dehghantanha, "A Mitigation Approach to the Privacy and Malware Threats of Social Network Services," in *Digital Information Processing and Communications*, Springer, 2011, pp. 448–459.

[9] L. Carettoni, C. Merloni, and S. Zanero, "Studying bluetooth malware propagation: The bluebag project," *IEEE Secur. Priv.*, vol. 5, no. 2, pp. 17–25, 2007.

[10] D. M. Nicol, "The impact of stochastic variance on worm propagation and detection," in *Proceedings of the 4th ACM workshop on Recurring malcode*, 2006, pp. 57–64.

[11] S. Zanero, "Wireless malware propagation: A reality check," *Secur. Priv. IEEE*, vol. 7, no. 5, pp. 70–74, 2009.

[12] C. Greamo and A. Ghosh, "Sandboxing and virtualization: Modern tools for combating malware," *Secur. Priv. IEEE*, vol. 9, no. 2, pp. 79–82, 2011.

[13] S. J. Vaughan-Nichols, "Virtualization sparks security concerns," *Computer*, vol. 41, no. 8, pp. 13–15, 2008.

[14] J. Van Randwyk, K. Chiang, L. Lloyd, and K. Vanderveen, "Farm: An automated malware analysis environment," in *Security Technology, 2008. ICCST 2008. 42nd Annual IEEE International Carnahan Conference on*, 2008, pp. 321–325.

[15] M. Apel, C. Bockermann, and M. Meier, "Measuring similarity of malware behavior," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, 2009, pp. 891–898.

[16] M. F. Zolkipli and A. Jantan, "An approach for malware behavior identification and classification," in *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, 2011, vol. 1, pp. 191–194.

[17] M. R. Rieback, P. N. Simpson, B. Crispo, and A. S. Tanenbaum, "RFID malware: Design principles and examples," *Pervasive Mob. Comput.*, vol. 2, no. 4, pp. 405–426, 2006.

[18] B. Stone-Gross, M. Cova, B. Gilbert, R. Kemmerer, C. Kruegel, and G. Vigna, "Analysis of a botnet takeover," *Secur. Priv. IEEE*, vol. 9, no. 1, pp. 64–72, 2011.

[19] D. Stevens, "Malicious PDF documents explained," *Secur. Priv. IEEE*, vol. 9, no. 1, pp. 80–82, 2011.

[20] J. Boutet, "Malicious Android Applications: Risks and Exploitation," *Inst. InfoSec Read. Room*, vol. 2, 2010.

[21] A. J. O'Donnell, "When malware attacks (anything but windows)," *Secur. Priv. IEEE*, vol. 6, no. 3, pp. 68–70, 2008.

[22] J. Yonts, "Mac OS X Malware Analysis," *Inst. InfoSec Read. Room*, vol. 2, 2009.

[23] A. Dehghantanha, N. I. Udzir, and R. Mahmod, “Towards data centric mobile security,” in *Information Assurance and Security (IAS), 2011 7th International Conference on*, 2011, pp. 62–67.

[24] S. H. Mohtasebi, A. Dehghantanha, and H. G. Broujerdi, “Smartphone Forensics: A Case Study with Nokia E5-00 Mobile Phone,” *Int. J. Digit. Inf. Wirel. Commun. IJDIWC*, vol. 1, no. 3, pp. 651–655, 2012.

[25] G. Yan and S. Eidenbenz, “Modeling propagation dynamics of bluetooth worms (extended version),” *Mob. Comput. IEEE Trans. On*, vol. 8, no. 3, pp. 353–368, 2009.

[26] S. Abraham and I. Chengalur-Smith, “An overview of social engineering malware: Trends, tactics, and implications,” *Technol. Soc.*, vol. 32, no. 3, pp. 183–196, 2010.

[27] M. Carpenter, T. Liston, and E. Skoudis, “Hiding virtualization from attackers and malware,” *Secur. Priv. IEEE*, vol. 5, no. 3, pp. 62–65, 2007.

[28] C. P. Fuhrman, “Forensic value of backscatter from email spam,” in *Digital Forensics and Incident Analysis, 2008. WDFIA’08. Third International Annual Workshop on*, 2008, pp. 46–52.

[29] A. Vasudevan, “MalTRAK: tracking and eliminating unknown malware,” in *Computer Security Applications Conference, 2008. ACSAC 2008. Annual*, 2008, pp. 311–321.

[30] T. Yagi, N. Tanimoto, T. Hariu, and M. Itoh, “Investigation and analysis of malware on websites,” in *Web Systems Evolution (WSE), 2010 12th IEEE International Symposium on*, 2010, pp. 73–81.

[31] M. Szczepanik and I. Jóźwiak, “Detecting New and Unknown Malwares Using Honeynet,” *Adv. Multimed. Netw. Inf. Syst. Technol.*, pp. 173–180, 2010.

[32] Y. Alosefer and O. Rana, “Clustering client honeypot data to support malware analysis,” *Knowl.-Based Intell. Inf. Eng. Syst.*, pp. 556–565, 2010.

[33] F. Daryabar, A. Dehghantanha, N. Udzir, N. Fazlida binti Mohd Sani, and S. bin Shamsuddin, “Towards secure model for SCADA systems,” in *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*, 2012, pp. 60–64.

[34] H.-D. Huang, C.-S. Lee, H.-Y. Kao, Y.-L. Tsai, and J.-G. Chang, “Malware behavioral analysis system: TWMAN,” in *Intelligent Agent (IA), 2011 IEEE Symposium on*, 2011, pp. 1–8.

[35] Q. Jiang, X. Zhao, and K. Huang, “A feature selection method for malware detection,” in *Information and Automation (ICIA), 2011 IEEE International Conference on*, 2011, pp. 890–895.

[36] M. Ramilli, M. Bishop, and S. Sun, “Multiprocess malware,” in *Malicious and Unwanted Software (MALWARE), 2011 6th International Conference on*, 2011, pp. 8–13.

[37] C. A. Martínez, G. I. Echeverri, and A. G. C. Sanz, “Malware detection based on cloud computing integrating intrusion ontology representation,” in *Communications (LATINCOM), 2010 IEEE Latin-American Conference on*, 2010, pp. 1–6.

[38] M. Ramilli and M. Prandini, “Always the Same, Never the Same,” *Secur. Priv. IEEE*, vol. 8, no. 2, pp. 73–75, 2010.

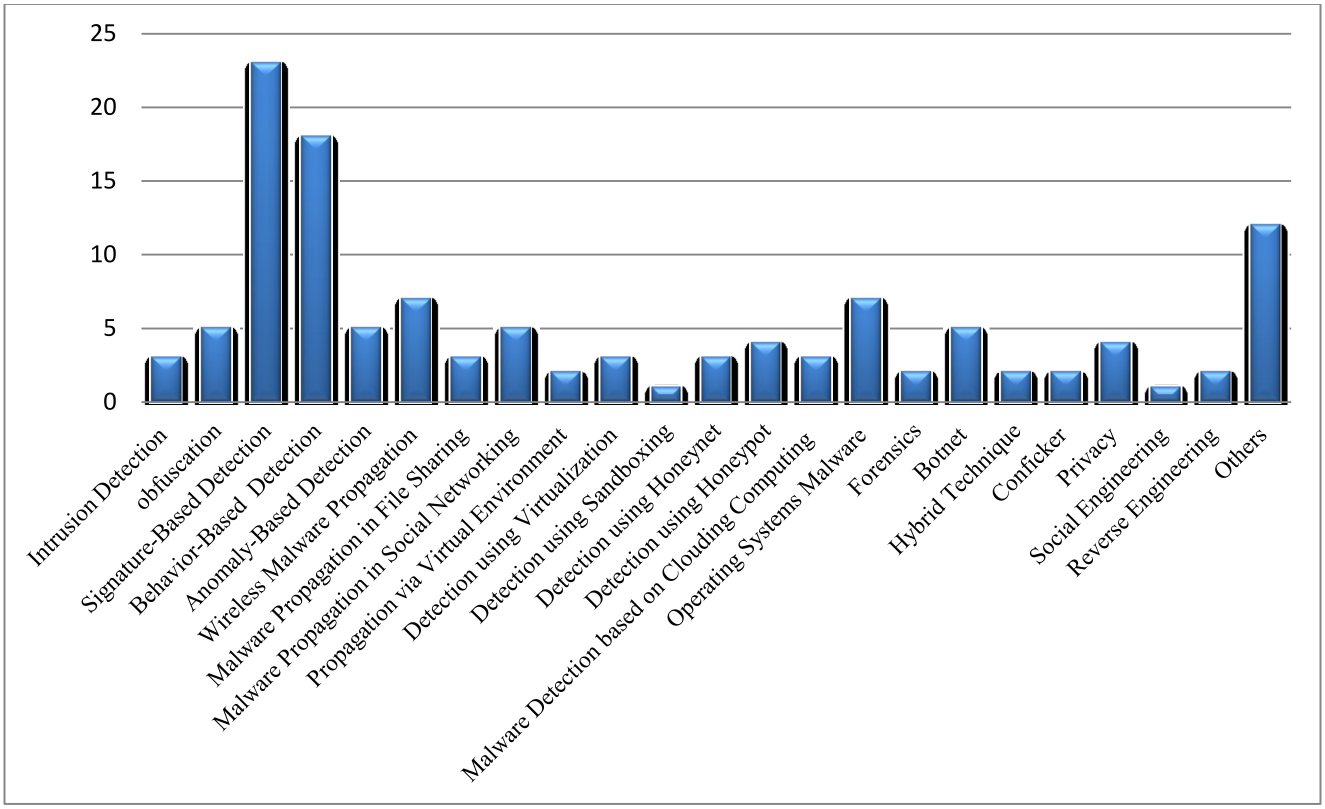
[39] U. Zurutuza, R. Uribeetxeberria, and D. Zamboni, “A data mining approach for analysis of worm activity through automatic signature generation,” in

Proceedings of the 1st ACM workshop on Workshop on AISec, 2008, pp. 61–70.

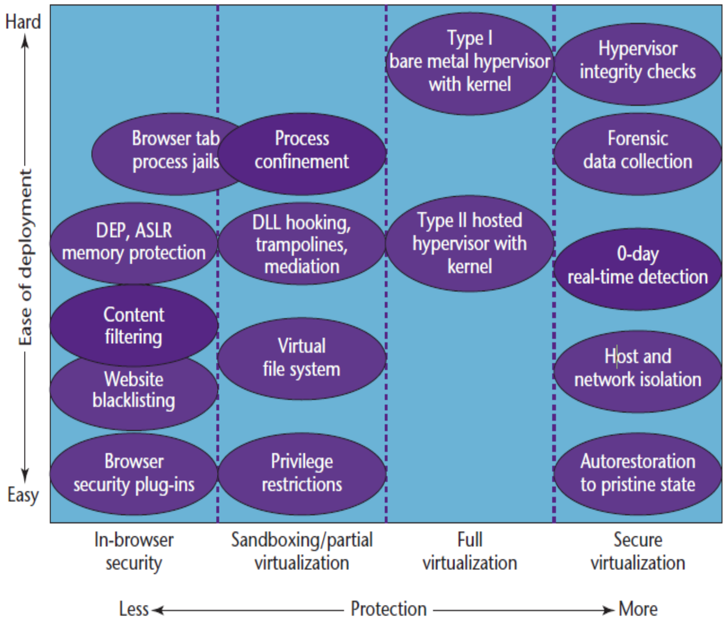
[40] K. Roundy and B. Miller, “Hybrid analysis and control of malware,” in *Recent Advances in Intrusion Detection*, 2010, pp. 317–338.

APPENDIXES

Appendix 01: Most important topics in the studied field



Appendix 02: Comparison between sandboxing and virtualization security in terms of protection level and ease of deployment


Hard
Ease of deployment
Easy
Browser tab process jails
Process confinement
Type I bare metal hypervisor with kernel
Hypervisor integrity checks
Forensic data collection
DEP, ASLR memory protection
DLL hooking, trampolines, mediation
Type II hosted hypervisor with kernel
0-day real-time detection
Content filtering
Website blacklisting
Virtual file system
Host and network isolation
Browser security plug-ins
Privilege restrictions
Autorestoration to pristine state
In-browser security
Sandboxing/partial virtualization
Full virtualization
Secure virtualization
Less
Protection
More