Research of Dynamic Information Flow Monitoring Based on Finite State Automaton

Yuehua Zhao & Yu Tao

School of Computer Science & Telecommunications Engineering

Jiangsu University, Zhenjiang 212013, China

Abstract

Because the current information-processing system security analysis of the dynamic monitoring is not mature enough, this paper will make use of finite state automata techniques to improve dynamic information flow monitoring methods, and design its monitors. This method is the use of finite state automata theory, information flow from the dynamic monitoring of proceeding through a secure stack to real-timely monitor the user's operation. This method can prevent high-density users through covert channel from leaking secrets to the low-density users and reach the purpose of Protection System confidentiality.

Keywords: Dynamic monitoring, Finite state automata, Information flow, System security

1. Introduction

Confidentiality is the most basic and common requirement of the information system. Access control is tried to prevent the information diffusion from checking and controlling the operations, but it can only prevent the information diffusion from the explicit way of access operations and not from the indirect way of non-access operations like transmission by influencing the system status, this indirect information diffusion method is usually referred to hidden information flow, This problem has been studied previously (Yan Li, 2008, p.51-57). Under certain conditions, it also is known as the covert channel. In order to overcome the weakness of access control, information analysis came into being.

Information flow analysis is divided into two kinds of static analysis and dynamic analysis, as the static analysis is only in the compilation stage of implementation, it cannot change real-timely with the corresponding situation. Compare to that, dynamic analysis is more user-centric and achieve on-line monitoring. End-user can customize the security level; dynamic analysis can monitor the system based on the level. End-user can modify the strategy according to their security requirement (Gurvan, 2007, p.9-30). Because of this, dynamic analysis has been paid more attention and developing.

Feng Qin et al (2006, p.135-148) proposed that every storage is set up a security level, and then checks the data's security level in the storage. This method can be applied to security analysis of the code without the need for its code source, but it cannot handle the complicated implicit information flow; Xu, Bhatkar and Sekar(2006, p.121-136) added the protection in the code to track the information flow, it is easier to understand than Qin's method, but still cannot handle the complicated implicit information flow; Nagatou and Watanabe(2006, p.577-584) established a kind of security automata as a tool to monitor system to detect and prevent unauthorized dissemination of information flow through covert channels, compared with the previous two methods this mechanism can judge the more complex covert channel, but the mechanism security automaton based on the input events cannot reflex the information flow in the system, and the mechanism has the potential to become a tool of covert channel.

In the real system, the system is described as the form of state transition; the system's current operation can be described as transferring state. Finite state machine which is used to describe the state of the system has been widely used. (Zi Xiao-Chao, 2008, p.1460-1466)

Therefore, using the finite state machine as a tool to improve the dynamic information flow monitoring method, this method can real-timely monitor covert channel and handle it safely, so that the confidentiality of information systems is guaranteed.

2. The finite state machine of the information system

2.1 Formal finite state machine

As the finite state machine of the static analysis has been studied (Zhang Yang, 2009, p.709-719), in this paper, the definition of the finite state machine is given directly.

The definition is FA (I, S, S_0, L, O, F):

(1) I defines all the way of the subject to access the object;

(2) S defines all the security context of the subject and object;

(3) $S_0 \subseteq S$ defines the initial state;

(4) L is the state transition function, $L:S \times I \rightarrow S$, L defines the type(domain) relationship between the legal transformation;

(5) O defines the finite state machine output which is not given in the literature(ZHANG Yang, 2009, p.709-719), we define O as $O \subseteq \{True, False\}$, if O is true, the transfer between states is legal, the direction of the information flow is decided by the read and write operation, if O is false, the transfer between states is not legal.

(6) $F \subseteq S$ defines system final state.

2.2 Improvement of finite state machine

The finite state machine can analyze the security strategies and determine whether they are fit for the requirement, but it needs all the states to make judgments which cannot fit the real-time requirements. When analyzing, the machine use the information flow chain of the user's operation to determine the user's action which cannot fit the real-time requirements. To solve this problem, we need to improve this machine:

Adding the stack Γ, the alphabet, $\Gamma = \{H, L\}$, an empty stack is recorded as ε. Stack is used to track security context to prevent the information flow from leaking.

Adding a flag to record the high-density user's illegal operation, range for the {H, L}, H represents dangerous operation of high-density user, L represents the normal operation or no operation.

Therefore, the definition of finite state machine is revised to FA (I, S, S_0, L, O, F, Γ, flag).

According to TCSEC requirement, the bandwidth of the covert channel above the 100 bits / s must be eliminated; the bandwidth below 1 bit / s can be accepted and bandwidth between 1 bit / s and 100 bits / s will be based on the actual situation to decide whether the auditors done to eliminate processing. According to these provisions, To determine whether a low-density user to get information through a cover channel whose bandwidth is above the specified value or not need to take the operations before into account. Because of the advanced features of the stack, it can be used to track the user to enhance the expression of the finite state machine capacity.

The modified finite state machine can show the transfer of system security status and determine whether the operation is dangerous or not based on the modified finite state machine output set O.

In the modified finite state machine, I define the operations which the subject takes; S defines the object's current security context; S_0 define initialized security context.

3. Finite state machine based on dynamic information flow control

3.1 System model

The system in which users run applications is called the target system. In the target system, two users are considered as S_H and S_L. S_H is a high-density user who can access the confidential information; S_L is a low-density user who cannot access the confidential information. S_H performs some actions to cause a series transfer in the target system, S_L wants to observe the behavior of S_H through the way of reading operation to output the result. If no matter what actions the S_L performs, S_L can observe nothing, In other words, there isn't a cover channel between S_H and S_L.

Therefore, In order to monitor the information flow, the above finite state machine is used in the target system to form a security automaton. It can receive the state transfer, check the legality of the information flow and determine whether the low-density user can observe the high-density user or not; the machine can guide the system based on the result.

3.2 Automatic machine monitoring system security principles

According to the current study, high-density S_H sends information to low-density S_L by changing O_i's properties A_i through the encoding in advance. S_L can observe the changes of object O_i, made by S_H to get the information from S_H (Wang Chang-da, 2006, p.1488-1492), Fig 1 shows the principles of the cover channel.

(Insert Fig 1)

The security automatic machine mechanism: The security automatic machine is initialized.

Stack Γ is emptied. When the high-density S_H operates on the object O, the automatic machine judges whether or

not there is information flow into the object O. If there are influences on the S_L made by S_H, then the security automatic machine sets the flag with H. When the low-density S_L operates on the object O, the security automatic machine pushes flag into the stack and set flag with L. the security automatic machine has a timer to trigger the checking mechanism, when the timer is trigged, the mechanism give the number of the stack values, if the number is over the rating, the mechanism silence the S_L because of the cover channel is between the S_H and S_L. if the number isn't over the rating, the mechanism empties the stack. The timer and the rating can be changed by the security requirement of the system. The higher of the security level is, the shorter of the timer. Through the mechanism, the security of the system is ensured.

According to the principles of the cover channel, the high-density user can only send the information to the low-density user trough the object O and only one bit information a time. Then in order to explain how to use the security automatic machine mechanism to protect the system from harm, this paper gives a example to show the mechanism in the Table 1

There are two functions in the table, readlike and writelike. If the operation is similar like read operation, the function readlike return true, Otherwise return false; If the operation is similar like write operation, the function writelike return true, Otherwise return false;

(Insert Table 1)

3.3 Realization of the security automatic machine mechanism

The security automatic machine mechanism can be realization to a security monitor system by the support of the SELinux system functions. When the system is loaded, the security monitor system can be loaded to monitor information flow to protect the system from the cover channel harm at the place shown in Fig 2.

(Insert Fig 2)

The main programs of the security monitor system:

The Finite state machine initial:

```
FA_Load()
{
  //All security context in the SELinux
  State<StringArray> S = new State< StringArray >;
  S = {(user_u:object_r:user_home_t:Kmls), (user_u:object_r:user_sys_t:Kmls)
(user_root:object_r:root _t:Kmls)……(user_root:object_r:sys _t:Kmls)};
  //initial context
  State S0 = (user_u:object_r:user_home_t:Kmls);
  //Operations on the object
  Operation<String> I = {Create,Delete,Open……,Lock};
  //Output of the FA
  Boolean Output = false;
};
```

The event trigged by the timer:

The event can check the cover channel by counting the number of the stack values.

```
private static void Onesecond_Elapsed(object source, ElapsedEventArgs e)
{
  if (mystack.Count >= 100)
    //if the number of the stack values is over 100, then there is a cover channel in the system, the
  security monitor system must handle it.
  {
    if (!Delayover)
  {
```

```
        Console.WriteLine("Cover Channel is found!!!!!!Bandwidth is {0}", mystack.Count);//Print
the value of the cover channel bandwidth
        }
        ……
        Delaytime.Interval = Convert.ToInt32(mystack.Count / bandwidth) * 1000;
        //Caculate the delay time based on the cover channel bandwidth
      }
      else
      {
          …….
        mystack.Clear();
        //If one second has passed and the number is not over 100, the security monitor system
 empties the stack,
        }
}
```

The programme above show the way to realize the security monitor system and the security monitor system how to monitor the cover channel , caculate the bandwidth and silence the low-density user based on the bandwidth.

4. Conclusions

In this paper, finite state machine is used to improve the dynamic information flow control method. Compared with the methods before, this method is user-centred,can change with the security level changed and realtimely monitor the information flow to protect the system from the cover channel.

However the security monitor system based on the security automatic machine mechanism can only deal with two security levels and some simple operations. In the futher, this mechanism should be improved to do a good job to face the muti-level system and complicated operations.

References

Feng Qin, Wang Cheng, LI Zhen-min, Ho-seop Kim, ZHOU Yuan-yuan & Wu You-feng. (2006). LIFT: A Low-Overhead Practical Information Flow Tracking System for Detecting Security Attacks, Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture, p.135-148, December 09-13, 2006.

Gurvan Le Guernic. (2007). Confidentiality Enforcement Using Dynamic Information Flow Analyses. Kansas State University, p.9-30, 2007.

Naoyuki Nagatou, Takuo Watanabe. (2006). Run-Time Detection of Covert Channels. Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06), p.577-584, April 20-22, 2006.

WANG Chang-da, JU Shi-guang. (2006), Simulation Analysis of Covert Channels. *Journal of System Simulation*, p.1488-1492,August 2006.

Wei Xu, Sandeep, Bhatkar & R.Sekar. (2006). Taint enhanced policy enforcement: A practical approach to defeat a wide range of attacks. In Proceedings of the USENIX Security Symposium. In Proceedings of the USENIX Security Symposium, p.121-136, August 2006.

YAN Li, JU Shi-guang, & WANG Chang-da. (2008). Real-time monitoring and controlling to secure information flow. *Journal on Comunication,* p.51-57, Octobor, 2008.

ZHANG Yang. (2009). A Information-Flow-Based Verification Solution with Security Sensitivity to Check Security Policy of SELinux. *Chinese Journal of Computers,* p.709-719, April, 2009.

ZI Xiao-Chao, YAO Li-Hong, LI Lan. (2008). A State-Based Approach to Information Flow Analysis. *Chinese Journal of Computers,* p.1460-1466, August, 2008.

Table 1. the example of the security automatic machine mechanism works:

Subject	Object	Operation	Stack	flag	Timer	Return value	Result
S_H	O	Read(readlike(p)==1)	ε	L	Start	True	Allow
S_L	O	Write(Writelike(p)==1)	L	L	Less than 1s,waiting	True	Allow
⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮
S_L	O	Write(Writelike(p)==1)	L,...,L	L	Less than 1s,waiting	True	Allow
S_L	O	Write(Writelike(p)==1)	L,...,L,L	L	Less than 1s,waiting	True	Allow
S_H	O	Lock(writelike(p)==1)	L,...,L,L	H	Less than 1s,waiting	True	Allow
S_L	O	Write(Writelike(p)==1)	L,...,L,L, H	L	Less than 1s,waiting	True	Allow
			L,...,L,L, H		One second is passed	False	Deny
			ε		Restart	True	Allow

Note: the timer can be changed based on the security requirement

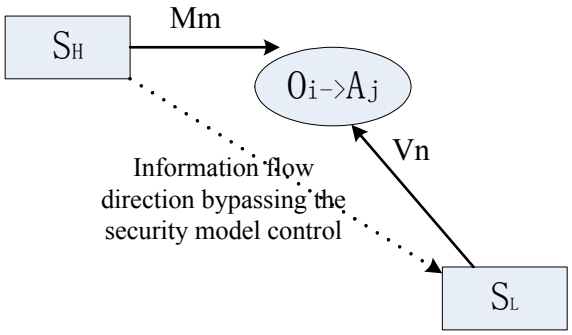

Figure 1. the principles of the cover channel

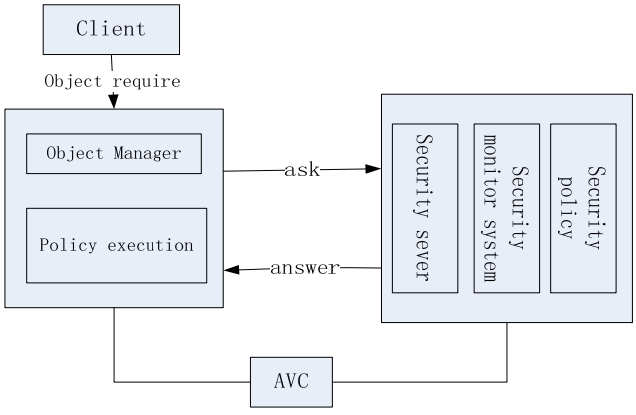

Figure 2. the place of the security monitor system