基于基因规划的主机异常入侵检测模型*

苏璞睿+, 李德全, 冯登国

(中国科学院 软件研究所 信息安全国家重点实验室,北京 100800)

A Host-Based Anomaly Intrusion Detection Model Based on Genetic Programming

SU Pu-Rui+, LI De-Quan, FENG Deng-Guo

(State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: 86-10-62528254 ext 801, Fax 86-10-62520469, E-mail: supurui@263.net

http://www.iscas.ac.cn

Received 2002-04-22; Accepted 2002-09-17

Su PR, Li DQ, Feng DG. A host-based anomaly intrusion detection model based on genetic programming. *Journal of Software*, 2003,14(6):1120~1126.

http://www.jos.org.cn/1000-9825/14/1120.htm

Abstract: Anomaly Detection techniques assume all intrusive activities deviate from the norm. In this paper a new anomaly detection model is found to improve the veracity and efficiency. The proposed model inestablishes a normal activity profile of the systemcall sequences by using Genetic Programming. One instance of the model monitors one process. If the model finds the real systemcall sequences profile of the process deviating from the normal activity profile, it will flag the process as intrusive and take some actions to respond to it. And a new method of calculating the fitness and two operators to generate the next offspring are provided. According to the comparison with some of current models, the model is more veracious and more efficient.

Key words: intrusion detection; genetic programming; anomaly detection

摘 要: 异常检测技术假设所有的入侵行为都会偏离正常行为模式.尝试寻找一种新的异常入侵检测模型改善准确性和效率.模型利用应用程序的系统调用序列,通过基因规划建立了正常行为模式.模型的一个例程管理一个进程.当它发现进程的实际系统调用序列模式偏离正常的行为模式时,会将进程设标记为入侵,并采取应急措施.还给出了基因规划的适应度计算方法以及两个生成下一代的基本算子.通过与现有一些模型的比较,该模型具有更好的准确性和更高的效率.

关键词: 入侵检测;基因规划;异常检测

中图法分类号: TP309 **文献标识码**: A

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1999035802 (国家重点基础研究发展规划(973)); the National Foundation of China for Palmary Youth under Grant No.60025205 (国家杰出青年基金)

SU Pu-Rui was born in 1976. He is a Ph.D. candidate at the Institute of Software, the Chinese Academy of Sciences. His research interest is network security. **LI De-Quan** was born in 1969. He is a Ph.D. candidate at the Institute of Software, the Chinese Academy of Sciences. His research interest is network security. **FENG Deng-Guo** was born in 1965. He is a professor and doctoral supervisor at the Institute of Software, the Chinese Academy of Sciences. His research area is information security.

The goal of an intrusion detection system (IDS) is to identify unauthorized use, misuse, and abuse of computer systems or networks by internal or external users in nearly real time. Host-based intrusion detection started in the early 1980s before networks were as prevalent, complex and interconnected as they are today. In this simpler environment, it was a common practice to review audit logs for suspicious activity. Today's host-based intrusion detection systems remain a powerful tool for understanding previous attacks and determining proper methods to defeat their future application. Most of the Host-based IDSes still use audit logs, but they are much more automated, having evolved sophisticated and responsive detection techniques[1].

Anomaly Detection techniques assume all intrusive activities deviate from the norm. These tools typically establish a normal activity profile (a statistical model that contains metrics derived from system operation) and then maintain a current activity profile of a system[2]. Observed metrics that have a significant statistical deviation from the model are flagged as intrusive. When the two profiles vary by statistically significant amounts, an intrusion attempt is assumed.

In this paper, we introduce an anomaly host-based intrusion detection model based on Genetic Programming. It uses the systemcall sequence patterns to establish the normal activity profile, which is generated by the Genetic Programming algorithm according to the systemcall sequences of the application collected in absolutely secure environment. And It watches the real profile of the process. When it finds the real profile deviating from the norm, it flags the system as intrusive and takes some actions to response it.

The rest of the paper is organized as following. We begin with the introduction of the framework of the model. In section 2, we will discuss the data source of the intrusion detection model. And in section 3, we will describe some specifications of the pattern lib and Genetic Programming used to generate the pattern lib. In section 4, we will introduce the Analyzer and the Responsor. In section 5, we will discuss the related works. In section 6, we will finish with the conclusions and future work.

1 Framework of the Model

Because of the prevalence of the network, the maximal menace to the system comes from the network. The daemon applications which provide the network service are the only way the attackers entering the host system. And most of the attacks can only be brought into effect by the process running in the target host. So the processes running in the host are the most important things we should protect.

The following anomaly intrusion detection model detects intrusions according to the systemcall sequences of the key processes running in the system, and one instance of the model monitor one process. The host-based anomaly intrusion detection model includes four parts:

The sensor, is responsible for collecting data, the systemcall sequence of the process, in nearly real time.

The pattern lib, is the normal activity profile of the process which consists of many pieces of systemcall sequences.

The analyzer, analyzes the data the sensor has collected and determines whether the system is being or will be attacked according to the pattern lib.

The reponsor, reports the alert message when the analyzer finds the clue of attack. And they are organized as the Fig.1.

The Sensor traces the process, and pushes the systemcall into the systemcall queue of the process in turn. Parallelly, the analyzer gets a systemcall sequence with the length of L to analyze, and the point of the queue increases one. The analyzer analyzes all the systemcall sequences according to the Pattern Lib, which defines the normal activity of the process. The algorithm the analyzer uses will be discussed in section 4. If the analyzer finds the clue of the attack, it sends the alert message to responsor, which is responsible for reporting the message to user

in user's favorite way.

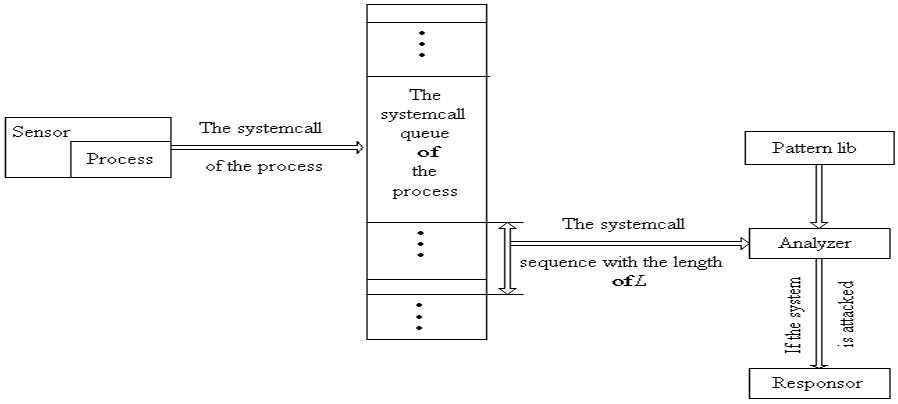


Fig.1

2 The Data Source

The Data Source is the most important thing for the intrusion detection system, which determines the performance of the system. The Data Source of the anomaly intrusion detection system should meet the following requirements:

It should not be too diverse. If the data source is too diverse, it's difficult for the system to define the normal or legal manner of the system. If the lib which defines the normal activity profile is too massive, it will decrease the performance of the intrusion detection system.

It can be used to represent the manner or state of the user or the system. The data source, which is not concerned with system activity and security should not be chosen.

It should be easy to collect. The data source which can be collected in near real time will be better The earlier we have collected the data source, the earlier we could find the attack.

It could not be too massive. If the data collected by the sensor in unit time is too massive, the intrusion detection system could not parse it effectively, and could not find the attack in time.

According the requirements, we choose the systemcall sequences of the process as the data source. The systemcall sequences have the following characteristic:

The systemcall sequence adequately represents the manner of the process, which is much better than the CPU usage, storage, and so on;

The set of the systemcall sequence of a application is limited compared with other data source, such as the traffic of the network. It is easier to define the normal manner by the sequence than others. And the size of the lib used to define normal manner will be much smaller than others.

The systemcall sequence could be collected almost in real time, so it is helpful for analyzer to find the attack in time. So, the systemcall sequence is a good data source for anomaly intrusion detection.

3 The Pattern Lib

The Pattern Lib is an important part of the model, which defines the normal activity profile of the process. Whether the intrusion detection system is good depends much on whether the lib is exact and comprehensive. We use the systemcall sequences to construct the lib.

In the Pattern Lib, each pattern is made up of systemcall numbers and wildcard characters, represented by number –1, and the length of each pattern is L. In Linux, there are 250 systemcalls, numbered from 0 to 249.

In this paper,we introduce Genetic Programming[3] to generate the Pattern Lib. And the algorithm is illustrated in Fig.2. Before discussing the algorithm, we define some arguments:

N_s — the number of all the possible systemcalls; In Linux, N_s=250

A_L — the set of all the possible systemcall sequences with the length L

$S(Q)$ — the size of the set Q,

$$S(A_L) = N_s{}^L$$

$E(Pattern_i)$ — the numbers of the systemcall number in $Pattern_i$, except the wildcard character; for example,

If

$Pattern_i$ = 23 24 * 36 * 39 40 33 32

Then

$E(Pattern_i) = 7$

$M(Pattern_i,Q)$ — the number of systemcall sequences in the set Q that matches the $Pattern_i$;

$P(Pattern_i,Q)$ — the probability of $Pattern_i$ matching the systemcall sequeces in the set Q;

$P(Pattern_i,Q) = M(Pattern_i,Q)/S(Q)$;

The algorithm is as following:

1) Generate a set of L-length systemcall sequences of the application, T. The set should be generated on the following conditions:

I) The environment that the application is running in is absolutely secure;

II) The application should execute all kinds of possible legal operations. The more comprehensive the possible legal operations are executed, the better the set T is.

2) Generate the Pattern Lib by Genetic Programming, which is explained as Fig.2.

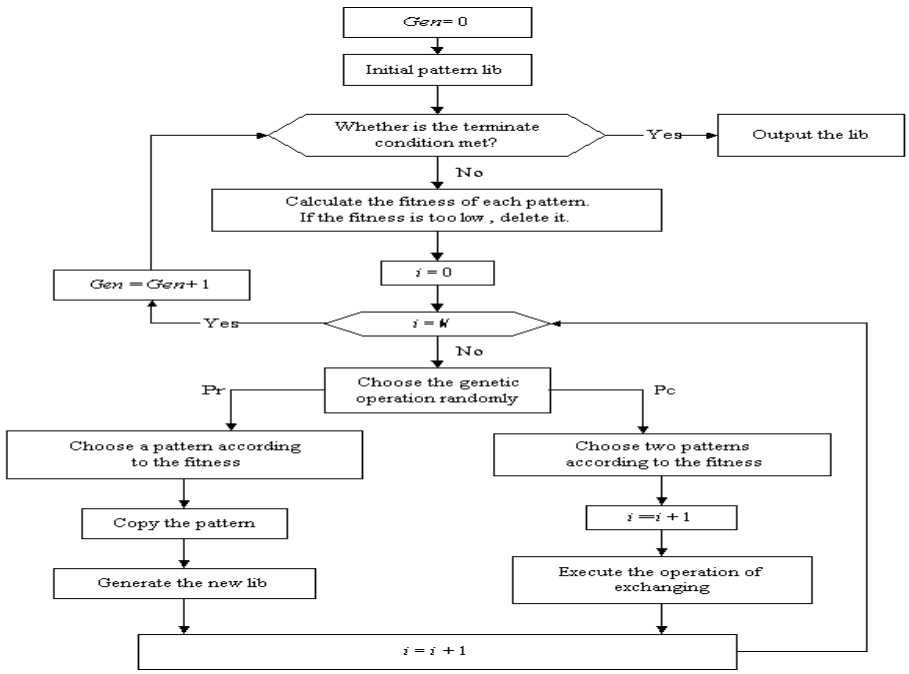


Fig.2

I) Generate the initial pattern lib, which consists of the systemcall patterns generated randomly. Each pattern is

made up of systemcall numbers and wildcard characters.

II) Calculate the fitness F_i of each element of the lib. The fitness F_i of $Pattern_i$ is calculated as following expression:

$$F_i=P(Pattern_i,T)/P(Pattern_i,A_L)$$

And

$$P(Pattern_i,A)=M(Pattern_i,A)/S(A_L)$$
$$= N_s^{L-E(Pattern_i)}/S(A_L)$$
$$= N_s^{L-E(Pattern_i)-L}$$
$$= N_s^{-E(Pattern_i)}$$
$$P(Pattern_i,T)=M(Pattern_i,T)/S(T)$$

So

$$F_i=M(Pattern_i,T)/[S(T) * N_s^{-E(Pattern_i)}]$$
$$=M(Pattern_i,T) * N_s^{E(Pattern_i)}/S(T)$$

If the fitness F_i is too little, $F_i<f$, we delete the $Pattern_i$ from the pattern lib.

III) Randomly choose one genetic operation from the following to generate the next offspring:

i) According to the fitness F_i, choose one element to copy as a element of the next offspring;

ii) According to the fitness F_i, choose two elements which exhange part of the sequence to generate the two new sequences of the next offspring. Just as following(all the systemcall sequence examples are collected from sendmail):

197 * 125 90 6 5 * 90 197 exchange 197 * 125 90 5 197 192 3 3

192 3 6 * 5 197 192 3 3 192 3 6 * 6 5 * 90 197

The point P underlined is chosen randomly. And P should be greater than 1, because if P=1, the exchanging will generate two patterns same as chosen

There are two arguments Pr and Pc, to control the probability of the elements being chosen to execute i) or ii).

IV) Repeat II, III until the terminate condition is satisfied. For the terminate condition, there are two choices: The first is controlling the number of the offspring, Gen. If $Gen>=M$, the algorithm should be terminated; and the second, we can terminate the algorithm when the lib is in relative stabilization.

3) Clear up the lib. At last, we delete some patterns which are covered by others to increase the system performance. For example, in the following patterns:

221 * 197 192 3 * 91 13

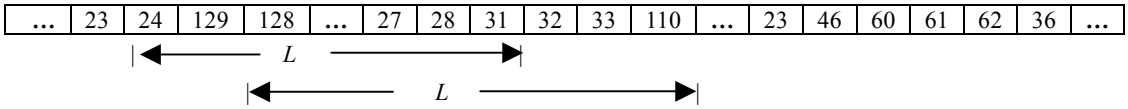
221 221 197 192 3 * 91 13

221 * 197 192 3 6 91 13

The first pattern has included the other two patterns, so we delete the last two patterns.

4 The Analyzer and Responsor

The analyzer judges whether the process is attacked or not according to the queue of systemcalls and the pattern lib of the application. The algorithm is described as following:



If in the continuous C_s L-Sequences of systemcall, there are $l_{\min}$ L-Sequences of systemcall not matching any pattern in the Pattern Lib, the analyzer concludes that the process is or will be attacked, and sends the alert message to the responsor.

The host is the best location to respond to any attacks. The reponsor of the host-based intrusion detection system has much more advantages than the network-based. The host-based IDS can get a fine granularity of information more easily than network-based, such as who is accessing the system and when the users log in and out of servers. It is appropriate for protecting an individual computer systems and the information it contains.

Because the host-based IDS can easily get the information of the user, the responsor can take some measures about the user to prevent the user's father attack. The network-based just can take some measures about the link which it finds suspicious. According the kinds of information and the alert message, the reponsor can choose one or more of the following operations:

1) Notify the administrator;
2) Terminate the process;
3) Set the user invalid and prevent the user from logging in
4) Delay, replace or abort the systemcall
5) Others

In the listed options, 4) is the especial response of the model using systemcall as the data source. Because it traces the systemcall of the process, it can control the systemcall entry and the return value. Then it can control the process's running.

5 Related Work

With the network becoming more and more prevalent, many network intrusion detection systems come into being. Snort is a successful one of them[4]. It is a misuse network intrusion detection system which uses the pattern matching to find attack or attack intention. Now it also adopts some anomaly detection techniques to find the attacks, such as DOS, port scanning, and so on. It chooses the network traffic as the data source. But with the bandwidth increasing, Snort encounters its development bottleneck. Because of the detection algorithm, Snort could not be used to monitor the network with high bandwidth. In the model in this paper, the systemcall sequence is chosen as the data source, which is much less than the traffic of the network.

And it is another shortcoming of Snort that it cannot find the attack which is not known. The intrusion model in this paper finds the attack by analyzing the data according to the normal activity profile, so it can find all the attacks deviating from the normal activity profile, including known and unknown.

The model in this paper is motivated by the research of Stephanie Forrest[5~7], in University of New Mexico. Forrest proposed and used a negative selection algorithm for various anomaly detection problems. The algorithm defines "self" by the pattern lib of normal activity profile. It generates many detectors which are patterns generated randomly and do not match any self pattern. And the detectors monitor the subsequent profiled patterns of the monitored system. During the monitoring stage, if a detector pattern matches any newly profiled pattern, it is considered that there is an anomaly in the monitored system. In the model in this paper, we use the Genetic Programming to generate the pattern lib. We introduce the wildcard in the pattern, decrease the number of the patterns, and improve the performance of the intrusion detection system.

Based on the Forrest's theory, Lee Wenke, in Columbia University, has brought forward a new detection model, which using Data Mining to generate the rules lib of the application's systemcalls[8,9]. In the model, the user should supply with a set of training data containing pre-labeled "normal" and "abnormal" sequences as training data. It applies the RIPPER, a rule learning program to the training data to induce the rule sets for normal and abnormal systemcall sequences. And then it uses a post-processing scheme to detect whether a given trace based on the RIPPER predications of its constituent sequences. In the Lee Wenke's Data Mining Model, we have to prepare many pre-labeled "normal" and "abnormal" sequences as training data. The more comprehensive the training data are, the

more the result of Data Mining will be. The normal sequences are easy to get. But in order to get the abnormal sequences, we have to simulate kinds of attack, and analyze the sequences manually. It's difficult for the user to collect enough abnormal sequences of the application, especially for the new products. In the model in this paper, we only need the normal sequences to establish the normal activity profile. The normal sequences are much easier to get than the abnormal sequences. We can develop a tool which can collect the normal sequences of kinds of applications.

6 Conclusions and Future Work

In this paper, we introduce an anomaly host-based intrusion detection model. It is motivated by the research of the Forrest. It chooses the systemcall sequence as data source, and adopts Genetic Programming to generate the normal activity profile. First, we collected a mass of systemcall sequences of the application in absolutely secure environment. We use Genetic Programming to generate the normal activity profile according to the sequences And then the analyzer analyzes the sequences collected by the sensor in nearly real time, according to the activity profile. If the sequences deviate from the profile, we flag the status of the process as intrusive. And the responsor will report it to the user.

In this paper, we have discussed the method of calculating the fitness, and introduced two operator ,copy and exchange, to generate the offspring in Genetic Programming. In the future, we should design much more and much better operators to improve the quality of the pattern lib generated by the algorithm. And we also should improve the algorithm of the analyzer to increase the performance and veracity of the system.

References:

[1] ISS Documents. Network- vs. Host-based Intrusion Detection. 1998. http://documents.iss.net/whitepapers/ nvh_ids.pdf.

[2] IATF Release 3.0, Host-Based Detect & Respond Capabilities Within Computing Environments. 2000.

[3] Yun QX, Huang GQ, Wang ZQ. Genetic Algorithm and Genetic Programming. Beijing: Publishing House of Metallurgy Industry, 1997 (in Chinese).

[4] Martin R. Snort-Lightweight intrusion detection for networks. 1999. http://www.snort.org/docs/lisapaper.txt.

[5] Forrest S, Hofmeyr SA, Somayaji A, Longstaff TA. A sense of self for Unix process. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, 1996.

[6] Hofmeyr SA, Forrest S. Architecture for an artificial immune system. Evolutionary Computation Journal, 2000,8(4):443~473.

[7] Warrender C, Forrest S, Pearlmutter B. Detecting intrusions using system calls: Alternative data models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy. 1999.

[8] Lee W, Stolfo SJ. Data mining approaches for intrusion detection. In: Proceedings of the 7th USENIX Security Symposium. 1998.

[9] Lee W, Stolfo SJ, Mok KW. A data mining framework for building intrusion detection models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy. Oakland, CA, May 1999.

附中文参考文献：

[3] 云庆夏,黄光球,王战权.遗传算法和遗传规划.北京:冶金工业出版社,1997.