A Large-scale Trojans Control Model Based on Layered and P2P Structure

Qindong Sun
School of computer science and Engineering, Xi'an University of Technology, Xi'an, China
Shaanxi Key Laboratory of Network Computing and Security, Xi'an University of Technology, Xi'an, China
Email: sqd@xaut.edu.cn

Xiuwen Sun , Nan Wang, Qian Wang
School of computer science and Engineering, Xi'an University of Technology, Xi'an, China

***Abstract*—In order to achieve large-scale Trojan control in an effective way, a Trojan control model based on layered and P2P structure has been proposed in this paper. According to our model a hundred thousand magnitude Trojan server could be under control. This model could adjust the number of Trojan dynamically by revising the layer number. In addition, the load balancing of severs have been realized by peer-to-peer network which could control large scale Trojan within an acceptable range of system resources consumption. In the end, a prototype system has been established to prove the model validation. The experiment results have shown that the large-scale Trojan control model is effective and powerful.**

***Index Terms*—Network security, Trojan control model, Layered structure, Peer-to-peer network**

I. INTRODUCTION

Along with the fast development of information technology and computer science, Security-related researches have become more and more important in various fields [1][2]. Computer virus also belongs to the security research fields and has great value because of its special influence to computer users.

Trojan is a kind of typical computer virus which is backdoor that can be used by hackers to steal personal information from other users and even control their computers remotely. It usually spread in various ways and defraud users to execute for stealing passwords and any other kinds of information[3]. Trojan usually contains two parts that the server and the client. The server implanted in the target host with high concealment and established connections to the client for receiving all kinds of control commands and sending the execution results back.

In recent years, the research of Trojan is focused on improving the concealing of the server and the survivability of the individual[4]. For national information security, the monitor targets are large-scale networks in war of network and information, what's more, focusing on the effectively control[5]. However, with the development of the technology of Trojan, the control models are mostly still stuck on the star topology model based on C/S structure, which is difficult to meet the needs of current and future applications.

In this paper, we have proposed a layered and peer-to-peer model by learning mature network services model and combining the technical characteristics of star topology model with peer-to-peer network architecture. The number of Trojan in this implementation can be a few to hundreds of thousands. In addition, it has realized load balancing of connection to the large-scale servers and efficient storage and retrieval of the monitoring data.

II. COMMON TROJAN CONTROL MODELS

From the generation of Trojan until now, it has experienced the simplest one-to-one C/S structure, to the central server-based star topology model, and then the model that aiding of the other applications, such as Http Services, Web Mail[6,7], Web Services and any other various network service model[8]. The introduction of common Trojan models as below.

Generally, the server listens on a specific port and the client sends request of active connection in this structure as shown in Fig. 1. The processing which implemented in this model is a one-by-one control with narrow scope of applications.

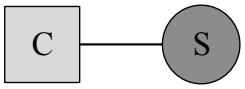


Figure 1 simple C/S model

Star topology model is usually listening by the client and initiating connection by the server. Fig.2 shows the structure of the star topology model. It's the most widely used model, such as gray pigeon, gh0st. For this model, the connection between the client and each server using standard C/S structure, establishing 1: N's connection and control.

*Corresponding author.
Email address: sqd@xaut.edu.cn (Qindong Sun)

doi:10.4304/jsw.9.6.1423-1427

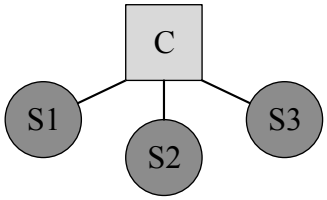


Figure 2 Star topology model

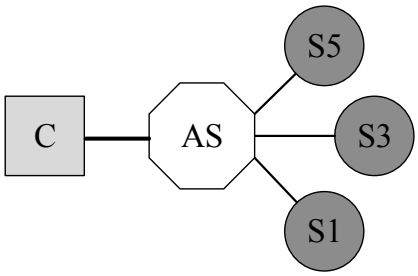


Figure 3 Model based on AS

As shown in Fig.3, the model based on application service can be seen an extended form of the star topology model. The client and server established connections with AS (Application Server) through a variety of application protocols, such as HTTP, POP. Then, the client sends command to the AS and the server detects and receives it in timing. At last, the results of execution and data of monitoring are sent back to the AS.

The hybrid model of star topology and peer-to-peer network has been applied in many TCP / IP applications, such as SMTP, WWW services. Applied it to the remote control system, each host can be regarded as the server and the client were the ISP's servers. As shown in Fig.4, it's star topology between the server and client and peer-to-peer architecture between the client and client.

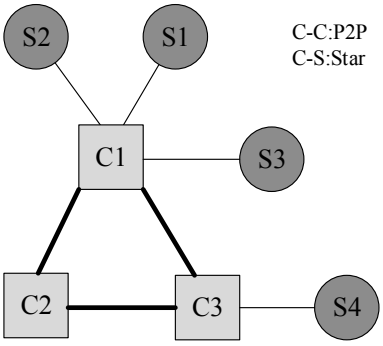


Figure 4 Hybrid model of star topology and P2P network

Through the introduction of the Trojan control models, we can realize that the star topology model and model based on AS would achieve 1: N's control, but the number of server cannot meet to the demand of large-scale network applications for the shortage of resource in the client. Especially, it is difficult to conduct real-time monitoring in the model based AS, because of the intermediary AS between the client and the server. The hybrid model of star topology and peer-to-peer network can support real-time monitoring for large-scale server, but further researches are needed for the application in this field. For example, the problems of node discovery cross the client and efficiency of monitoring.

III. LAYERED AND P2P MODEL

We proposed a layered and P2P model by researching and analyzing of existing Trojan model. It has solved the problem of large number of server and provided a solution on efficiently control.

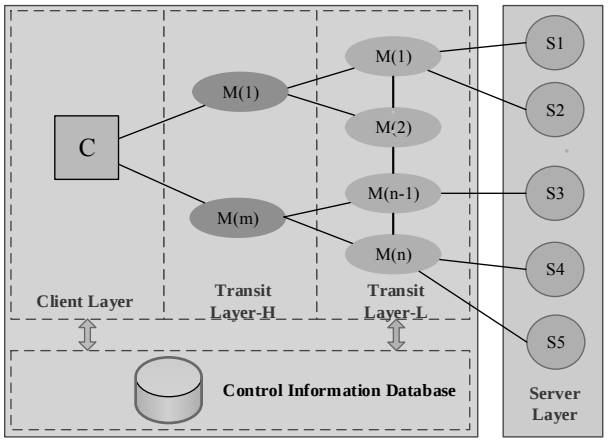


Figure 5 Layered and P2P model

A. Model Introduction

In general, as shown in Fig.5, this model is divided into three layers that server layer, transit layer and client layer. They are connected by star topology model between layers. It can be extended from 0 to N (N> 0) according to actual needs in transit layer and the H-layer and L-layer are respectively located in the top and bottom layer of transit layer. The control information database, which connected with L-layer and client layer, mainly stored three categories information: list of commands, results of execution list, and list of server hosts. More details and main functions are described as follows.

(1)The server layer consists of one or more server host which is the actual control object and distribute in the network that can be reached.

(2)The transit layer is composed by few layers of control module and each module in a layer is connected to about 500 modules in the next lower layer, but except L-layer. It forwards commands and results of the execution between the upper and lower layers. The architecture of L-layer is unstructured peer-to-peer network that storing and sharing monitoring data and realizing load balancing of connection. Each L-layer control module is responsible for managing about 5000 hosts.

(3) The client layer is the true sense of control entity that sending commands and receiving the results from the server layer.

B. Working Principle and Control Logic

There are two kinds of commands according to whether need immediate execution. The commands of need immediate execution are issued and returned results synchronized in one connection that usually for single host. The others, which contain commands on group, are issued and returned results asynchronously and may not be in the same connection.

a. The basic logic of server layer and L-layer

(1)Host in the server layer establish connection with the specific module in L-layer according to the default address and request the control module address in working time when predefined events are triggered.

(2)After queried its conditions of loading, the module would maintain the current connection while it hasn't exceed the threshold. Otherwise, it would obtain an appropriate address back to the host after querying the current condition of system.

(3)The host would maintain the current connection or re-establish a new one based on the result of the request. After that, it upload its identification which assigned by the predefined module in L-layer in its first connection. Then, L-layer saved basic information to the control information database and synchronized the list of users to upper layer.

(4)L-layer checked whether there are group commands or command for this host which hasn't been processed in the command buffer list, and sending these commands to the server layer in turn.

(5)Server layer upload command execution result lists. L-layer accepted and saved them to database, and then send summary of the results to client layer. If continuous output is needed, for example, keyboard monitoring, L-layer forward resulting data to client layer.

b. The logic of issuing group command

(1)The client issues group commands to H-layer. At last, they are forwarded to the L-layer through the transmit layers in turn.

(2)The L-layer receives and stores them in its buffer list and begins waiting for the connection of hosts. It would send them directly if they had connected.

(3)The host in server layer executes commands and returns the results.

c. The logic of issuing server command that just for single host

(1)The client gets the address of an L-layer module according to the current connection status for the online host which is connected with it and the algorithm 1 for the offline host. After that, the client sends commands to L-layer directly over any other layers.

(2)The L-layer receives and stores commands in its buffer list and begins waiting for the connection of hosts. It would send them directly if they had connected

(3)The hosts execute commands and return the results of them.

d. Returning results from server layer

As to the commands that results returned non-immediately

(1) While L-layer hasconnected with the server host, it sends command lists and accepts result lists. Then, storing result lists in database with serial number of commands as their index.

(2) L-layer uploads the summary of results to client layer through layer by layer.The client layer could extract detailed results from database and present to the administrator when they are needed.

As to the commands that results returned immediately

(1) L-layer sends commands which are needed to return immediately after have connected with server host.

(2) L-layer uploads and presents the results to the client layer over any other transit layers.

(3) Based on these two steps, sending the remaining lists of command while all the results have returned.

C. Model Features

(1) Flexibility: the number of server hosts that the model carrying can be widely adapt by adjusting the transit layers' number.

(2) Real-time: It has realized real-time control between the client and the server by forwarding commands and data through transit layer. Furthermore, even in the case of carrying a large number of server hosts, the transit layers' number won't be too large and having little influence on forwarding of commands and data, because of the transit layers' number changing exponentially.

(3) Efficiency: L-layer can take reasonable allocation for new connections and take full advantage of system resources, such as network bandwidth, CPU usage and storage space.

(4) Security: The releasing of the control commands and accessing to the monitoring results can be operated in the client layer only. In addition, commands and data which are forwarded in the model can be encrypted. These have limited illegal access of unauthorized users significantly and improved the security of model.

IV. IMPLEMENT AND VERIFYING OF PROTOTYPE SYSTEM

In this section, we design the communication protocol and implementation of the transit layer for a prototype system in accordance with all the above description. The next, we will elaborate the key technical problems in this prototype system.

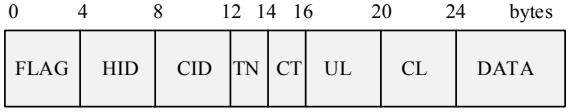


Figure 6 The communication protocol of prototype system

TABLE I.
DESCRIPTION OF COMMUNICATION PROTOCOL

Fields	Length (byte)	Description
FLAG	4	Marking of protocol header
HID	4	Host ID. Identification of server host when issuing commands. Number 0 indicate group commands and the others are divided by region of L-layer.
CID	4	Command ID. Identification of commands that issued by the client and associated with execution results.
TN	2	Token. Types of commands.
CT	2	Encryption types.
UL	4	The Length of uncompressed data.
CL	4	The Length of compressed data
DATA	0-4096	Transmission data between the client and server.

A. Communication Protocol

The design of communication protocol is shown in Fig.6 according to the logic of the model. The length of each fields in the protocol are described in Table 1 as below.

B. Realization of L-layer

a. The control module framework

Transit layer is the most important layer of this model and the L-layer, which shown in Fig.7, is the most complicated part in it. Now, we focus on the features and implementation of each module in L-layer.

Directly control module: Realized by IOCP model, it responsible for communication of the connected server that issuing commands and receiving the results.

Load balancing module: Building a peer-to-peer network that realizing load balancing of new connection from the server hosts.

Local operating agency: Communicating with the upper and lower control modules, forwarding and saving commands and results of execution.

Local operator interface: Communicating with local operating agency and realizing user interface that showing the server lists and results of execution. It can set and adjust the control module parameters partly.

User management module: Synchronizing user list to upper layer.

Database interface: Storing user information, commands list, results of execution.

b. The load balancing of connected server

In a computer network, the load balancing technology is used between two or more hosts, allocating their workload to ensure system reliability and data availability under the premise, achieving efficient use of resources and improving the performance of overall system[9]. The distribution system and some other large scale systems do need a rational load balancing strategy for well operation of the system [10].

For the convenience of system extension, L-layer use unstructured peer-to-peer network in this control model that each node only saving the information of local and several neighbor nodes. Thus, it has realized load balancing through flooding searching. Due to the restrictions of host resource, each control module set the upper limit of the controllable server host number as *MAX* according to Eq. (1) during initializing and calculate the threshold value $T=MAX*0.8$. The control module call the load balancing module for obtaining address of lighter control module host only if $m>T$.

$$MAX = \frac{f*0.5 + m*0.3 + b*0.2}{\rho} * 5000 \quad (1)$$

In Eq. (1), f (GHz) is frequency of CPU and m (GB) is capacity of memory, b (100MB) is the bandwidth of the connection between the client and L-layer, ρ is the adjustment factor that $\rho = 2.9$ in the prototype system.

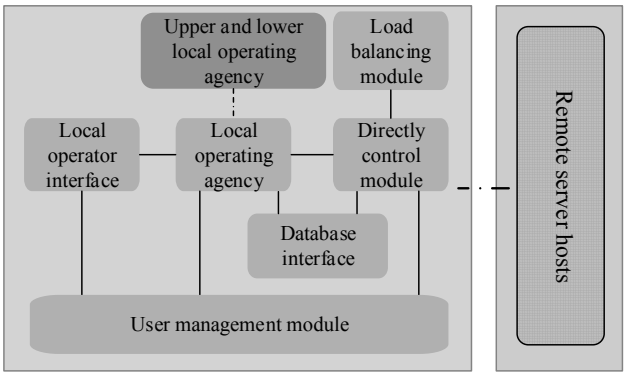
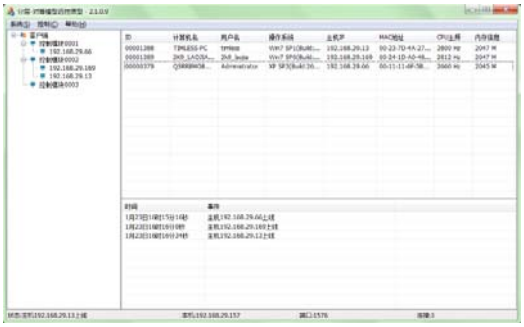


Figure 7 The framework of L-layer control module

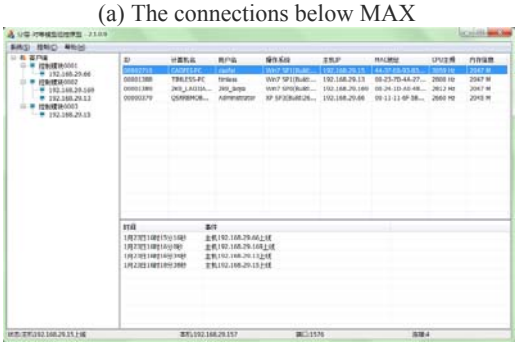
C. Verifying of Prototype System

For the limitations of experimental conditions, transit layer deploy one and only three control modules in the prototype system which set the maximum number MAX to 5. In the experiment, we selected six server hosts wherein one is connected to control module 1 and the other five are connected to the control module 2. In Fig.8 (a) and (b), it's showing the load balancing situation in L-layer. The last host has set to connect with control module 2, but, it was assigned to control module 3 as a lighter module with the action of the load balancing module.

In addition, we have applied a performance test about control module by multiple threads technique. Fig.9 shows the memory usage of the control module when the number of connections has increased from 0 to 6534. Fig.10 shows the situation of the control module when the number of connections has achieved to 6534.



(a) The connections below MAX



(b) The connections above MAX

Figure 8The load balance in prototype system of layered and P2P model

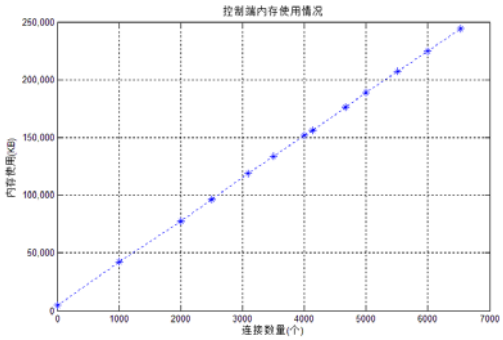


Figure 9 The relationship between Memory usage and connections.

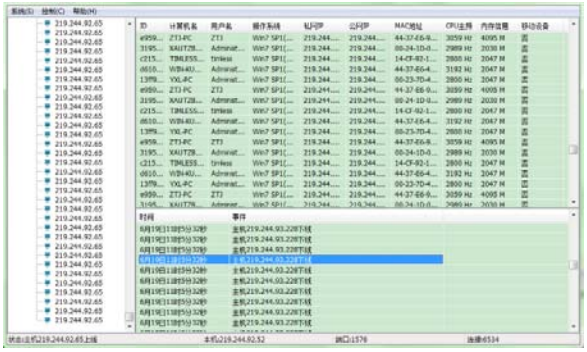


Figure 10 The status of control module with large connections.

The results have shown the number of connections of control module could be achieved the design request in theory. Therefore, the model could control large-scale Trojan in an effective way.

V. CONCLUSIONS.

In this paper a Trojan control model based on layered and P2P model has been proposed for supporting large-scale server hosts as the existing Trojan control models are most based on the basic topology structure. In order to research the model validation a prototype system has been established to verify the effectiveness of the large-scale Trojans control model based on layered and P2P structure. The results have shown that with our model large scale Trojan could be controlled in an effective way as the system resources consumption is under a acceptable range.

However, there are still some improvements needed to be done about the model. Concerning on the future works, we are going to do further researching studies in the following two points:

(1) Using structured peer-to-peer network for L-layer and deploying the control information database on the hosts that in L-layer to achieve data sharing and searching.

(2) Studying an adaptive load balancing algorithm based on connection regulation between server and client.

ACKNOWLEDGMENT

The research presented in this paper is supported partly by the Innovation Support Program of Xi'an Science Technology Bureau (Grant No.: CX12178(4)), the International Cooperation Project of Shaanxi Province (No.: 2013KW01-01), The Key Laboratories Development Program of Shaanxi Province Education Department (No.: 13JS085)and the National Natural Science Foundation of China (No.: 61172124).

REFERENCES

[1] Lixia Xie, Xiao Zhang, Jiyong Zhang. "Network Security Risk Assessment Based on Attack Graph". Journal of Computers, Vol 8, No 9, pp. 2339-2347, 2013.

[2] Jiyi Wu, Qianli Shen, Tong Wang. "Recent Advances in Cloud Security". Journal of Computers, Vol 6, No 10, pp. 2156-2163, 2011.

[3] Wikipedia.Trojan Horse. http://en.wikipedia.org/wiki/Trojan_Horse. Oct,2012.

[4] HU Bo, CAO Jiu-xin, SUN Xuesheng, YAO Yi, LIU Yong-sheng. "Research on self-adaptive Trojan horse model based on function atomization". Computer Engineering and Design, Vol.31,pp. 2683-2686, 2010.

[5] Kan Jianguai, Mao Xinjun. "The Model and Technology of Three-Levels Trojan Horse Based on Software Agent". Hunan, China: Graduate School of National University of Defense Technology,2011.

[6] WANG Juan, GUO Yong-chong, WANG Qiang. "New Trojan Horse Communication Model Based on WebMail System".Computer Engineering, Vol.34, pp. 157-159, 2008.

[7] Ji Wang, Yong Fang, Liang Liu."Research on Trojan Communication Model Based on IRC and Webmail System". 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE, pp. 682-685,2012.

[8] Shieh W G, Wang J M, Horng W B. "Secure remote control model for information appliances". Intelligence and Security Informatics, 2008. ISI 2008. IEEE International Conference on. IEEE, pp. 222-224, 2008.

[9] Ji G, Liu X, Zhang K, et al. "Design and implementation of remote model predictive control system". Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on. IEEE, pp. 19-23, 2011.

[10] Adeela Bashir, Sajjad A. Madani. "Task Partitioning and Load Balancing Strategy for Matrix Applications on Distributed System". Journal of Computers, Vol 8, No 3, pp. 576-584, 2013