Registry access anomaly detection system based on the rough set algorithm

Mingshu Zhang, *Bin* Wei*, and *Longfei* Liu

College of Cryptographic Engineering, Engineering University of Armed Police Force, Xi'an Shaanxi, China

Abstract. With the rapid growth and the popularization of the Internet, network security problems become increasingly serious. This paper analyzes the impact on several malicious codes on registry access behaviour and builds a rough set of algorithms-based registry access intrusion detection systems. Related attributes are constructed considering both the time sequence attribute and one-time access attribute as our input of the training module. Attribute reduction and rules extraction using rough sets algorithms give the registry access to normal behaviour modal. Experiments show that the system differentiates normal and abnormal registry behaviour successfully.

1 Introduction

While people are enjoying the various material and spiritual fruits brought about by networking, at the same time network security issues have become increasingly prominent. Malware spreads through e-mail and instant messaging software, making it hard to guard against. The emergence of phishing websites has also caused incalculable losses to consumers and businesses [1]. It can be seen that the issue of network security has received a high degree of national attention.

It is necessary for the generation and development of an intrusion detection system (IDS), which is an active defense technology to detect and prevent intrusion [2], and it plays an important role in computer network security systems [3]. The registry plays an important role in the configuration and control of the Windows system. It is a system-wide and user-set storage library and is a window to understand the various memory structures maintained by the Windows executive program and the kernel [4]. However, because 1) On Windows, all programs (including virus programs) must operate the registry when they are running; 2) More and more hacker programs to turn their attack targets for the registry; 3) The registry is a Windows operation one of the most frequently used parts of the system, all computer operations will affect the registry; 4) Most Windows programs only access some specific keys; for other reasons, it is also a very effective data source for intrusion detection [5].

Therefore, this paper designs a network security defense system based on the rough set algorithm for registry access. The system combines the time sequence characteristics of the

* Corresponding author: weibin82@126.com

registry behavior and the one-time access feature. The rough set algorithm is used to simplify the attributes and extract the rules. The normal behavior model of registry access is established, and finally, the effectiveness of this system is proved through experiments.

2 Intrusion detection model

2.1 Registry normal behavior model

To detect the abnormal access behavior of the process to the registry, we must first establish a normal access model of the registry. The original data we can get from the system sensor module contains five characteristics: process, num, query, path, return.

To detect the abnormalities listed above, we not only need to check whether the combination of Process, Path, and Query is legal in one visit but also check whether the combination of these characteristics for several consecutive visits is legal.

First, we separate the original data according to the process, model each process separately, and remove some less important attributes based on experience, including the access sequence number and return value.

Then, for the Path obtained during each visit, we further extract its top 5 levels as its parent directory item.

We consider 3 consecutive visits each time and form an access group (Access Group) so that there are 9 attributes in each access group: Path 1, Parent 1, Query 1, Path 2, Parent 2, Query 2, Path 3, Parent 3, Query 3. As the input of the modeling training algorithm.

At last, we can get the normal rules of the decision table access behavior. There are 33 items in total. The rule set composed of these rules can be used to describe the normal access behavior of the registry, which is called the normal model of registry access behavior.

2.2 Anomaly detection model

From the influence of malicious programs on registry access behavior, it can be seen that malicious programs often cause the appearance of unknown processes. In this paper, we use misuse detection (the steps are as follows) to realize the determination of unknown processes.

Algorithm 1

1. For a process to be detected, determine whether its access mode to the system registry is in the misuse rule base.
2. If it is not in the rule base, it is regarded as a normal visit, and the visit is accepted.
3. If the visit is in the misuse rule base, it is regarded as an abnormal visit, an exception is generated, and it is dealt with accordingly.

For known processes, to improve the overall operating efficiency of the system, we only perform anomaly detection on key processes. The anomaly detection algorithm based on the normal model of registry access behavior is:

Algorithm 2

1. For a process to be detected, a detection tuple is constructed from the current registry access and the previous two registry access behaviors.
2. In the ruleset representing the normal model, look for a rule that matches this tuple.
3. If there is no such rule, the prediction result that appears most frequently in the training set is selected as the prediction for the decision attribute.

4. If there is more than one rule that can be matched, the final prediction result is determined by voting. Each rule casts one vote for its prediction result, and the prediction result with the most votes is selected as the final result.

5. If the predicted result is the same as the decision attribute of the sequence segment, the prediction is successful, and sf=0 is returned.

6. If the predicted result is different from the decision attribute of the sequence segment, the prediction fails, and sf=1 is returned.

7. Do LFC for 10 consecutive sf. If the LFC value exceeds a threshold ζ, it is calibrated as an abnormality.

2.3 Misuse detection module

The key to this step is to construct an abnormal access behavior pattern library for the registry. The main aspects of the abnormal pattern library include the following:

1) Protection of system automatic operation items.
2) IE protection of registry entries.
3) The system sets the protection of related items.
4) File association protection.

Algorithm 3

1. A detection tuple is formed for a process to be detected, especially the current access behavior.

2. In the misuse detection pattern library, look for a rule that matches this tuple. 3. If there is no such rule and the match fails, sf=0 will be returned

4. If there is such a rule and the match is successful, it will return sf=1

5. Perform LFC for 5 consecutive sf, if the LFC value exceeds a threshold ζ, it is calibrated as an abnormality.

3 Result

3.1 Dataset

The dataset used in the experiment is collected from a test web forum and FTP server website built by the author.

3.1.1 Generating normal data

Use the Win registry to access the sensor to collect the registry accesses behavior data onto the key processes of the operating system and the processes that provide network services. To ensure the completeness of normal data, various possible normal accesses and operations are performed on the server:

1) The web forum was visited, registered, logged in, and left a message.
2) Managed the IIS server-side.
3) Operations such as connection, download, upload, delete, and new creation to the FTP server were performed.
4) Serv-u server-side management work.

3.1.2 Generation of abnormal data

Scan the web server for the following possible vulnerabilities: Unicode encoding vulnerabilities, FrontPage extensions, trying to obtain SAM files, trying to obtain PcAny Where password files, and data obtained from CGI vulnerability scanning. First use UNICODE, secondary coding and other vulnerabilities to upload the elevated ISAPI program to an executable directory of IIS, such as /scripts, and then use the process ispc.exe to connect and use the permissions of IIS5.0+SP0 (SP1, SP2) check for vulnerabilities to gain system privileges. ServU Hectic attacks exploiting serv-u buffer overflow vulnerability.

3.2 Results and discussion

Normal data test: There is no abnormal process, and the abnormality of all processes is at a very low level.

Table 1. Normal data test.

Number of records	Number of processes	Abnormal process
16327	35	/

Abnormal data 1 tests: Streamer scans the following possible vulnerabilities in the IIS web server. Intrusion detection of the data shows that this attack is manifested in the process dllhost.exe and the process cmd.exe. Abnormal data 2 tests: Use IIS5.0 permission check vulnerabilities to obtain system permissions. Intrusion detection of this data shows that this attack is manifested in the process cmd.exe and the process lsass.exe. Abnormal data 3 tests: An unsuccessful HTTP flood attack using Hackertools.

Table 2. Abnormal data 1 test.

dataset	Number of records	Number of processes	Abnormal process
Abnormal data 1	837	5	Dllhost, cmd
Abnormal data 2	312	6	Cmd, lsass
Abnormal data 3	33748	4	Dllhost, cmd

Through the above experiments, the behavior of malicious software is often significantly reflected on the registry, and a malicious attack is often manifested in the registry access behavior of multiple normal processes and may lead to the appearance of some unknown processes. Through the above experiments, it can be seen that using the registry accesses behavior model to detect abnormalities caused by malicious code is a simple and effective method.

4 Summary

This paper analyzes the parameters related to the registry and obtains the registry behavior data of the system. In the process of data modeling, we considered the information about one access behavior of the registry and the time sequence information about several consecutive access behaviors at the same time. In addition, the rough set theory is introduced into the modeling process of the system registry to access the normal model in the detection, only a small part of the normal data is needed for training, and a simple normal prediction rule model can be obtained, and the experiments prove that the model is effective to distinguish the registry accesses behavior caused by normal and malicious code. Because the model obtained is simple, this model is used for real-time detection and has little impact on system performance. It is an efficient and low-load detection method.

References

1. Ke Bao; Yourong Ding. Network security analysis using big data technology and improved neural network. Journal of Ambient Intelligence and Humanized Computing, 2020. p.1-11
2. Zhao, J.-H.a.W.-H.L., Intrusion Detection Based on Improved SOM with Optimized GA. Journal of Computers, 2013. 8(6): p. 1456-63.
3. Chung, C.-J., et al., NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems. Ieee Transactions on Dependable and Secure Computing, 2016. 10(4): p. 198-211.
4. Hubballi, N., S. Biswas, and S. Nandi, Towards reducing false alarms in network intrusion detection systems with data summarization technique. Security and Communication Networks, 2018. 6(3): p. 275-285.
5. Ruijuan, Z., et al., Analysis and application of Bio-Inspired Multi-Net Security Model. International Journal of Information Security, 2015. 9(1): p. 1-17.