Modeling and Detection of Web Application Worm

Shreyasi P. Shuddhalwar[1], Mansi P. Shuddhalwar[2], Prachi Dakhole[3], Sonal Bharre[4]
Department Of Computer Technology
KDK College Of Engineering, Nagpur

***Abstract*--Self-duplicating, self-propagating malicious codes known as computer worms spread themselves without any human interaction and launch the most destructive attacks against computer network. Active worm's is also one type of worm which causes more security threats to the internet, due to ability of active worms to propagate in an automated fashion.. This article presents web application worms modeling and detection. This worm undergoes propagation from existing worm. So in order to model, identify and analyze these worms. We are designing modeling, propagating and detection techniques. Here we developed first model and analyses characteristics of worm through their behavior and classify worm detection algorithms based on parameters used in the algorithm, hence to remove he worm.**

Keywords*-*Modeling, Self-propagation,, Behaviour, Detection, Vulnerability, Algorithms.

I. INTRODUCTION

Web application has become a very popular application nowadays. Lots of people get connected to the internet regularly in order to fulfill their needs through all sorts of web application. Beside web application issues arising around internet user, web browser also plays important roles in causing malware attack. There are several types of malware that can infect a web application, such as worms and viruses. Different type of malware has different objective carried from The creator. For example, worms actually self-circulate from one host to another.

Infecting them and do not depend on the language used to develop the web application. Active worm's refers to malicious software program that propagates itself on the internet to infect other computers. Self-propagating malicious codes known as computer worms spread themselves without any human interaction and launch the most destructive attacks against computer networks. Being fully automated, a worm's behavior is usually repetitious and predictable, making it possible to be detected.A computer worm, after it is released, includes the following phases: target finding, worm transferring, worm activation, and worm infection. During the first two phases, the worm is active over the internet thus making it possible to be detected by detection systems

The other two phases are hard to detect by detection system .After detecting any worm we will have to make containment which is known as recovery

In order to know the behavior of worms, we have to understand how it attacks.

II. WEB WORM ATTACKS

A. Exploits some vulnerability in web application.

B. Sends specially crafted request which database.

- Execute code on target.
- Injects code into

C. All attacks travel over HTTP.

A computer worm, after it is released, includes the following phases:

- Target finding
- worm transferring
- Worm activation
- worm infection

The first step of a worm's life is to find targets. A worm may find its target or next victim using many different strategies. These strategies include: blind scan- the blind scan method includes sequential, random and permutation scanning, though they are probabilistic because of high failure connection rate and because the worm has no prior knowledge about the targets. The second target finding strategy is using a hit-list. The hit-list is a list of pre-scanned vulnerable addresses; by this the worm knows exactly where the targets are. The variation for this type of target finding strategy is that the larger the size of the hit-list, the more accurate and the more damage it can cause.

Thirdly, the use of network topology can enable a worm to find its target because many hosts on the internet store information about other hosts on the network revealing their vulnerabilities.

Fourthly, a passive strategy is another approach worms employ in finding targets by patiently waiting for victims to communicate with where the worm is resident.

Lastly, web searching is another strategy used by worms to find their targets because web searches avoid being detected by traditional detection techniques.

The second phase of a worm's life cycle is its transferring or propagation. It does this by employing three different schemes, namely: self-carried- this method allows the worm

code to be transferred in a packet by itself. Second channel this method allows the worm, after finding its target, to go into the target and download the worms code through a 'backdoor' that has been installed by some applications. Embedded- this method allows the worm to attach its code to legitimate traffic for example an e-mail in order to hide itself. This method is very deceitful and often unnoticed.

The third phase of a worm's life cycle is worm activation, that is, how the worm is transmitted over the network. There are two basic ways in which worms are transmitted over the network; they are transmission control protocol (TCP) a ndUser datagram protocol (UDP). The main difference is that TCP worms are connection oriented because they require a connection to be established before infection can begin, unlike UDP worms that are connectionless and requires no Connection to infect targets, this makes them spread very rapidly.

The last phase of a worm's life cycle is worm infection. This phase of the worm is associated with the actual worm code format. Worms usually send their code in a direct manner which causes detection systems to identify them quickly. Worms can be monomorphic in format; filling the code with irrelevant data but maintaining a single signature. They can also be polymorphic in format; that is, their code changes dynamically by scrambling them so that the worm takes different forms from different views though maintaining the same function. This type of worm format is very hard to be detected by signature-based detection. Another worm format is metamorphic worms. It changes not just appearance but also Behavior.

III. WEB APPLICATION WORMS

The Web Application Worm has the capability to intelligently manipulate its scan traffic volume over time, thereby camouflaging its propagation from existing worm detection systems. This Worm has a self-propagating behavior similar to traditional worms, i.e., it intends to rapidly infect as many vulnerable computers as possible.

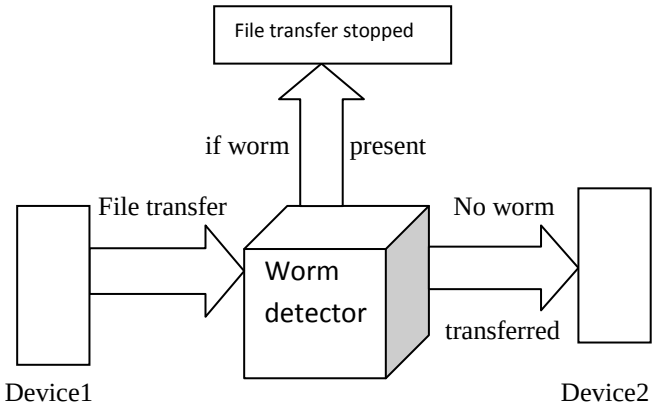

Fig1. Block diagram of worm detection system

Fig shows the block diagram of worm detection system. It consists of worm detector , devices .The device 1 will start sending files to device2 ,while transferred worm detector is in between them ,which will check the existence of worm , if present stop sending files else send it to other device 2. The worm detector is main system which will is based on various data mining algorithms of classification. This paper investigates new techniques to detect worms. Initially at first when worm will start affecting, it will find target first in order to attack or spread itself. Once target is find out, it will start propagation by using either any technique of propagation. So device 1 will transfer a file to worm detector module This module will detect the worm, if there is no worm, it will directly pass file to device 2, otherwise if worm is present, it will stop transfer of file.

IV. SYSTEM ARCHITECTURE

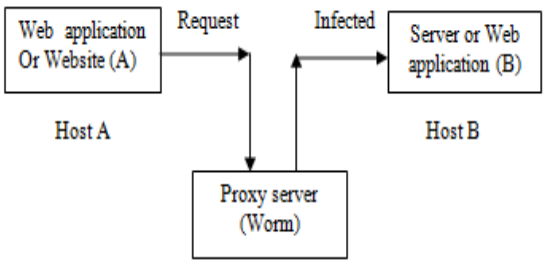

Fig 2. System architecture

V. MODULES FOR MODELING AND DETECTION OF WEB APPLICATION WORM

We are about to create the following modules:

A. TRANSACTION

In first module we will show the transaction which machines generally do. That is, host A send request to host B and we get the response from destination host i.e. host B. While this process happens there may be possibility of worm attack.

B. TRASACTION AFTER ATTACK

In second module after sending a request from host A to host B there is a server with code of worm written in it which infects the request send to Host B from Host A. And the request reach to Host B is infected by worm.

C. MECHANISM

In third module we are going to detect the worm by various detecting algorithm which infects the request

that sends to Host B. That is we have to remove attacks on Host B. By applying the various deleting algorithms.

VI. MOTIVATION, OBJECTIVES, AND LIMITATIONS:

A. Motivation

A lot of hacking activities have been taking place on the Internet and one method employed is the use of computer Worms. These attacks disrupt a lot of business activities Destroying their resources and causing huge financial Damages. Data recovery procedures are more expensive than

The cost of implementing a network. Our motive is make Sure that the network is secured using the appropriate measure and attack trace back should be effective to bring these attackers to book.

B. Objective

This analysis on web application worm will be done to fulfill several objectives.
Stated below are all the objectives:

- To analyze vulnerabilities in various web applications
- To find out the probability of worm attack on web application.
- To measure processing time of attack via malicious code as a worm.
- To verify the method of worm attack on web application.
- To come out with a guideline and detection techniques to develop secure web application.

C. Limitations

Attack trace back is largely dependent on detecting these attacks. A major challenge in attack detection is the use of Steganography, which is the principle of hiding information behind objects, image, sound and even ordinary text files.
The second limitation is the technique of IP spoofing. This technique allows an attacker to impersonate an IP address and pretends to be the sender of packets even while he is somewhere else. This makes attack trace back fruitless because the wrong host will be traced.

VII. IMPORTANCE OF STUDY

In this day and age, almost every single application can be transform into a web application. Building a web application is not a real problem since there are currently many software developers coming into the scene. The real crisis happen when the software engineers themselves did not aware of the vulnerabilities of the applications they have written, let alone the defect on the web browser. In some situation, some of the defect in browser can be overcome during the application development. But the main problem was arising due to the availability of various difficulties in handling such problem. This kind of defect will lead to various kind of attack from public once the vulnerabilities get to be known. In addition, there are lots of tools available in the internet that can be use to scan the vulnerabilities on any web application. So, it is indeed a software engineer's responsibility to make sure that their applications are not vulnerable to be attacked from inside and outside of the network.

VIII. LITERATURE SURVEY

A lot of businesses and organizations depend on computer networks for efficient and effective operations. One of the biggest threats to such businesses and organizations is computer worms. Computer worms are malicious pieces of Code that propagate themselves via network connections, exploiting the security lapses in computers on the network. They propagate without human intervention.

In 1988, Robert Tapper Morris launched the first computer worm at Cornell University. The worm was later called the "Morris Worm." It caused expensive and wide spread damage on all kinds of computer. It takes advantage of buffer overflow vulnerability. Its original intention was to discover the number of hosts on the internet, but flaws in the program caused the code to copy itself multiple times to already infected hosts, slowing them down until they became unusable.

Another known computer worm is the Code Red I; it was first discovered running on Microsoft's internet information server (IIS) web service in 2001. It uses a blind scan which scans port 80 on random IP addresses in order to find vulnerable hosts and then launches a denial-of-service (DoS) attack. Later Code Red II was discovered which, unlike Code Red I, installs a 'backdoor' on infected systems. In 1987, Dorothy Denning proposed the concept of intrusion detection as a solution to the problem of providing a sense of security in computer systems. The concept involves abnormal usage of the system. Although many other techniques were later used.

An ideal intrusion detection system should function in the Following manner:

- It should be able to detect known and unknown worms
- Minimal processing overhead during detection
- Very low level of administrator involvement
- Correctly detect worms that change signature
- Design should not be complex
- Require minimum memory

In recent times, the modern scheme being employed in detecting worm attacks is the use of intrusion detection systems. They are a piece of hardware or software placed usually behind firewalls to detect illegitimate traffic, they can be network-based called NIDS or host-based called HIDS.

There are many proposed algorithms for these detection methods but the two most common schemes used are signature-based detection and anomaly-based detection The earliest reactive attack trace back approach is link testing. As the name implies, it traces packets from link to link, that is, hop-by-hop in order to determine the source of attack traffic. Starting from the router closest to the victim

In 2000, Burch and Cheswich developed a link-testing trace back technique that does not require much of ISP cooperation This technique was called controlled flooding, this is because it tests links by flooding them with large bursts of traffic from the attacker. Techniques to determine the path that the packets traversed. Its major challenge was that the logging of packets consumes router memory which in turn could affect the speed of transmission

IX. CONCLUSION

It can be concluded that computer worms, as described in this research, poses some characteristics that enables them to evade traditional network security methods. We have identified the characteristics of existing and hypothetical worms during the target finding and propagation phases of a worm's life cycle. Computer worms can be stealth in attack, propagate quickly, change their form and infect computers on networks, causing great financial and operational losses to businesses and organizations

We have discussed here that there can be a various techniques to detect web application worm. This survey also helps to understand how we going to accomplish our aim to model and detect the web application worm.

REFERENCES

[1] http://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Hoffman/BH-Fed-06-Hoffman-up.pdf

[2] http://www.ijarcsse.com/docs/papers/12_December2012/Volume_2_issue_12_December2012/V2I12-0154.pdf

[3] https://www.scribd.com/doc/90323062/PPT-for-Project-Review-1

[4] www.ijetae.com/.../IJETAE_1012_96.pdf

[5] www.ijarcsse.com/../V2112-0154.pdf