Advances in Computer Science and its Applications (ISSN 2166-2924)
Vol. 1, No. 1, March 2012

www.worldsciencepublisher.org

Novel Approach for the propagation and detection of active worm in a Virtual System

A.S.N. Chakravarthy[*], K. Manasa Veena[**], K. Renuka[**]

*(Professor, Department of Electronics and Computer Engineering, K L University, Guntur
**(Student, Department of Electronics and Computer Engineering, K L University, Guntur

Email: asnchakravarthy@yahoo.com, manasaveena_555@yahoo.com, renuka.kolla@gmail.com

ABSTRACT: Active worms are one of the major security threats to the Internet. This is because of their ability to propagate in an automated fashion as they continuously compromise computers on the Internet. Camouflaging Worm (C-Worm in short) is one of the active worms. The C-Worm is different from traditional worms because they can camouflage (hide) itself from the detection schemes by manipulating their scan traffic volume. In this paper we have implemented an approach to propagate and also to detect and delete different types of worms in the network and we implemented it in a virtual system. For this approach we have used socket programming. The file streams for propagation and detection and java swings for better GUI purpose. Our paper thus propagates a selected worm from the list of worms given to the user, detects and deletes them in the virtual system.

Keywords– Worm; C-Worm; Virtual System; Socket; File Streams; Warhol Worm

1. INTRODUCTION

Virus and worms are menace to many systems. The main difference between a virus and a worm is a virus need human interventions to start replication where as worms need none. A worm is similar to a virus by design and is considered to be a sub-class of a virus. Worms are considered as internet virus as they use internet as medium for their propagation. So the worm takes advantage of file or information transport features on your system, which is what allows it to travel unaided. The biggest danger with it is its capability to replicate itself on your system, so rather than your computer sending out a single worm, it could send out hundreds or thousands of copies of itself, creating a huge devastating effect. One example would be for a worm to send a copy of itself to everyone listed in your e-mail address book. Due to the copying nature of a worm and its capability to travel across networks the end result in most cases is that the worm.

Active worms are the special type of worms which change their properties to go undetected from the detection schemes while scanning process. Examples for active worms are Attack worm, self-stopping worm, Stealth worm, C--worm. These worms have capabilities to change their nature of propagation while getting scanned. For example “self-stopping” active worm propagates [1] for a random period of time and automatically stops itself from propagation so as to confuse the detection schemes during the scanning procedure. Worm might also use the evasive scan and traffic morphing technique to hide the detection .This worm attempts to remain hidden by sleeping (suspending scans) when it suspects it is under detection. Thus active worms adopt smart attack strategies could exhibit overall scan traffic patterns different from those of traditional worms.

In this paper, we have designed a paper whose objective is to propagate a worm and then detect and delete it in the virtual system.

The rest of the paper is organized as follows. Section 2 describes about recent studies on active worms and various approaches for the modelling and detection of a worm.Section3 depicts the approach we have followed to meet our objective in the virtual system. Section4 gives conclusion to the paper. Future Scope is pointed out in the Section 5.

2. RELATED WORK

Recent studies on active worms shows that they can be propagated through an approach called random Ip scan[2].An example for such worm is Deloder worm, that tries to drop a backdoor component. This worm is spreading through vulnerable machines by scanning random IP addresses, trying to connect on Port 445. Port 445 (Microsoft SMB over TCP/IP) allows outsiders to access Windows file shares. If a successful connection is made, Deloder drops a called INST.EXE in the Windows Start folder. This is a Trojan designed to open a backdoor access to compromised computer. It then copies a file called DVLDR32.EXE, a copy of the worm itself, onto infected machines. Then it tries to obtain a list of computers connected to the same network and attempts to access them using default passwords. Finally, Deloder disables shared network resources and places entries in the Windows Registry of compromised machines to make sure it is always run. This action has the side-effect of disabling network sharing [3]. Another approaches can be differentiated based on how the worm chooses the host. For example Warhol worm. The comparison of an active worm Code Red and Warhol is as follows. The recent outbreak of the Code Red active worm could have been much worse. But although it was fast, the 12 hours it took to reach epidemic levels still allows for an organized response. But by simply changing the infection pattern, it is possible for a malicious programmer to build a "Warhol Worm", able to attack all vulnerable machines, worldwide, in 15 minutes. A reactive, human defense

would fail before such an onslaught. It is an important exercise to realize just how vulnerable we are. This active worm such as Code Red [4] or the original Morris worm takes advantage of a security hole in a server. It scans through the Internet, looking for machines running that service. Then it tries to break into that service. If successful, it infects the target machine with another copy of itself. The difference between an active worm and a Warhol worm is minor, just a different strategy of choosing hosts to infect [5]. C-worms are the smart worm that camouflages (hides) themselves from the detection schemes.

The camouflage is achieved by manipulating the scan traffic volume of worm-infected computers. Other novel approaches for the detection of active worm (especially C-worm) include Design rationale and Spectral Based detection scheme[1].The Design rationale scheme involves the identification of the C-Worm propagation in the frequency domain, by the use of the distribution of Power Spectral density(PSD) and its corresponding Spectral Flatness Measure(SFM) of the scan traffic. Particularly, PSD describes how the power of a time series is distributed in the frequency domain. Where in Spectral Based detection, it uses a "destination count" as the number of the unique destination IP addresses targeted by launched scans during worm propagation. An ITM system collects logs from distributed monitors across the Internet. In general, an ITM system consists of one centralized data center and a number of monitors distributed across the Internet. Each monitor records traffic that addressed to a range of IP addresses (which are not commonly used IP address also called the dark IP addresses) and periodically sends the traffic logs to the data center. The data center then analyzes the collected traffic LOGS and publishes reports (e.g., statistics of monitored traffic) to ITM system users. In this spectrum-based detection scheme, the distribution of PSD and its corresponding *SFM* are used to distinguish the CWorm scan traffic from the non-worm scan traffic.

3. PAPER WORK

Our paper illustrated in this paper has the objective to propagate the worm in a virtual system and then detect and delete those considering different drivers as different hosts.

As changing the extensions of the files leads to the crash of the system we have implemented our paper on virtual system with a normal text file as the worm.

3.1 GUI module

For a better effective Graphical User Interface, we have used java swings to design the interface module. This screen enables the user to select the type of the worm for propagation in the system, enables the user to switch on the containment of scanning. This screen has three buttons that gives the user an option to start/stop spreading of worm, start the scanning procedure-which in turn start another scanning screen.

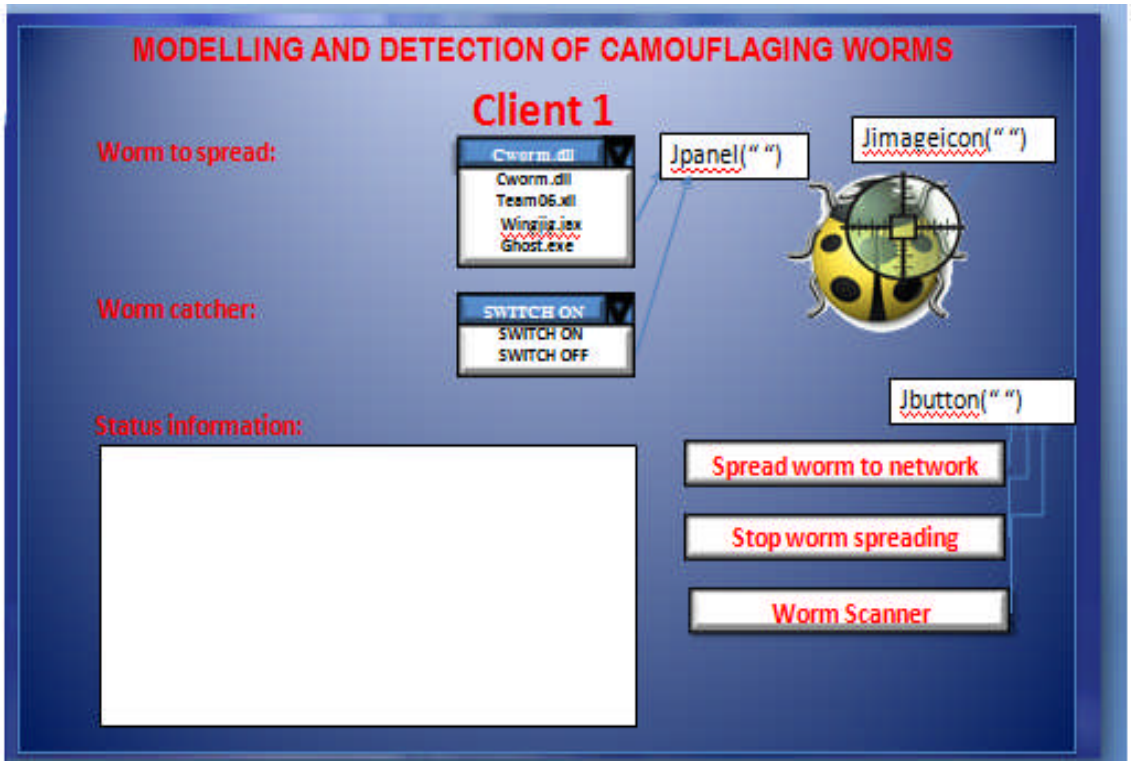

Fig. 1 GUI Screen for Propagation

This screen also has the status information text area which is for the user convenience. The GUI screen 1 also has an image.

When the user presses the "Worm Scanner" button, the application then displays another GUI screen called "CONTAINMENT" solely for detection and deletion of the worm. This containment screen enables the user to a. select the worm to be scanned b. to start the scan. This screen has three text areas a. The scan path to show the user the type of worm which is being scanned. b. Worm

detection area which shows the list of worms and type of worms under detection. c. The third area is scanned file having the number of files, number of folders and number of subfolders that are just scanned. Thus this module deals with the interface section using Java swings which is a non-web technology of Java.

Java swings enables to uses different classes to enhance the interface module by providing classes like JButton, JPanel, JTextArea, JImage. Invoking start() on the Swing Worker causes a new Thread.

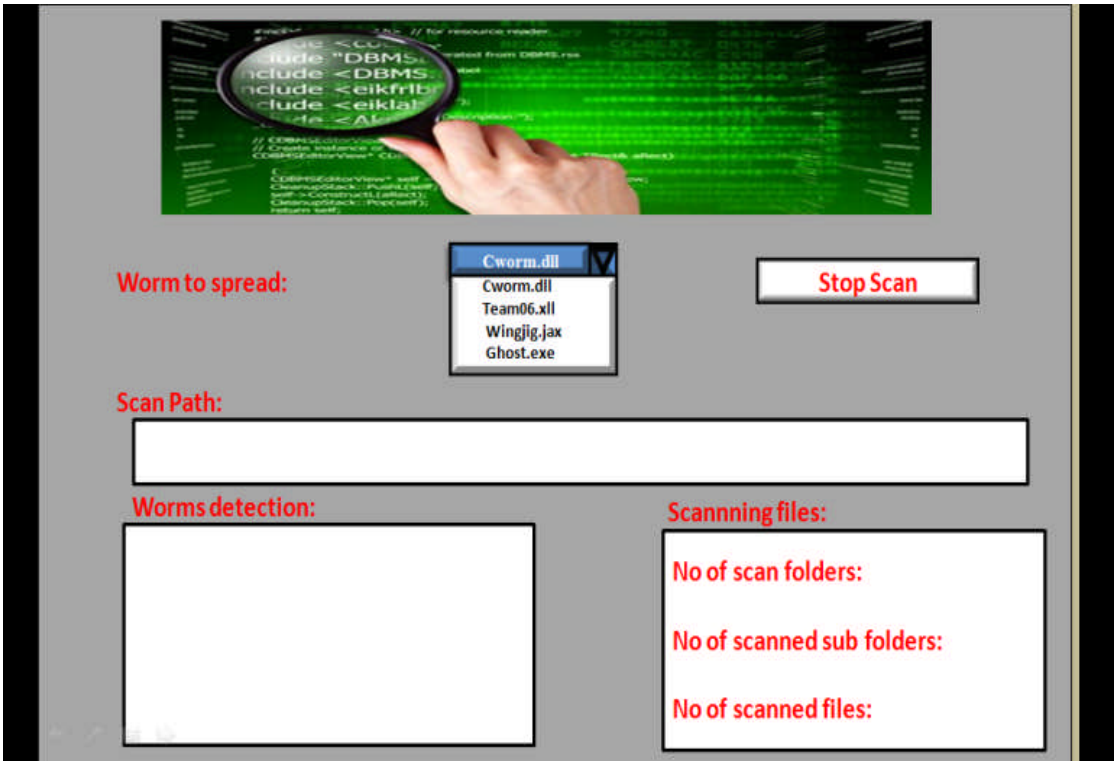

Fig. 2 Containment Screen

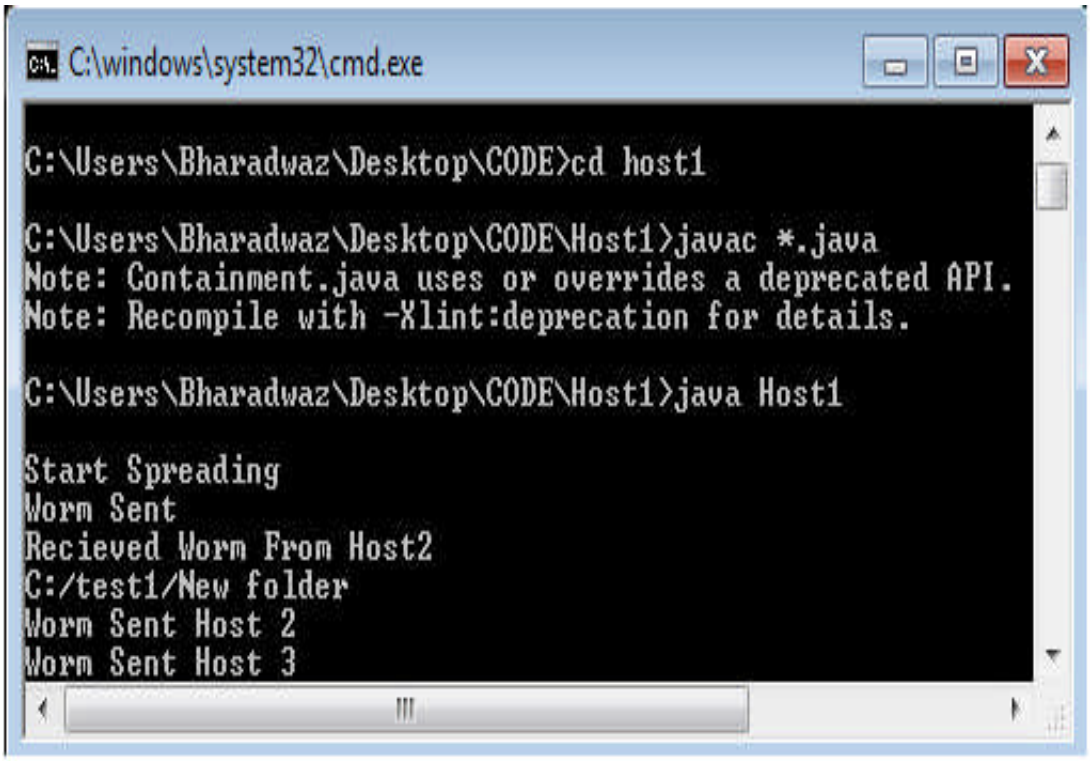

Fig. 3 Worm Spreading

This containment screen enables the user to a. select the worm to be scanned b. to start the scan. This screen has three text areas a. The scan path to show the user the type of worm which is being scanned. b. Worm detection area which shows the list of worms and type of worms under detection. c. The third area is scanned file having

the number of files, number of folders and number of subfolders that are just scanned. Thus this module deals with the interface section using Java swings which is a non-web technology of Java.

Java swings enables to uses different classes to enhance the interface module by providing classes like JButton, JPanel, JTextArea, JImage. Invoking start() on the Swing Worker causes a new Thread

3.2 Propagation of worm

This section deals with the propagation the selected worm in the virtual system. After the selection of type of worm. User presses the JButton "Start spreading" .when this Button is invoked the worm gets spreaded into different drivers. Socket programming is used for the propagation of the worm in the virtual system.

Thus the command prompt thus shows the path from where the worm has been sent and the name of the host which received those worms.

Now to stop the propagation user need to click JButton named "stop spreading" in all hosts, since once the propagation starts the worms produce child worms that are capable of being replicated in the child hosts. So when the user clicks the JButton "Stop spreading in all the hosts the screen appears as in Fig.3

Socket programming is initiated by creating as many numbers of sockets as the number of drivers present in the system. When implemented in the network, this socket is created using the IP address and port number of different hosts. But here as it is implemented in the virtual network, the IP address remains as "localHost" for the hosts.The port number differs and can be given within the range.File streams are used to copy these worm files to other hosts.

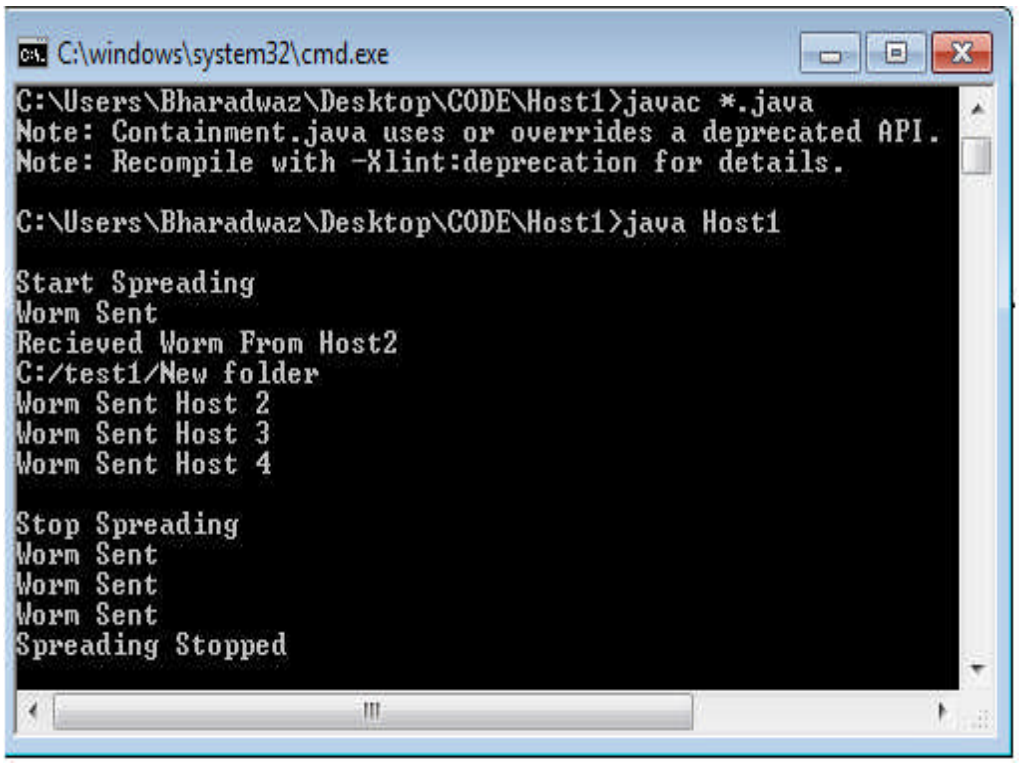

Fig. 4 Worm Propagation Stopped

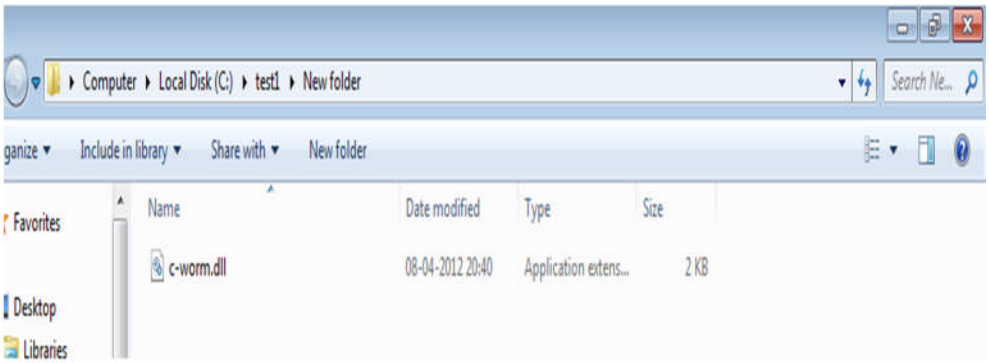

Fig. 5 Worm Detection

3.3Worm detection and deletion

The mechanism involved in the detection of worm is as follows: When the user is done with the propagation of the worm, it is his duty to start the scanning procedure by clicking the JButton "Worm scanner". This action invokes another screen to appear named "Containment".

The containment screen again asks the user to select the type of worm to be scanned. After the propagation of worm to the defined paths the folder has the worm

Then after scanning by scanning the folders, files and subfolders in every given path, it detects the worm by extension match and then by the file name match. Whenever the scanning starts, the path defined by the programmer is file streamed and then detection starts.

The file gets deleted after scanning and also as soon as the scanner window gets closed. In addition to the deletion, the containment also specifies the path scanned the number of detections, the files, folders and subfolders scanned in the process. Thus our paper detects the propagated worm by using file streams in java.

In real time applications it uses the random IP scan method to identify a host system nearby and then scans the system. It also considers the shortest path method to find the system to scan.

Fig. 6 Propagation and Deletion Screen

4. Conclusion

An active worm hides from the detection schemes by undergoing many internal changes and makes the detection even more difficult. Camouflaging worm also called c-worm especially changes the extensions of files and thus it's very difficult for us to implement our paper in real time system. So our paper suggests and implements a successful approach for the propagation and deletion of worm in a virtual system.

5. Future Scope

Our paper is implemented in virtual system as well as in LAN networks and via Bluetooth using IP address of other systems. This paper can further be extended to find the worms that the even changes the file names and change the scan traffic volume in the real time network.

References

[1] R. Vogt, J. Aycock, and M. Jacobson, "Quorum sensing and selfstopping worms," in *Proceedings of 5th ACM Workshop on Recurring Malcode (WORM)*, Alexandria VA, October 2007.

[2]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-june2011.

[3] http://www.icsi.berkeley.edu/~nweaver/warhol.old.html

[4]D. Moore, C. Shannon, and J. Brown, "Code-red: a case study on the spread and victims of an internet worm," in *Proceedings of the 2-thInternet Measurement Workshop (IMW)*, Marseille, France, November 2002.

[5]http://www.symantec.com/security_response/writeup.jsp?docid=2003-030812-5056-99

[6]http://www.theregister.co.uk/2003/03/10/network_worm_uses_weak_windows/

[7] Z. S. Chen, L.X. Gao, and K. Kwiat, "Modeling the spread of active worms," in *Proceedings of the IEEE Conference on ComputerCommunications (INFOCOM)*, San Francisco, CA, March 2003.

[8] Yudong Zhang, Lenan Wu, A Robust Hybrid Restarted Simulated Annealing Particle Swarm Optimization Technique, Advances in Computer Science and its Applications, vol.1, no.1, pp.5-8, 2012

[9] Chengwen Zhong, Min Zhong, Cunru Bai, A high-order discrete scheme of Lattice Boltzmann method for cavitation simulation, Advances in Computer Science and its Applications, vol.1, no.1, pp.73-77, 2012

[10] Priyadharshini Muthukrishnan, Sneha Raichel Mathew, Baskaran R, Suganya V, Message Level Security Realization in Web Services Using AES and Diffie Hellman Key Exchange, Advances in Computer Science and its Applications, vol.1, no.1, pp.78-83, 2012