A Rule-based Approach for Port Scanning Detection

Urupoj Kanlayasiri, Surasak Sanguanpong and Wipa Jaratmanachot

Applied Network Research Group
Department of Computer Engineering, Faculty of Engineering
Kasetsart University, Chatuchak , Bangkok, Thailand 10900.
Phone (66-2) 9428555 Ext. 1433 Fax 5796245, E-mail: g4265106@ku.ac.th

Abstract

Intrusion detection has been performed at network and host level for detecting various attacks. Port scanning could be classified as one of the network intrusions. This paper presents a method for detecting port scanning attacks using rule-based state diagram techniques. A set of rules corresponding with the appropriate thresholds was designed for intrusion decision. Experiment results under real environment show that port scanning patterns are successfully detected in real-time.

Keywords: port scanning, rule-based state diagram, intrusion detection, host-based intrusion detection system, network security

1. Introduction

An intrusion detection system (IDS) is an automated system intended to detect computer intrusions. The main goal of IDS is to identify, preferably in real time, unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrators [1]. There are two domains of intrusion detecting techniques: misuse detection and anomaly detection. Misuse is an attempt to recognize the well-known flaws or vulnerabilities of software or computer system. In this way, it can detect the general attack signatures that stem from the known attack such as exploiting a software bug. Anomaly detection, on the other hand, can be identified intrusions by unusual behavior of operations. According to Kumar's paper [2], "Anomaly detection attempts to quantify the usual or acceptable behavior and flags other irregular behavior as potentially intrusive"

Traditionally, there are two categories of IDS grouped by audit source location, host-based and network-based intrusion detection system. The host-based IDS monitors a single host machine employing the audit trails of a host operating system as a main source of input. It was regarded as a forerunner of the network-based intrusion detection system. The host-based IDSs, which have been widely developed in the past several years, can detect both anomaly and misuse behaviors. Generally, they often appear as the system embedded in a risky machine. The host-based IDS architecture is shown in Figure 1.

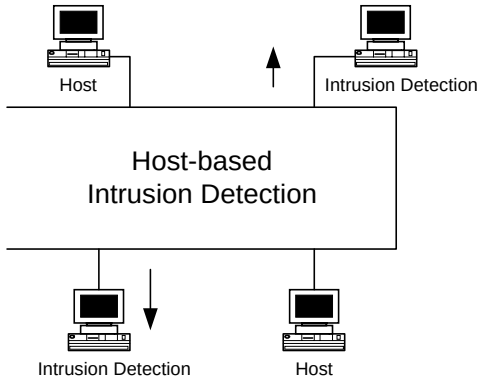

Figure 1. Host-based intrusion detection system

The network-based IDS monitors any numbers of hosts in network segments. It peruses the audit trails of multiple hosts and network traffic to identify the intrusion signatures. This novel approach is a stand-alone system that can detect an intruder invaded into any systems via a computer network. Unlike the host-based IDS, it does not depend upon any operating system.

The rest of this paper is organized as follows. In Section 2, the related works and relevant researches are provided. In Section 3, we describe the basic concept of port scanning attack and then propose the idea design and system architecture. The system testing is provided in Section 4. Finally, Section 5 gives conclusion and future works.

2. Related works

The intrusion detection research topic has been an increased growing interest since the last decade. A vast number of detection approaches are proposed [3]. Different methods were operated in different ways. Several theories were applied to contribute a more powerful detection but there is no one best solution that can cover all penetrations. Each approach may be technically appropriate to identify a specific scenario of security violations. Some techniques have to fuse with the others to further detection. Currently, there are many approaches for intrusion detection, such as Threshold Detection, Anomaly Detection, Misuse Detection, Rule-based Detection and Model-Based Detection.

Threshold detection is a simple method to identify the intrusion. It is one of the most rudimentary forms of detection. The key point of this technique is to alarm when the number of occurrences of suspicious event surpasses a reasonable threshold. When implementing intrusion detection using this approach, it is difficult to select the suitable measured threshold for each specific event. Threshold analysis is a poor detector of intrusion detection. Therefore, it is often used as a sub-component operation to enhance the efficiency of a large IDS. The Network Anomaly Detection and Intrusion Reporter or NADIR [4] is an example of threshold detection.

Anomaly detection has been widely used in recent years. The objective of this method is to separate the abnormal event from the normal activity. It defines the normal pattern within auditing trails. System and/or user profiles are established from the normal usage over duration of time. Typically, the two primary types of anomaly detection are statistical profile-based and rule-based [5]. Profile-based anomaly detection employs statistical method to identify the behavior but the rule-based detection utilizes sets of rules instead. The well-known profile-based anomaly detection system is the Intrusion Detection Expert System (IDES) The Wisdom and Sense (W&S) is the example of rule-based anomaly detection system.

To detect the exactly known vulnerabilities, misuse detection is selected. Due some detection tool can not efficiently detect some type of intrusions. The misuse detection appears as a sub-component in most intrusion detection system. Intrusion signatures are usually specified as a sequence of conditions and events that lead to a break-in. Most systems use this approach to deal with the adversary by employing the pattern-matching technique. The rule-based detection system decides the event as a penetration when audit records are parsed the predefined rules. A rule or a sequence of rules describes the suspicious event that becomes an attack as an IF-THEN expression.

The model-based detection represents the dangerous scenario as a high-level abstraction, not same as an audit record. The main goal is to build the model identified the behavior of intrusion. Model-based technique differs from current rule-based technique, which simply attempts to match the pattern with audit record.

3. Methodology

In our approach, the rule-based state diagram technique is proposed. We design rules, thresholds and system architecture, then develop our host-based intrusion detection system to detect the port scanning attack. Finally, we test our system with real attack situations.

3.1 Port scanning

Port scanning attack is a method for discovering exploitable communication channels that has been used for a long time. The key idea is to probe the network ports and then keep the information about them that are useful for an attack. In some viewpoints, port scanning is not regarded as a network intrusion but it is considered as the method for finding the possibilities to adverse system. At this time, there are many techniques to accomplish port scanning probes [6] such as, TCP connect scanning, TCP half-connect scanning, Stealth scanning, Xmas Tree scanning and NULL scanning. All above techniques require TCP packet to complete scanning.

Table 1 shows the flag setting and characteristics of connection request and reply packets used in each technique. To clarify how they work, we give a brief summary as follows.

Table 1. Characteristics of connection request and reply packets of well-known port scanning techniques

Technique	Flag	Listening port	Closed port
TCP connect	SYN	SYN/ACK	RST/ACK
TCP half-connect	SYN (then RST)	SYN/ACK	RST/ACK
Stealth	FIN	-	RST/ACK
Xmas Tree	URG/PSH/FIN	-	RST/ACK
NULL	No flags	-	RST/ACK

TCP connect scanning is the rudimentary method using the three-way handshaking establishment. The `connect()` system call is used to open a connection to an interesting port on the machine. First, the attacker sends a packet with SYN flag to a victim then waits for a reply packet. To identify the opening port, this technique calls for the target host to acknowledge with SYN/ACK packet. If the port is not listening, the victim will send back RST/ACK packet. After that, the attacker will send ACK packet to complete the full connection but TCP half-connect, on the other hand, will employ RST packet to close the connection immediately.

For some systems, there is a bug in the TCP implementation that helps Stealth scanning works. Stealth or TCP FIN scanning plays a role of attack by sending packets with FIN flags to the opponent. A RST/ACK packet will be given back if the port at target host is closed. Otherwise, the attacker receives nothing from an opening port of the victim. Xmas Tree gives the target host with TCP packets a FIN/URG/PSH flag setting; but a NULL scanning sends packet without any flag setting. Both of them listen to RST/ACK packet to indicate a closed port and no acknowledgement packet to indicate a listening port.

3.2 Detection Assumption

Our system relies on the following assumptions:

- **Attack types**: TCP connect, TCP half-connect, Stealth, Xmas Tree and NULL scanning.
- **Attack behavior**: at the same time, there is the only one attack event pointing to the only one target host.
- **System location**: the system acts as the host-based and is embedded into the target host.
- **Input**: packets received by the target host
- **Network environment**: shared or switched based 10 Mbps Ethernet.

3.3 System architecture

The key idea of system is to scrutinize an extensive set of features that were extracted from network packets. The system employs threshold values to aid the conclusion. The architecture is mainly comprised of 2 parts: (1) Feature Selector (FS) and (2) Decision Engine (DE) as shown in Figure 2.

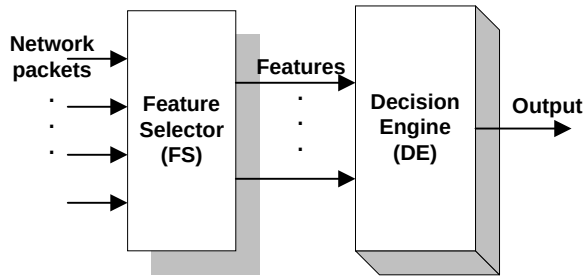

Figure 2. System architecture

3.3.1 Feature Selector

The main task of FS is to extract important parameters (features) and other information from packets. The significant features are such as source IP address, TCP flags, destination port number, packet time interval, and number of received packets, which have been prior defined in the Predefines Feature component. The structure of FS is shown in Figure 3.

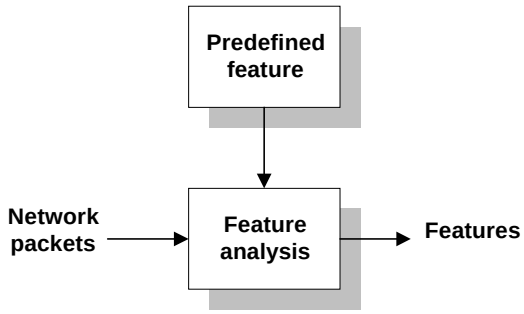

Figure 3. Feature Selector components

The Predefined Feature (K) is in the form of 5-tuples of parameters. Its characteristic can be defined as the system (F, S, P, ΔT, N) as follows.

$$K = (F, S, P, \Delta T, N)$$

Each parameter has the following meaning:

1. F = {URG, PSH, SYN, FIN, ACK, RST} is a set of flags marking on TCP header of an anomalous packet.
2. S is a source IP address of packet
3. P is the destination port number.
4. ΔT is an interval time of two consecutive packets.
5. N is the number of suspicious packets.

3.3.2 Decision Engine

Decision engine is an analysis part of the system. It analyzes events with a set of detection rules and identifies an intrusion. To identify intrusive patterns, sequence of packets must conform to several conditions. First, the packet must have the followings flag setting (F), namely, SYN, FIN, FIN/URG/PSH or no flag. Secondly, each packet has the same source address (S) and different destination port, P. Next, the time interval, ΔT, must less than α. Current value of α is 1 second, which is derived from experimental measurements of several well-known scanning tools. With this value, we found that there are no two consecutive packets establishing a legal request to different ports on the host under the same source address within 1 second. Finally, the number of all considered packets (N) must not be less than β. In this paper, β is equal to 20. This threshold, again, is obtained from the number of frequently exploitable ports in most scanning tools. Both ΔT and N may be adjustable depending on network environment. These above conditions are represented in a set of rules as the follows:

```
START: IF flag is a subset of set F THEN
          IF S are same THEN
             IF P are different THEN
                IF ΔT < α THEN
                   IF N = β THEN
                         Intrusion alerts
                   ELSE goto START
                ELSE stop detection
             ELSE stop detection
          ELSE goto START
       ELSE stop detection
```

We define 13 states of detection operation such as:

1. CLOSED: the idle state, no detection.
2. LISTEN: the initial state of detection operation.
3. FLAG_RCVD: received F of the 5-tuple K.

4. SAME_SRC: packet has the same source address.
5. DIFF_SRC: packet has the different source address.
6. SAME_PORT: packet has the same destination port.
7. DIFF_PORT: packet has different destination port.
8. TIME_LESS: interval time is less than α.
9. WAIT: waiting and saving state.
10. FINISH: all intrusive checks are done.
11. INTRUSION: intrusion indication, final decision.
12. STOP: the intrusive checks are stopped.
13. NORMAL: a normal behavior.

From above rules and states, we construct a detection state diagram as show in Figure 4.

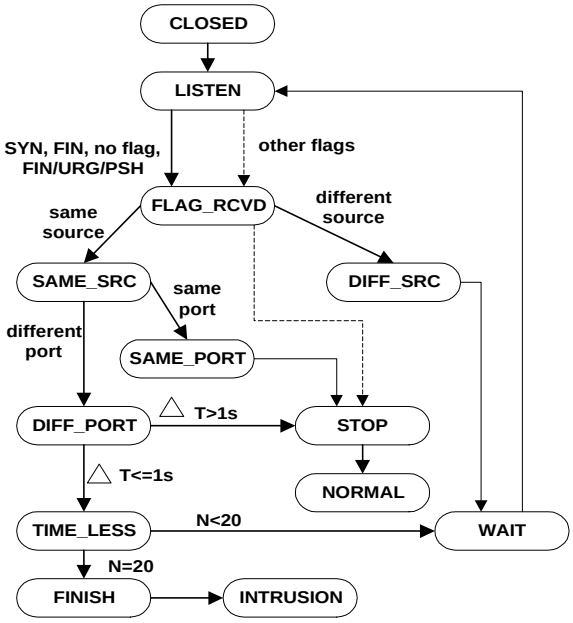

Figure 4. Detection state diagram

4. Testing

We conduct experimental tests of our detection program against several popular scanning tools such as nmap, SATAN, exscan and PortScanner under default scanning rate. The target system is Pentium II PC with 128 MB of memory running Linux Redhat 6.2. The overall system is shown in Figure 5.

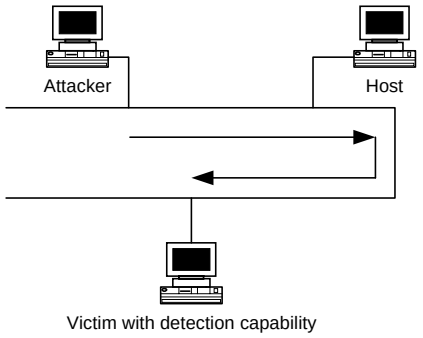

Figure 5. Testing environment

We arrange the tests in 3 different traffic loads: light load (20% of bandwidth), moderated load (50% of bandwidth), and heavy load (80% of bandwidth). The results in Table 2 show that successful scanning detection is achieved in all cases.

Table 2. Testing results

Scanning	20% BW	50% BW	80% BW
TCP connect	Detectable	Detectable	Detectable
Half-connect	Detectable	Detectable	Detectable
Stealth	Detectable	Detectable	Detectable
Xmas Tree	Detectable	Detectable	Detectable
NULL	Detectable	Detectable	Detectable

5. Conclusion and future works

We propose a rule-based port scanning detection and conduct several tests against well-known port scanning tools. Our main contributions are the design of systematic and pragmatic rules for successful detection of port scanning.

Our future work will focus on the refinement of decision techniques. The history profile of host's connections will be used to keep track for better detection. Profile filter is also one of the techniques to precisely select only relevant scanning packets.

6. Acknowledgement

We would like to thank Chalermkon Chongsanguan and Naruchit Chinawong for providing us the testing data as well as evaluating our approach.

References

[1] Biswanath Mukherjee, L. Todd Heberlein, and Karl N. Levitt, "Network Intrusion Detection", IEEE Network. 8(3):26-41, May/June 1994.

[2] Kumar, S, "Classification and Detection of Computer Intrusions", Ph.D. Thesis, Department of Computer Sciences, Purdue University, W.Lafayette, IN 1995.

[3] Debar H., Dacier M. and Wespi A., "Towards a Taxonomy of Intrusion-detection Systems", Computer Network 31 pp. 805-822, 1999.

[4] K.A. Jackson, D.H. DuBois and C.A. Stalling., "An Expert System Application for Network Intrusion Detection", Proceeding of the 14th National Computer Security Conference. pp.215-225, October 1991.

[5] Koral Ilgun, Richard A. Kemmerer and Phillip A. Porras, "State Transition Analysis: A Rule-Based Intrusion Detection Approach", IEEE transactions on software engineering. 21(3):181-199, Mar 1995.

[6] Fyodor, "The Art of Port Scanning", Phrack Magazine Volume 7, article 11, Issue 51 September,1997.