ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Detection of traditional and new types of Malware using Host-based detection scheme

Satish N.Chalurkar[1], Dr.B.B.Meshram[2]

[1]Computer Department,VJTI,Mumbai
[1]satishchalurkar1@gmail.com

[2]Head of Computer Department,VJTI,Mumbai
[2]bbmeshram@vjti.org.in

***Abstract*--- In this paper, we have discussed many traditional and new types of worms including c-worms also-worms stands for camouflaging worms because of its nature of self propagating and hiding nature. An active worm refers to a malicious software program that propagates itself on the Internet to infect other computers. The propagation of the worm is based on exploiting vulnerabilities of computers on the Internet. This paper also shows the working of various worms and their detection system. These detection techniques not only detect worms but also detect various malware attacks.**

***Keywords*--- active worms, c-worms, host based detection system, network based detection system, types of worms**

I. INTRODUCTION

Malware can infect systems by being bundled with other programs or attached as macros to files. Others are installed by exploiting a known vulnerability in an operating system (OS), network device, or other software, such as a hole in a browser that only requires users to visit a website to infect their computers. The vast majority, however, are installed by some action from a user, such as clicking an e-mail attachment or downloading a file from the Internet.

Some of the more commonly known types of malware are viruses, worms, Trojans, bots, back doors, spyware, and adware. Damage from malware varies from causing minor irritation (such as browser popup ads), to stealing confidential information or money, destroying data, and compromising and/or entirely disabling systems and networks.

Malware cannot damage the physical hardware of systems and network equipment, but it can damage the data and software residing on the equipment. Malware should also not be confused with defective software, which is intended for legitimate purposes but has errors or bugs.

What makes them so deadly and unidentifiable is the way they reach your PC and go unnoticed under the camouflage of reliable software. Most of the advertisers banking on adware use third party software bundles to pack their adware in. Since people want to install these software, they also end up installing the adware.

Malware also use rootkits to manipulate the operating system. What they do is make changes such that they are not identified on the Task Manager Panel. So in essence your PC might run evidently slow, you still can't see the applications running in the back ground and thus give malware ample time to spread to all the roots.

Malware can be roughly broken down into types according to the malware's method of operation. Anti-"virus" software, despite its name, is able to detect all of these types of malware.

An active worm refers to a malicious software program that propagates itself on the Internet to infect other computers. The propagation of the worm is based on exploiting vulnerabilities of computers on the Internet.

"A worm is a program that can run by itself and can propagate a fully working version of itself to other machines. It is derived from the word tapeworm, a parasitic organism that lives inside a host and saps its resources to maintain itself."

The worm used an interesting hook-and-haul method of propagation that masked its entrance to a site and kept its mechanisms secret. It was also multifaceted and multi-architecture, using multiple methods to gain entrance to a machine and affecting two entirely different computer architectures. It had an intensely computational part

that was meant to give it resilience through the ability to infiltrate through many user accounts, but it had an ineffective mechanism to limit its growth rate.

Particularly, the epidemic dynamic model assumes that any given computer is in one of the following states:

- immune, vulnerable, or infected. An immune computer is one that cannot be infected by a worm;
- a vulnerable computer is one that has the potential of being infected by a worm;
- an infected computer is one that has been infected by a worm.

In this paper,Section II described various types of worms and the details of active worms.Active worms are those that infect system.Section III described c-worms and various detection system.Section IV contain conclusion.

II. RELATED WORK

A. *Types of worms*

Many real-world worms have caused notable damage on the Internet. These worms include "Code-Red" worm, "Slammer" worm, and "Witty"/ "Sasser" worms. Many active worms are used to infect a large number of computers and recruit them as bots or zombies, which are networked together to form botnets.

1) Slapper Worms

Slapper would attempt to remotely compromise systems by randomly selecting a network to scan and doing a sequential sweep of all IP addresses in the network while looking for vulnerable Web servers.

The Slapper worm's P2P communications protocol was designed to be used by a hypothetical client to send commands to and receive responses from an infected host (a node). In this way, the client can perform several different actions while hiding its network location and making communications more difficult to monitor[1].

2) Slammer Worms

Slammer (sometimes called Sapphire) was the fastest computer worm in history. As it began spreading throughout the Internet, the worm infected more than 90 percent of vulnerable hosts within 10 minutes, causing significant disruption to financial, transportation, and government institutions and precluding any human-based response[2].

Slammer's most novel feature is its propagation speed. In approximately three minutes, the worm achieved its full scanning rate (more than 55 million scans per second), after which the growth rate slowed because significant portions of the network had insufficient bandwidth to accommodate more growth.

The worm's spreading strategy uses random scanning-it randomly selects IP addresses, eventually finding and infecting all susceptible hosts. Random-scanning worms initially spread exponentially, but their rapid new-host infection slows as the worms continually retry infected or immune addresses. Thus, as with the Code Red worm, Slammer's infected-host proportion follows a classic logistic form of initial exponential growth in a finite system.it label this growth behaviour a *random constant spread* (RCS) model.

While Slammer spread nearly two orders of magnitude faster than Code Red, it probably infected fewer machines. Both worms use the same basic scanning strategy to find vulnerable machines and transfer their exploitive payloads; however, they differ in their scanning constraints. While Code Red is latency-limited, Slammer is bandwidth-limited, enabling Slammer to scan as fast as a compromised computer can transmit packets or a network can deliver them.

3) Sobig Worms

Like its predecessors, Sobig.E is unremarkable in many ways. It's a piece of malicious code that targets Microsoft Windows operating systems. Written in Microsoft Visual C++, it makes use of threads, its executable is compressed with either UPX or TeLock, it collects email addresses by harvesting files (such as Windows Address Book [WAB], Outlook Express mailbox [DBX], HTM, HTML, Mail message [EML], or text [TXT]), and attempts to infect new systems by sending them an infected email message or by copying itself to an open network share. The worm also includes its own simple mail transport protocol (SMTP) engine, spoofs its emails' source address, encrypts and decrypts text strings as needed, and creates a mutual exclusion object (mutex) on infected systems to ensure they are not infected more than once.

4) Morris Worms

- It attacked one operating system, but two different computer architectures.
- It had three distinct propagation vectors.
- It had several mechanisms for finding both potential nodes to infect, particularly information about the local system's IP connectivity (its network class and gateway), and information found in user accounts.
- It traversed trusted accounts using password guessing.
- The worm made heavy use of this computationally intensive method by employing four information sources: accounts with null passwords (no password), information related to the user account, an internal dictionary, and a word list on the local machines, /usr/dict/words.
- It installed its software via a two-step "hook and haul" method (explained later in the "Inside the worm" subsection) that required the use of a C compiler, link loader, and a callback network connection to the infecting system.
- It evaded notice by obscuring the process parameters and rarely leaving files behind.
- It attempted to limit the reinfection rate on each node (but not the total number).
- It attempted to run forever on as many nodes as possible.

Although there had been worms before, no one had tried to run one on a complex topology. For this worm to achieve its purpose of widespread propagation, it had to discover local topology in an arbitrary graph[7].

5) Witty Worms

While the Witty worm is only the latest in a string of self-propagating remote exploits, it distinguishes itself through several interesting features:

• It was the first widely propagated Internet worm to carry a destructive payload.

• It started in an organized manner with an order of magnitude more ground-zero hosts than any previous worm.

• It represents the shortest known interval between vulnerability disclosure and worm release—it began spreading the day after the ISS vulnerability was publicized.

• It spread through a host population in which every compromised host was proactive in securing its computer and networks.

• It spread through a population almost an order of magnitude smaller than that of previous worms, demonstrating worms' viability as an automated mechanism to rapidly compromise machines on the Internet, even in niches without a software monopoly.

B. Active Worms

Active worms are similar to biological viruses in terms of their infectous and self-propagating nature. They identify vulnerable computers, infect them and the worm-infected computers propagate the infection further to other vulnerable computers. In order to understand worm behavior, we first need to model it. Active worms use various scan mechanisms to propagate themselves efficiently.

The basic form of active worms can be categorized as having the Pure Random Scan (PRS) nature. In the PRS form, a worm-infected computer continuously scans a set of random Internet IP addresses to find new vulnerable computers. Other worms propagate themselves more effectively than PRS worms using various methods, e.g., network port scanning, email, file sharing, Peer-to-Peer (P2P) networks, and Instant Messaging (IM). In addition, worms usedifferent scan strategies during different stages of propagation.In order to increase propagation efficiency, they use a local network or hitlist to infect previously identified vulnerable computers at the initial stage of propagation.

They may also use DNS, network topology and routing information to identify active computers instead of randomly scanning IP addresses.They split the target IP address space during propagation in order to avoid duplicate scans. Li *et al.* studied a divide-conquer scanning technique that could potentially spread faster and stealthier than a traditional random-scanning worm. Ha *et al.* Formulated the problem of finding a fast and resilient propagation topology and propagation schedule for Flash worms. Yang *et al.* studied the worm propagation over the sensor networks.

III. PROPOSED TOOL FOR MALWARE DETECTION

A. C-Worm

The C-Worm camouflages its propagation by controlling scan traffic volume during its propagation. The simplest way to manipulate scan traffic volume is to randomly change the number of worm instances conducting port-scans.

As other alternatives, a worm attacker may use an open-loop control (non-feedback) mechanism by

choosing a randomized and time related pattern for the scanning and infection in order to avoid being detected. Nevertheless, the open-loop control approach raises some issues of the invisibility of the attack.

First, as we know, worm propagation over the Internet can be considered a dynamic system. When an attacker launches worm propagation, it is very challenging for the attacker to know the accurate parameters for worm propagation dynamics over the Internet.

Consequently, the overall worm scan traffic volume in the open-loop control system will expose a much higher probability to show an increasing trend with the progress of worm propagation. As more and more computers get infected, they, in turn, take part in scanning other computers. Hence, we consider the Cworm as a *worst case attacking scenario* that uses a closedloop control for regulating the propagation speed based on the feedback propagation statuseasy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it[10].

B. Existing Detection System

1) Host Based Detection System

Worm detection has been intensively studied in the past and can be generally classified into two categories: "host-based" detection and "network-based" detection.

Host-based detection systems detect worms by monitoring, collecting, and analyzing worm behaviors on end hosts.Since worms are malicious programs that execute on these computers, analyzing the behavior of worm executables plays an important role in host based detection systems.

2) Network Based Detection System

In contrast, network-based detection systems detect worms primarily by monitoring, collecting, and analyzing the scan traffic (messages to identify vulnerable computers) generated by worm attacks. Network-based detection schemes commonly analyze the collected scanning traffic data by applying certain decision rules for detecting the worm propagation. For example,Venkataraman *et al*. and Wu *et al*. in proposed schemes to examine statistics of scan traffic volume, Zou *et al*. presented a trend-based detection scheme to examine the exponential increase pattern of scan traffic , Lakhina *et al* proposed schemes to examine other features of scan traffic, such as the distribution of destination addresses. Other works study worms that attempt to take on new patterns to avoid detection[10] .

Here are some of the techniques for identifying the worms in the host or in the networks.

1) Distributing sensors

Determining the number and strategic placement of distributed sensors (for example, at an enclave's gateway, at an upstream peering point, or both) for a particular-size enclavemaximizes coverage and minimizes communication cost and time to detect propagations and attack precursors.

2) Inferring intent

The relationships among common targeted victims suggest what a scanning source's intent might be. For example, a common source of stealthy scanning from an attacking IP address directed toward a set of unrelated victims appears fundamentally different than an attacker scanning a set of IP addresses all owned by, say, several different banks.

3) Profiling behavior.

A longitudinal study of attacker behavior and intent and their attacks against victims provide sufficient repeated behavior to accurately predict future attack steps.

4) *Classifying activities*

We need a way to quickly classify worms and scan or probe activity into useful clusters and profiles according to their characteristics (destination ports, interprobe delay, and payload length, for example) and behaviour.

C. Proposed Detection Scheme

There are three existing system to detect the worms as well as malware attack.

The first scheme is the volume mean-based (MEAN) detection scheme which uses mean of scan traffic to detect worm propagation; the second scheme is the trend-based (TREND) detection scheme which uses the increasing trend of scan traffic to detect worm propagation; and the third scheme is the victim number variance based (VAR) detection scheme which uses the variance of the scan traffic to detect worm propagation.

The C-Worm adapts their propagation traffic patterns in order to reduce the probability of detection, and to eventually infect more computers. The C-Worm is different from polymorphic worms that deliberately change their *payload signatures* during propagation.

1) Architecture of Detection Tool:

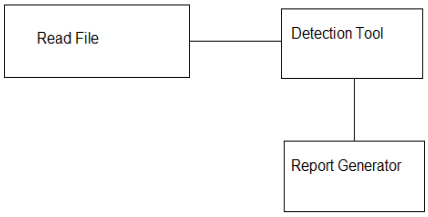
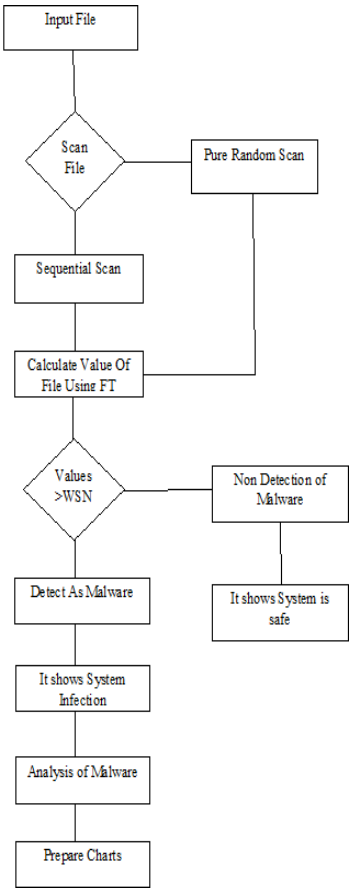


Fig 1:Architecture of detection Tool

The above diagram shows the architecture of detection tool.The First block take the file to read then it passed to detection tool block which contain the various block that shown in below diagram then it finally generate the report.
The flow of above architecture are given below,which are also proposed in the paper.



Recall that the C-Worm goes undetected by detection schemes that try to determine the worm propagation only in the time domain. Instead of time domain,frequency domain is used to detect the c worms as well as traditional worms.
The user give file to the scan block.It has to option,either it scan sequentially or it can randomly.For each file,it calculate value using the Fourier Transform.At every time,the file is scan,that value is compared with Window sliding number.if value of that file that is to be scanned are greater than WSN then it detect as malware otherwise it shows non-detection of malware.If the malware is found then analysis the malware and prepare the chart for that analysis.
Notice that the frequency domain analysis will require more samples in comparison with the time domain analysis,since the frequency domain analysis technique such as the Fourier transform, needs to derive power spectrum amplitude for different frequencies.

When we scan system,first it will take the single file and then generates its number using fourier transform.if that number is larger than the number which are gererated from the window sliding number then it will shows that worms or malware is detected.

It will not only scan sequential but also randomly because of its nature of self propogation in different location in the system.

IV. CONCLUSION

In this paper, we studied a new class of smart-worm called CWorm,which has the capability to camouflage its propagation and further avoid the detection. It showed that, although the C-Worm successfully camouflages its propagation in the time domain, its camouflaging nature inevitably manifests as a distinct pattern in the frequency domain. Based on observation, we creates Host-based detection scheme to detect the C-Worm.

REFERENCES

[1] IVAN ARCE,ELIAS LEVY ," An Analysis of the Slapper Worm".

[2] DAVID MOORE, VERN PAXSON," Inside the Slammer Worm"

[3] ELIAS LEVY," The Making of a Spam Zombie Army"

[4] Yong Tang, Bin Xiao, Member, IEEE, and Xicheng Lu" Signature Tree Generation for Polymorphic Worms", IEEE TRANSACTIONS ON COMPUTERS, VOL. 60, NO. 4, APRIL 2011

[5] Wei Yu, Member, IEEE, Nan Zhang, Member, IEEE, Xinwen Fu, Member, IEEE, and Wei Zhao, Fellow, IEEESelf-Disciplinary Worms and Countermeasures: Modeling and Analysis", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 21, NO. 10, OCTOBER 2010

[6] Yong Tang and Shigang Chen," An Automated Signature-Based Approach against Polymorphic Internet Worms" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 18, NO. 7, JULY 2007

[7] HILARIE ORMAN," The Morris Worm: A Fifteen-Year Perspective"

[8] Guanhua Yan and Stephan Eidenbenz," Modeling Propagation Dynamics of Bluetooth Worms (Extended Version)", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 8, NO. 3, MARCH 2009

[9] SALVATORE J. STOLFO," Worm and Attack Early Warning"

[10] Wei Yu, Xun Wang, Prasad Calyam, Dong Xuan, and Wei Zhao," Modeling and Detection of Camouflaging Worm", IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING ,VOL. 8, NO. 3, MAY-JUNE 2011

[11] Zhenhai Duan, Peng Chen, Fernando Sanchez ,Yingfei Dong ,Mary Stephenson, James Barker" Detecting Spam Zombies by Monitoring Outgoing Messages"

[12]Yanfang Ye, Tao Li, Qingshan Jiang, and Youyu Wang" CIMDS: *Adapting Postprocessing Techniques of Associative Classification for Malware Detection"*

[13] Carlos Raniery P. dos Santos, Rafael Santos Bezerra, Joao Marcelo Ceron," *Botnet Master Detection Using a Mashup-based Approach"*

[14] Zongqu Zhao" *A Virus Detection Scheme Based on Features of Control Flow Graph"*

[15] Mohamad Fadli Zolkipli Aman Jantan *"An Approach for Malware Behavior Identification and Classification"*

[16] M. Shankarapani, K. Kancherla, S. Ramammoorthy, R. Movva, and S. Mukkamala *"Kernel Machines for Malware Classification and Similarity Analysis"*

[17] Felix Leder, Bastian Steinbock, Peter Martini*"Classification and Detection of Metamorphic Malware using Value Set Analysis"*

[18] Desmond Lobo, Paul Watters and Xinwen Wu "RBACS: *Rootkit Behavioral Analysis and Classification System"*

[19] Siddiqui M.A.:" *Data Mining Methods for Malware Detection."*

[20] Moskovitch R, Feher, Tzachar N, Berger E, Gitelman M, Dolev S,et al." *Unknown malcode detection using OPCODE representation. "*

[21]Michael Erbschloe *"A Computer Security Professional's Guide to Malicious.*