Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 23 (2011) 653 – 658

Procedia Engineering

www.elsevier.com/locate/procedia

The Study on Network Attacks based on Automaton Theory

SHANG Qing-wei， CAO Ke， WANG Feng

Xuzhou College of Industrial Technology, Xuzhou 221140, China

Abstract

Because the lack of a mature intrusion detection technology theory basis so far, this is very important to use mathematical method description and to study various complex attack behavior. The attack is corresponding to the program run about automata recognition characteristics of the attack. Because an attack is very complex, so it is difficult to use a unified automata model to test various computer network attacks. But automata model, which is used to describe the different network attacks are not independent exist mutual relationship.

Keywords-network attack; automata; network intrusion

1. INTRODUCTION

Network intrusion is defined as some behavior upsets the confidentiality, integrity, and availability and controllable network system [1]. Rapid development and application, network attack network techniques become more complicated. This is increasingly important characteristics of how to abstract and describe the network attack against the process, so that they will be able to effectively detection. Natural language can be used to describe attacks program. Although this method directly difficult to process natural language and computer. Tidwell used attack tree model of network intrusion program [2]. But his methods can not use words to describe the system state changes more effective. As we know, when the system it changes from one state to another country. These states represent a different meaning. They may be some normal and abnormal state of the state. But the system is a national corresponding only at any given moment, whether it is in normal or abnormal. The system operation of the limit state finally. This will ensure a system state is limited. So in the transition process can be described the system state and deterministic finite automata. When the system is under attack state will change, this procedure can be described by state directly transfer diagram. This makes against program is easy to understand.

2. DETERMINISTIC FINITE AUTOMATA

A deterministic finite automaton M is an automatic recognition device [3]. It consists of:

doi:10.1016/j.proeng.2011.11.2561

1. A finite set of states, often denoted Q.
2. A finite set of input symbols, often denoted $\sum$. It is usually called a condition set.
3. A transition function that takes a state and an input symbol as arguments and returns another state (or itself). The transition function is commonly denoted F. In the diagram representation of automata, F is represented by arcs between states and labels on the arcs. If $q \in Q$ is a state, and $s \in \sum$ is an input symbol, then F(q,s)=p($p \in Q$) and there is an arc labeled s from q to p.
4. A start state q0, $q0 \in Q$.
5. A set of final or accepted states Z, $Z \subset Q$ .A deterministic finite automaton M is abbreviated as DFA. It is often defined as a five-tuple:

$$M = (Q, \Sigma, F, q0, Z)$$

Where Q is a finite state set and it is not empty. One element of set Q presents a state of the system. ¬ consists of all conditions occurred in the system and it may represent the running of a program, the happening of an attack or an other event. $F : Q \times \sum \to Q$, it is a function with a single value, For $q \in Q$ and $s \in \sum$ there exists a state $p \in Q$, p is equal to F(q,s). q0 is only one start state of the system. $Z \subset Q$, it is a set of final or accepted states.

As described above, the state of a computer system can be described with deterministic finite automata. It is supposed that there are m states in set Q and there are n transition conditions in set $\sum$. Then there are m state nodes at most in the corresponding DFA. Each state node can be transferred to n neighbor nodes at most. The whole state transition procedure of a computer system can be described directly with the state transition diagram.

3. THE FORMAL DESCRIPTION OF SOME TYPICAL NETWORK ATTACKS

Network attacks are complicated generally. Their feature and mechanism may be very different. So it is difficult to use a uniform model to describe various different attacks. In order to find the common features of different network attacks deterministic finite automata are used to describe some typical attack procedures.

For a DFA model $M = (Q, \Sigma, F, q0, Z)$ which is corresponding to a different attack procedure, the system state Q may represent different meanings. It can be used to describe the states of hosts which are monitored, the states of processes, and so on. The condition set $\sum$ is called as the transition function. It is the cluster of functions and it consists of attack functions, communication functions, feature judgment functions, and so on. During the happening of attacks some functions are activated, the system transfers from one state to another state. For different attack procedures each component of DFA model may be different.

When the DFA model is used the caught data packet and log file are analyzed and audited, some feature parameters are got and they are used to judge whether the system is abnormal or there exist intrusion behaviors. With system running a state transition diagram is generated from the start state to the end state. What has happened to the system can be got by analyzing its end state. Some automaton models about typical network attacks (e.g. SYN-Flooding attack) are given as follows.

3.1. SYN-Flooding Attack

TCP is an oriented connection protocol in Internet architecture. When two nodes want to communicate each other, they set up their connection at first by three handshake procedure. It is supposed that host A want to access the resource of server B, then host A must set up the connection with server B before their exchanging information. The detail procedure is shown as Fig. 1. At first, host A send a connection request packet with SYN mark to server B. This packet consists of the initial serial number x of host A. After server B receives this request packet its state is transferred to SYN.RCVD and it allocates the corresponding resource for this connection. Then Server B sends the ACK packet with SYN/ACK mark to host A and this data packet consists of the initial serial number y of server B. It is obvious that the ACK serial number is x+1. At this moment the state of the system is called as the semi-connection state. After host A receives the SYN/ACK packet it sends the ACK packet to server B again. The ACK serial number in this packet is y+1. Server B receives the ACK packet and its state is transferred to "established". The connection is set up at this moment and host A can exchange information with server B[4-5].

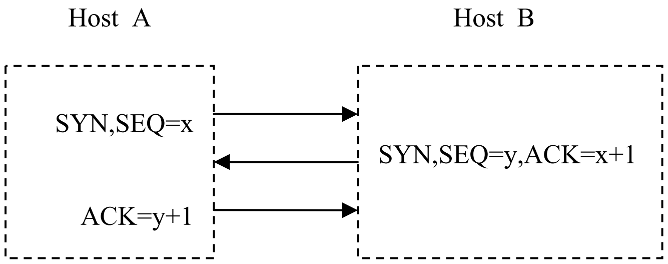

Figure 1. Setting up the connection between host A and server B

The procedure of setting up connection mentioned above is the normal situation for TCP protocol. But after server B sends SYN/ACK packet to host A, it maybe not receive the responsive packet from host A for a long time. Then server B has to wait for a moment. If such semi-connection exceeds a certain amount it is possible to use up all system resources(e.g. buffers) of server B which is used to set up the connection between server B and other nodes. Once the resource of server B is exhausted other normal connection requests for server B cannot be responded. Denial of Service(DoS) attack happens. This is the basic theorem of SYN-Flooding attack.

The detail procedure of SYN-Flooding attack is described as follows:

The intruder forges a non-existed host C or more hosts and it sends a large amount of the connection request to server B. Because the forged host doesn't exist in fact, for each connection request server B cannot receive any responsive information so that it has to wait for a long time. So a lot of requests with semi-connection state happen in a short time and the relevant resource of server B is used up quickly. In this case some normal connection requests will not be satisfied. This means that server B refuses to serve for any other normal request and DoS attack happens.

By means of a deterministic finite automaton the attack of SYN-Flooding is described as follows:

$$M = (Q, \Sigma, F, s, Z)$$

Where $q \in Q$ q = (Intruder-status, Server-status, System-status). Intruder-status is the state of intruder, Intruder-status $\in$ {listen, faked, SYN.SENT, ACK.SENT, failed, established}. Server-status is the state of the server, Server-status $\in$ {listen, SYN.RCVD, SYN-ACK.SENT, ACK.RCVD, blocked, established}. System-status represents whether the intrusion has happened or not, System-status $\in$ {false, true}, it represents some intrusions have happened when System-status is true. $\sum$ is a set of transition functions and it consists of attack functions, communication functions, testing functions and other functions. ┐ is defined as follows:

{ E0: fake( )
E1: Communication(s-host,d-host,SYN-ISN,0)

E2: Communication(s-host,d-host,SYN-ISN, ACK-ISN)
E3: Tcp_resource_used_out( ) }

E0 is used to forge a non-existed host randomly. E1 is used to send SYN request packet from source host: s-host to destination host: d-host, SYN-ISN is the sending serial number of source host. E2 is used to send SYN-ACK packet from the source host: s-host to the destination host: d-host, SYN-ISN is the sending serial number and ACK-ISN is the ACK serial number. E3 is used to judge whether the resource of the server about TCP connection is used up, if the resource is used up E3 returns true, or E3 returns false.

The state transition diagram which describes the attacking procedure of SYN-Flooding is shown as Fig. 2.

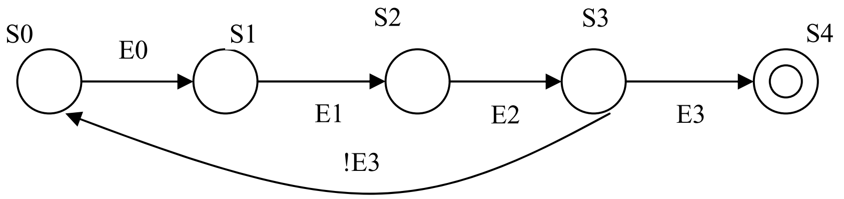

Figure 2. The state transition diagram of SYN-Flooding attacks

All states in Fig. 2 are described as follows individually:

S0=(listen, listen, false)
S1=(faked, listen, false)
S2=(SYN.SENT, SYN.RCVD, false)
S3=(failed, SYN-ACK.SENT, false)
S4=(listen, blocked, true)

Where S0 is the start state of the system. After the intruder forges a non-existed host the system enters S1 and the intruder is in the state: “faked”. Then the intruder try to set up the connection with server B and the system enters S2. The server gives its response as soon as it receives the connection request and the system enters S3. But the intruder is in the state “failed” because it forges a non-existed host and it cannot receive the SYN-ACK packet.

3.2. IP-spoofing Attack

If an intruder wants to hide its true identity or it try to utilize the privilege of the trusted host in order to attack other hosts it often fakes the IP address of other hosts. It is supposed that host A is a trusted host of server B. If the intruder wants to forge host A to communicate with server B it must steal the IP address of host A to spoof server B. This is called as IP-Spoofing attack[6-7].

IP-Spoofing attack is described in detail as follows:

1. The intruder makes host A blocked by DoS attack so that host A can not disturb attacks which will occur.

2. The intruder sends the connection request to server B at first and it guess the TCP serial number according to the responsive packet from server B.

3. The intruder uses the IP address of host A as its source address, then it sends SYN request packet to server B and try to set up the connection with server B.

4. Server B sends SYN-ACK packet to host A. But at this moment host A has been blocked and it can not receive SYN-ACK packet from server B.

5. The intruder forges host A again to send ACK packet to server B so that it sets up the connection with server B by three times handshakes.

The model M which is used to describe the procedure above is shown as follows:

$$M = (Q, \Sigma, F, s, Z)$$

Where $q \in Q$ q = (A-status, B-status, Intruder-status, System-status). A-status is the state of host A, A-status ∈ {listen, blocked, SYN.SENT, SYN-ACK.RCVD, ACK.SENT, failed, established}. B-status is the state of server B, B-status {listen, SYN.RCVD, SYN-ACK.SENT, ACK.RCVD, failed, established}. Intruder-status is the state of the intruder, Intruder-status ∈ {listen, faked-A ACK.SENT, SYN-ACK.RCVD, KNOWN-TCP-NO, failed, established}. System-status represents whether the intrusion has happened, System-status ∈ {false, true}. When System-status is true it represents that the intrusion has happened. $\sum$ is the set of the transition functions. It consists of attack functions, communication functions, serial number guessing function, and so on.

The set of attack functions are defined as:

{ A1: Land( )
A2: SYN_Flooding( )
A3: DoS( ) }

Where A1, A2 and A3 represents Land attack, SYN-Flooding attack and DoS attack respectively.

Communication function is defined as follows:

Communication(s-host, d-host, Syn-no, Ack-no)

Where s-host and d-host are the source IP address and the destination IP address respectively. Syn-no and Ack-no are SYN serial number and ACK serial number respectively.

The set of other functions are defined as follows:

{ E1:Communication(faked-A, B, Syn-no, 0)
E2:Communication(B, A, Syn-no, Ack-no)
E3:Communication(faked-A, B, Syn-no, Ack-no)
E4:Guess_tcp_packet_isn(B) }

Where “faked-A” in E1 and E3 represents that the intruder has succeeded in forging host A. E4 is used to guess the initial serial number of TCP packets of server B. If E4 is successful it will return “true”, or it will return “false”.

The state transition diagram of the automaton model to recognize IP-Spoofing attack is shown in Fig.3.

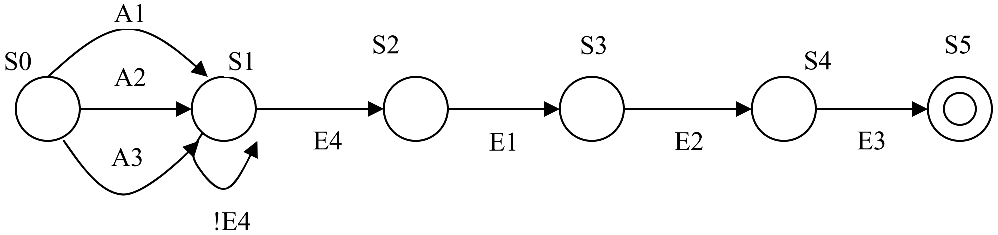

Figure 3. The state transition diagram of IP-Spoofing attack

Some states shown in Fig. 3 are defined as follows:

S0=(listen, listen, listen, false)
S1=(blocked, listen, listen , false)

S2=(blocked, listen, KNOWN-TCP-NO, false)
S3=(blocked, SYN.RCVD, SYN.SENT, false)
S4=(blocked, SYN-ACK.SENT, listen, false)
S5=(blocked, ACK.RCVD, ACK.SENT, true)

Where S0 is the start state. The system enters S1 after the intruder makes host A blocked by A1, A2 or A3. The intruder repeats to send the connection request to server B, after it guesses the serial number of TCP packet the system enters S2. Then the intruder forges host A to send SYN packets to server B for setting up the connection and the system enters S3. Server B sends the responsive packet with SYN and ACK to host A. But host A has been blocked and it cannot give any response to server B. Then the system enters S4. The intruder forges host A to send ACK packet to server B. After three handshakes have been finished the system enters S5. At this moment the intruder has set up the connection with server B and it forges host A to communicate with server B.

4. CONCLUSION

A model corresponding to a country can be another model or corresponding to a state transition function. We can construct various automata model intrusion activities, these models can flexibly to detect all kinds of complex network attack effect. So automata theory and its figure provide a compelling method described in the form of extension of the invasion of the program.

References

[1] Joseph S. Sherif et al.. Intrusion detection: the art and the practice. information Management and Computer Security, pp.175-186, Nov. 2003.

[2] T. Tidwell et al.. Modeling internet attacks. Proceedings of the 2001 IEEE workshop on information assurance and security, New York, pp.54-59,2001.

[3] John E. Hopcroft, Rajeev Motwanti and Jeffrey D. Ullman. Introduction to automata theory, languages and computation. Beijing: Qinghua University Press, 2002.

[4] Yan Xue-xiong, Wang Qing-xian et al.. The attack theory and prevention method of SYN_Flooding. Computer Application, Vol.20, pp.41-43, Aug. 2000.(in Chinese)

[5] Hu Wei-dong and Wang Wei-nong. A processing method of SYN/Flooding. Computer Engineering, pp.112-115. Aug. 2001.(in Chinese)

[6] Liu Xiang-hui, Yin Jian-ping et al.. Analyzing the security of the handshake procedure in TCP by deterministic finite automata. Computer Engineering and Science,Vol.24. pp.21-23, April 2002.(in Chinese)

[7] Chen Xiao-shu, Li Rong-hui et al.. Research on IP-Spoofing attack by the state analyzing method. The Journal of Huazhong Science and technology University, Vol.31, pp.3-5, May 2003.(in Chinese)