Defending Polymorphic Worms in Computer Network using Honeypot

R. T. Goswami[a], Avijit Mondal[b]
[a,b] Department of Computer Science, Birla Institute of Technology Extension Centre, Kolkata, India–700107

Bimal Kumar mishra[c1], N.C. Mahanti[d]
[a,b] Department of Applied Mathematics, Birla Institute of Technology, Mesra, Ranchi, Indian-835

***Abstract*— Polymorphic worms are a major threat to internet infrastructure security. In this mechanism we are using gate-translator, double honeypot, sticky honeypot, internal translator and antivirus of Cloud AV,which attracts polymorphic worms. We are proposing an algorithm to detect and remove polymorphic worms and innocuous traffic related packets.**

Keywords- Polymorphic worm; Honeypot; Honeynet; Sticky honeypot; Cloud computing.

I. INTRODUCTION

Worms are computer programs that self replicate without requiring any human intervention, by sending copies of their code in network packets and ensuring the code is executed by the computers that receive it. When computers are infected, they spread copies of themselves and perform other malicious activities. A polymorphic worm is a worm that changes its appearance with every instance [1]. There are two basic types of intrusion detection: host-based and network-based. Host-based IDSs examine data held on individual computers that serve as hosts, while network-based IDSs examine data exchanged between computers [3, 4].

Security experts manually generate the IDS signatures by studying the network traces after a new worm has been released. Our research is based on Honeypot technique. Developed in recent years, honeypot is a monitored system on the Internet serving the purpose of attracting and trapping attackers who attempt to penetrate the protected servers on a network. Honeypots fall into two categories. A high-interaction honeypot such as (Honeynet) operates a real operating system and one or multiple applications. A low-interaction honeypot such as (Honyed) simulates one or multiple real systems. In general, any network activities observed at honeypots are considered suspicious [1, 2].

Security experts need a great deal of information to perform signature generation. Such information can be captured by tools such as honeynet. Honeynet is a network of standard production systems that are built together and are put behind some type of access control device (such as a firewall) to watch what happens to the traffic [1]. We assume the traffic captured by honeynet is suspicious. Our system reduces the rate of false alarms by using honeynet to capture traffic destined to a certain network.

The attackers will try every possible way to extend the life time of Internet worms. In order to evade the signature-based system, a polymorphic worm appears differently each time it replicates itself. This subsection discusses the polymorphism of Internet worms. There are many ways to make polymorphic worms [2]. One technique relies on self encryption with a variable key. It encrypts the body of a worm that erases both signatures and statistical characteristics of the worm byte string. A copy of the worm, the decryption routine, and the key are sent to a victim machine, where the encrypted text is turned into a regular worm program by the decryption routine. The program is then executed to infect other victims and possibly damage the local system. If the same decryption routine is always used, the byte sequence in the decryption routine can serve as the worm signature. A more sophisticated method of polymorphism is to change the decryption routine each time a copy of the worm is sent to another victim host. This can be achieved by keeping several decryption routines in a worm. When the worm tries to make a copy, one routine is randomly selected and other routines are encrypted together with the worm body.

The number of different decryption routines is limited by the total length of the worm. Given a limited number of decryption routines, it is possible to identify all of them as attack signatures after enough samples of the worm have been obtained. Another polymorphism technique is called garbage-code insertion. It inserts garbage instructions into the copies of a worm. For example, a number of nop (i.e., no operation) instructions can be inserted into different places of the worm body, thus making it more difficult to compare the byte sequences of two instances of the same worm. However, from the statistics point of view, the frequencies of the garbage instructions in a worm can differ greatly from those in normal traffic. If that is the case, anomaly-detection systems can be used to detect the worm. Furthermore, some garbage instructions such as nop can be easily identified and removed.

A Cloud AV: N-version antivirus identifies malicious software by multiple, heterogeneous engine in parallel to provide N-version protection. Cloud AV includes a light weight, cross platform host agent, with ten antivirus engine and two behavioral detection engines [5].

The attacker sends one instance of a polymorphic worm to a network, and this worm in every infection automatically attempts to change its payload to generate other instances. So, if we need to capture all polymorphic worm instances, we need to give a polymorphic worm chance to interact with hosts without affecting their performance. So, we propose new detection method “Double-honeynet” to interact with polymorphic worms and collect all their instances. The proposed method makes it possible to capture all worm instances and then forward these instances to the Signature Generator which generates signature.

II. SYSTEM ARCHITECTURE

In this architecture we used a double honeypot system to detect new worms. Following figure 1 shows the system architecture of the system. Firstly, the incoming traffic goes through the Gate Translator which samples the unwanted inbound connections and redirects the samples connections to Honeynet1. The gate translator is configured with publicly-accessible addresses, which represent wanted services. Connections made to other addresses are considered unwanted and redirected to Honeynet 1 by the Gate Translator. Secondly, once Honeynet 1 is compromised, the worm will attempt to make outbound connections. Each honeynet is associated with an Internal Translator implemented in router that separates the honeynet from the rest of the network. The Internal Translator 1 intercepts all outbound connections from honeynet 1 and redirects them to honeynet 2 which does the same forming a loop. Only packets that make outbound connections are considered malicious, and hence the Double-honeynet forwards only packets that make outbound connections.

This policy is due to the fact that benign users do not try to make outbound connections if they are faced with non-existing addresses. Lastly, when enough instances of worm payloads are collected by Honeynet 1 and Honeynet 2, they are forwarded to the Signature Generator component which generates Signature. Signature generator consists of two honeypots, one high interaction, one low interaction and a Cloud AV which consist of ten antivirus engine and two behavioral detection engine. Here we are using sticky honeypot in between honeynet 1,2 and honeynet 3 to minimize instance of worm propagation and to generate effective signature for the worm using CloudAV. If cloudAV unable to detect worms then unused IP address system is automatically quarantined [6-7].Since honeypot 3 has set of blocks of antivirus to remove future polymorphic worms, which are developed with the help of behavioral detection engine which is deployed on unused system continuously till the removal of polymorphic worms.

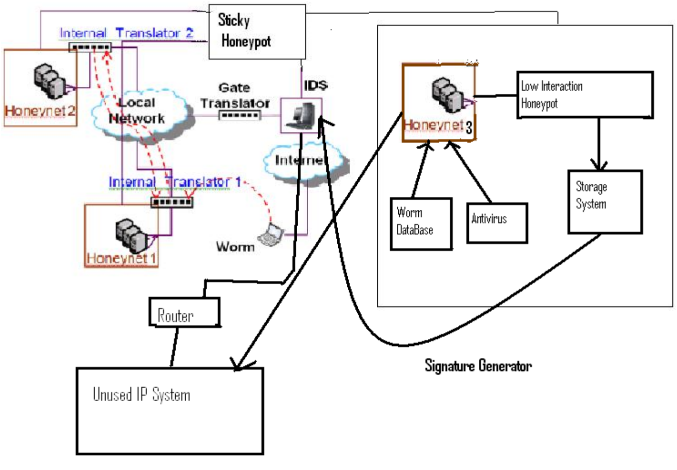

Fig 1: System Architecture

III. ALGORITHM

a. Gate-translator collects incoming traffic and redirects them towards honeynet-1.
b. Internal translator implemented in router that separates honeynet from rest of the network.
c. Internal translator 1 intercepts all outbound connections from honeynet 1 and redirects them to honeynet 2.
d. When enough instances of worm payloads are collected by honeynet 1 and honeynet 2, they are forwarded to the signature generator.
e. Signature generator consist of two honeypots(one high interaction and one low interaction).When collected payloads are transferred to the Honeypot 3 ,we used sticky honeypot in between them that will minimize the worm propagation and signature will be generated at Honeypot 3.
f. Honeypot 3 has CloudAV antivirus which consist of ten antivirus engines and two behavioral detection engines which continuously run at Honeypot 3.
g. All the signatures are transferred to the storage system through low interaction honeypot. Then IDS can get all information about that payload.
h. If Cloud AV at Honeynet 3 unable to remove those worms then unused IP address system is automatically quarantined.
i. On quarantined unused system, blocks of future worm's removal capabilities antivirus is run continuously till it is removed.
j. After removal of polymorphic worm unused IP address system is again connected to the network.

IV. CONCLUSION

We have defined an algorithm to detect and defend newly detected polymorphic worms. The framework is designed using double honeynet and sticky honeypot. To detect newly polymorphic worms, we have used CloudAV antivirus which consist of ten antivirus engine and two behavioral detection engine, that continuously run at Honeynet 3.The undetected worms will be automatically quarantined at unused IP address system. In future we want to propose an automated signature generation system for polymorphic worms. We have proposed new detection method "Double-Honeypot" to detect new worms that have not been seen before. The proposed system will be based on Principal Component Analysis that will determine the most significant data that are shared between all 'polymorphic worms' instances and use them as signatures.

REFERENCE

[1] L. Spitzner, "Honeypots: Tracking Hackers," Addison Wesley Pearson Education: Boston, 2002.

[2] Yong Tang, Shigang Chen," An Automated Signature-Based Approach against Polymorphic Internet Worms," IEEE Transaction on Parallel and Distributed Systems, pp. 879-892 July 2007.

[3] Snort – The de facto Standard for Intrusion Detection/Prevention, Available: http://www.snort.org, 14 February 2011.

[4] Bio Intrusion Detection System. Available: http://www.bro-ids.org/, 14 February 2011. International Journal for Information Security Research (IJISR), Volume 1, Issues 1/2, March/June 2011 Copyright

[5] John Oberheide.,Evan Cooke., Farnam .Jahanian., CloudAV: N Version Antivirus in the Network cloud, University of Michigan,Ann, Arhor, USENIX pp 1-18,2008.

[6] B.K.Mishra., N.Jha., SEIQRS model for the transmission of the malicious object in computer network, Applied Mathematical Modeling, 34, pp.710-715,2010.

[7] D.Moore.,C.Shannon., G.M.Valker., S.Savage., Internet Quarantine requirement for containing self replicating code ,Proceeding of the 22nd annual joint conference of the IEEE Computer and communication Societies, Infocom 2003, San Francisco, California, U.S.A, April, 2003.

[8] Cohen.F., Computer worms theory and experiment, Computer and Security, Vol 6,pp. 22-35,1987.

[9] Yong Tang and Shigang Chen., Defending Against Internet Worms: A Signature- Based Approach, Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL,USA., pp. 32611-6120,2010.

[10] Mohssen M. Z. E. Mohammed, H. Anthony Chan, Neco Ventura. “Honeycyber: Automated signature generation for zero-day polymorphic worms“; Proc.of the IEEE Military Communications Conference ,MILCOM, 2008.

[11] Tang,Y.; Chen, S. (2005). Defending Against Internet Worms: A Signature-Based Approach. In Proceedings of IEEE INFOCOM’2005, Miami, Florida, USA, pp.1-11.