Techniques Adopted for Containment of Polymorphic Worms – A Review

Leela. G. H
Department of EC,
Bapuji Institute of Engg. & Tech.
Davangere, K.S., India

T. D. Dongale
School of Nanoscience & Technology,
Shivaji University, Kolhapur,
M.S., India

R. K. Kamat
Department of Electronics,
Shivaji University, Kolhapur,
M.S., India

Abstract— Development in the computer network technology has also widened its application in the field of education, science and business. But for such a media insecurity exists due to polymorphic worms. It is necessary to evade such polymorphic worms. Various techniques have been developed by researchers to detect and stop these polymorphic worms. On the other hand, FPGA have a significance of parallel execution and reconfigurable architecture. Hence it is proposed to design and implement an FPGA based polymorphic worm containment system.

Keywords- Computer Network; FPGA, IDS; Polymorphic Worm.

I. INTRODUCTION

The development of the personal computer brought about tremendous changes for business, industry, science, and education. A similar revolution is occurring in data communications and networking. Technological advances are making communication links to carry more and faster signals. As a result, services are evolving to allow use of this expanded capacity. For example, established telephone services such as conference calling, call waiting, voice mail, and caller ID have been extended [18]. Such a media has security threat due to malicious software.

Malicious software is the software that is intentionally included or inserted in a system for a harmful purpose. Malicious software can be divided into two categories: those that need a host program, and those that are independent. The former are essentially fragments of programs that cannot exist independently of some actual application program, utility or system program. Viruses, logic bombs and backdoors are examples. The latter are self-contained programs that can be scheduled and run by the operating system. Worms and zombie programs are examples [19].

Worm is a program that can replicate itself and send copies from computer-to-computer across network connection. Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs disruptive or destructive actions. A worm seeks out more machines to infect and each machine that is infected serves as an automated launching pad for attacks on other machines. The types of worms’ technology include multiplatform, multiexploit, ultrafast spreading, polymorphic, metamorphic, transport vehicles and zero-day exploit.

To detect and deny these worms Intrusion Detection Systems (IDS) are developed which are deployed at the network edges. These IDS checks inbound traffic for known byte patterns (signatures) and if a worm arrives at the IDS, it raises an alarm [5]. But these IDS use fixed, contiguous substring set as signatures, which are not sufficient if the worm is polymorphic.

Polymorphic worms are those which change their payload on every infection. For denying a worm which is polymorphic in nature the available IDS are incapable due to its inability to match them in signature database. Hence it is necessary to update the database in IDS which can detect even the polymorphic worms. Section 2 provides an overview about various techniques adopted for denying polymorphic worms. Section 3 highlights significance of FPGA implementation.

II. TECHNIQUES FOR CONTAINMENT OF POLYMORPHIC WORMS

Worms uses network connection to spread from system to system and to detect and stop harmful worms the Intrusion Detection Systems are used. The Intrusion Detection Systems (IDS) checks the inbound traffic and accordingly deny the worms. These IDS identify the worms by byte pattern matching (signature). For earlier IDS these signatures were generated manually, but soon this became obsolete because the worm was identified only after it has attacked the host. Generating the signatures manually was a slow pace and even it was not suitable for detecting polymorphic worms—which change their payload on every attack; hence automated signature generation became necessary. Even though polymorphic worm changes its appearance on every attack, there exists some nature of worm which remains invariant which is exploited for detecting and denying it. Here is a review of techniques available in the literature.
A. Content based signature generation technique

Every worm has some invariant byte pattern which is used as signature for the worm detection. Some such techniques in the literature are as follows, Shield [26] uses manually generated vulnerability based signatures to filter out attack flows on a host. These have the limitation of manually generated signatures which is not suitable for the fast growing worms. Honeycomb [21] generates signatures consisting of a single contiguous substring of a worm's payload to match all worm instances – it uses longest common substring algorithm which looks for the longest shared byte sequences across pairs for connection. Kim et al., presents a system known as Autograph [22] which generates signatures automatically based on Rabin fingerprint that searches for repeated byte sequences by partitioning the payload into content block. Earlybird [23] system measures packet content prevalence at a single monitoring point such as a network DMZ and based on this generates signatures to detect worms. All these methods [21-23] automatically generates signatures for worms, but are single contiguous string and, often fail to match polymorphic worm payloads because they change payload on every attempt. To detect worm which are polymorphic in nature content based techniques like polygraph [2], hamsa [3] LISABETH [24] have been developed.

Newsome J et al., have used Polygraph- a suite of novel algorithms for automatic generation of signatures and match polymorphic worms [2]. These signature types include conjunctions of byte strings, token subsequences (substrings that must appear in a specified order, a special case of regular expression signatures) and Bayes- scored substrings. Zero day polymorphic worms pose a serious threat to the security of Internet infrastructures. Zhichun et al., have proposed [4] Hamsa-a network based signature generators can be connected to routers via a span port or an optical splitter for monitoring the traffic. LISABETH [23] is an improved version of hamsa. All these techniques generate automated signatures for polymorphic worms based on multiple invariant substrings. But these signatures are based on single instances of multiple worms. Hence they can detect only the known worms.

Yong Tang et.al [20] has adopted double-honeynet technique which includes two honeypots, one honeypot for inbound traffic with high interaction and other for outbound traffic with low interaction. Since the outbound honeypot is low interactive- it is not able to collect all the worm instances hence it is not able to generate an efficient signature. Whereas Mohssen et al., have proposed double honeynet [8] with high interactive honeypot for outbound connections, hence can collect sufficient amount of worm instances and then signatures for worm containment is done by using these instances. For signature generation different methods like protocol classifier [9], clustering based on destination port [7], substring extraction algorithm, an efficient algorithm that converts worm substrings into binary representations and using these binary representation for pattern matching [1], using principal component analysis technique[3] have been used to reduce the dimension of worm payloads, so only the most important signatures of the worm can be obtained and finally the authors have experimentally verified the same[8].

B. Anomaly Based signature generation technique

Here a virtual system is set to analyze the behavior of worm and this analysis is used for the detection of the similar worms. A specific worm after attacking a system goes in search of a system with similar vulnerability so this behavior is detected by the virtual machine and is used as signature for worm detection. The various researchers who have tried to detect the worm based on this are as follows :

Pan Xiaohui et al., have designed a hybrid method based on worms' propagation model. The propagation speed in the early stage is relatively smooth, considering this authors proposes a hybrid method for detecting polymorphic worm accurately in the early stage [11]. It combines port scan detection and emulation, port scan detects the suspicious packet and emulator first executes every instruction byte(s) and detects is it a worm or not.

Songging et al., proposes a Worm Terminator [14] which detects and contains the fast spreading worm based on its defining characteristic; a fast spreading worm will start to infect others as soon as it successfully infects one host. Worm Terminator also exploits by observation that a fast spreading worm keeps exploiting the same set of vulnerabilities when infecting new machines.

Stephenson B, et al., have proposed a model based on coevolution of the biological quasi-species to characterize the propagation of polymorphic worms. The model is used to derive the maximum allowable time of the IDS in order to contain the worm and the optimal mutation rate the worm should use in order to escape an IDS with a given response time[10][12].

C. Based on a specific characteristic of the exploit

Apart from the worm behavior or its invariant byte pattern their exists some more characteristics of the worm that it tries to alter the JUMP address by either varying some bits of the address or by inserting some bytes to attain buffer overflow and hence the return address is changed and which in turn change the execution flow and an attack occurs exploiting the vulnerability on that system. Based on this a worm is analyzed. Some of such techniques available in the literature are as below.

An architecture for generating semantics-aware signatures [25] described Nemean, which incorporates protocol semantics into the signature generation algorithm, this covers a wide class of attacks. Taintcheck [27] generates signatures automatically using information about the vulnerability and how it is exploited using semantic-based signature generation. This method can be used to identify other invariant parts for signatures such as protocol framing - it uses overwrite of the high order bytes of jump address. Buttercup [37] based on buffer overflow vulnerabilities but has a limitation to work for high speed internet worms. Wang et al., have proposed a technique to detect a worm that exploits buffer overflow vulnerability. Worms are detected through recognition of the JUMP address based on data mining such as Bayes and ANN technique [5]. The authors have proposed the model of
worm attack-the OSJMP (Overflow, Shellcode and JUMP) model. Tatara et al., have tried to detect the polymorphic worm by Abstract Payload Execution (APE) method – which determines the number of contiguous executable instruction i.e., based on the length of instructions [13].

III. IMPLEMENTATION ON RECONFIGURABLE HARDWARE

FPGA technologies are a growing area of research that has the potential to provide the performance benefits of ASICs and the flexibility of processors. Application Specific Hardware circuits can be created on demand to meet the computing and interconnect requirements of an application. Furthermore, FPGAs can be reconfigured very quickly, allowing their configuration to be altered according to the requirements of a computation. The complete device or a part of it can be reconfigured at runtime or before the computation commands. The evolution of the fabrication technology has led to the development of FPGAs with exceptional computational power at a lower cost [16]. A performance improvement is necessary along with an improved technique for denying the polymorphic worms. To improve the performance it is proposed to go with FPGA implementation because of its parallelism in execution and reprogrammable when the technique used is improved/modified. Nitish et al., [28], V. Paxson et al[29] Dandis et al.,[16] have implemented a polymorphic worms detection methods on FPGA platform. There exists very less work on implementation of Polymorphic worm detection on FPGA, hence, work needs to be focused on carrying out the implementation on FPGA which can be configured using a HDL language.

IV. CONCLUSION

Among the available techniques, an advantage of content-based signatures is that they allows us to treat the worms as strings of bytes and does not depend upon any protocol or server information. They can be easily incorporated into firewalls or Intrusion Detection Systems. Hence it is proposed to consider a content based detection technique. In the developmental work towards the design of FPGA based detecting systems, there has been less work. Because the research community requirements are often rapidly changing and increasing in complexity, it is necessary for any solution to be rapidly designed and modified, portable to the latest, most powerful processing platform, it is intended to design a content based signature technique which can detect even an unknown worm and implement it using a reconfigurable architecture like FPGA.

REFERENCES


workshop on Software engineering for secure systems, Leipzig, Germany, May 2008.


