Proxy-based Security Audit System for Remote Desktop Access

Shi-hai Huang, Chuang Lin, An’an Luo, Zhen Chen, Xin Jiang, Kai Wang, Hui Zhang, Xue-hai Peng
Department of Computer Science and Technology
Tsinghua University
Beijing, P.R.China
{hsh07,laa,jiangx05,zhanghui04}@mails.tsinghua.edu.cn, {clin,zhenchen,xpeng}@tsinghua.edu.cn

Abstract—Remote desktop access is commonly used to remotely access a host in enterprise networks; however it also brings in security problems in supervision and auditing. In this paper, a novel proxy-based security audit system is designed and implemented in order to ensure security supervising and auditing remote desktop access. Our system effectively monitors all the accessing sessions of RDP, VNC, and X-window, and provides replay function by recording all the graphics operations from users. Our performance test result shows that for most of small business, just one proxy server is enough to handle all the routine auditing workload of RDP sessions.

Keywords- RDP; VNC; X-window; Security; Proxy; Audit

I. INTRODUCTION

Nowadays, network attacks happen pervasively in Internet and those attacks, which usually cause serious damage, are always from the inside of enterprise networks. In 2008, the Computer Security Institute's survey [1] shows that the insider abuse and unauthorized access to systems were almost the top two categories of highest incidence. Therefore, it is significant and necessary to reinforce internal network security management. As an important component of network security system, security audit system helps administrators not only analyze and check users’ operations but also trace back and find the starting point of security events by recording all the sessions and events of some key information systems. Most of current security audit systems have strong monitoring and auditing ability for text-based network protocol, however, for graphics remoting protocols monitoring, e.g. remote desktop access, can easily evade the supervisor, causing unexpected results for security audit system.

Remote desktop access is commonly used in enterprise networks, which allows remote host’s graphical displays to be virtualized and served across a network to a client, and client can access the data and application in remote host just like local desktop. It is also a popular solution for thin-client computing. Currently, there are mainly three kinds of technologies for remote desktop access, which are: Remote Desktop Protocol (RDP) [2] used in Windows, Virtual Network Computing (VNC) [3] which is open source, and X-window [4] used in Linux/Unix operation system.

Remote desktop access is the most convenient way to telecommute and remote maintenance, but it also brings in security problems in supervision and auditing. Because of non-transparency among RDP, VNC, X-window, it often brings in security problems of resource abuse and breach of confidence, which are difficult to ensure security supervision and content inspection from external security tools. It is necessary to adopt effective mechanism to monitor and audit towards remote graphics operations, especially for those crucial servers with key data. In this paper, a novel proxy-based security audit system is designed and implemented in order to ensure security supervising and auditing for remote desktop access. Our system monitors all the accessing sessions of RDP, VNC, and X-window by using proxy technology, and provides replay function by recording all the graphics operations from end users. And our performance test results show that for most of small business, only one proxy server is enough to handle the routine auditing workload of RDP sessions.

The remainder of the paper is organized as follows. In section II, we introduce related background. In Section III, we propose an overview of system design. Section IV describes detailed technology and system implementation. Section V discusses the performance test results. Finally, we conclude the paper in Section VI.

II. BACKGROUND

There are two kinds of audit systems according to the means of audit information collection, i.e., Host-Based Audit (HBA) and Network-Based Audit (NBA).

HBA audits user behavior by collecting log files of hosts, which makes audit information more comprehensive and is little influenced by encryption protocol, but the weakness of HBA is hard to deploy, as well as causing performance decrease when running an agent at the target host.

NBA audits user behavior by collecting session information from network traffic. It can be deployed easily as a middle-box and is hard to be tampered with, as well as lessen cost in performance. NBA audit system generally collects log data from network devices, such as router, switch, firewall and IDS, where data may get lost. So some audit system with higher security level records the whole information of all the sessions, and is able to replay all the session records.

In order to simplify deployment and reinforce security supervision, security audit system for remote desktop access usually adopts NBA audit mode to record the whole session’s
information. So the key challenge is how to acquire and playback all the session records. Usually it can be done through by-pass monitor to collect network traffic of sessions; unfortunately, RDP protocol is an encryption protocol, and is hard to restore content directly from the traffic.

One solution to overcome this problem is to introduce proxy-based technology, which establishes connection between remote user and proxy server, and connects to target server through the proxy server. The proxy server is used to implement protocol conversion and packet forwarding between user and target server. The security audit system presented in our paper is based on proxy technology, and also strengthens authentication management, access control and session audit.

III. SYSTEM DESIGN

A. System Components

The security audit system presented in our paper includes two subsystems: proxy server and audit client. Fig. 1 shows a typical security audit system deployment.

B. Module Division

Security audit system consists of totally six function modules, which are: Configuration and Management module, RDP Proxy module, VNC Proxy module, XWIN Proxy module, Monitor & Audit module, and RDP playback module. Fig. 2 shows all the components of the whole system.

1) Configuration and Management module: By using Apache to build a WEB server, system provides interfaces for configuration and management of the proxy server. The interfaces include the following aspects: proxy server maintenance interface, configure information interface about registration and authorization of users, monitor & audit invoking interface, and user logon interface and so on.

2) RDP Proxy module: This module is simulated as RDP servers, which accepts requests from RDP clients, and does authentication and authorization of the RDP requests. After that, this module exchanges packets between RDP clients and RDP servers and also keeps record for session content.

3) VNC Proxy module: In order to make a RDP client visit a VNC server, this module plays a role of proxy between the RDP client and VNC server, converting protocols between RDP protocol and VNC protocol.

4) XWIN Proxy module: This module is used to perform a RDP client accessing X-window server. XWIN Proxy module accepts X-window logon requests from a specified remote computer with XDMCP(X Display Manager Control Protocol), and provides X-window service by invoking local VNC Server.

5) Monitor and Audit module: The Monitor and Audit module monitors current sessions and provides interfaces to query history logs and data statistics. Moreover, it can audit sessions by providing function of recording sessions, which can be replayed by RDP playback module.

6) RDP Playback module: This module running on auditing client, can replay the RDP session, which enables us to re-check the entire operation process.

IV. IMPLEMENTATION

A. RDP Proxy

The Windows RDP protocol developed by Microsoft Inc. is an extension protocol based on the T-120 [5] family of protocol standards, which is used to connect to Microsoft’s Windows Terminal Services. The implement of RDP technology is not publicly available, but the main technical details have been discovered by reverse engineering [6] and hacking/cracking technology [7]. As a protocol of TCP/IP application layer, RDP can be divided into four independent layers which from bottom...
to top are ISO [8], MCS, Secure, RDP layer. ISO and MCS (Multipoint Communication Service) layer implement major network communication functions. Secure layer is responsible for key negotiation and data encryption/decryption, RDP layer is mainly responsible for process remote desktop’s graphics, text, sound, keyboard and mouse events.

RDP is implemented based on a client-server technology, the main working principles are as follows:

1) RDP client and server establish a session for secure communication.

2) During the conversation, client’s input events e.g. keyboard and mouse events are sent to server after being encapsulated and encrypted.

3) Server receives and responds the client’s events, and sends the desktop graphics, sound and other information to the client after encapsulation and encryption.

4) Client receives server’s RDP packets and renders screen’s graphics or plays the sounds from server, thus shows the effect of a remote desktop.

The key point of RDP Proxy module design is data encryption and decryption, because RDP is an encryption protocol. RSA [9] authentication and RC4 [10] data encryption is used in RDP’s secure layer, and the main security mechanisms include: key negotiation, data encryption and decryption, HMAC (Hashed Message Authentication Code) calculation and validation. RDP Proxy mainly includes two parts: one part for the simulated RDP server accepts access from actual RDP clients; while the other part for the simulated RDP client accesses the target RDP server. The working mechanism of RDP proxy is shown in Fig. 3.

The RDP Proxy M simulates a RDP server MS and a RDP client MC. When RDP client C connects to MS, a secure session is established between C and MS, with C’s random salt and proxy’s own RSA public/private key-pair and random salt. At the same time, MC connects to target server S, and a secure session is established between MC and S, with MC’s random salt and S’s own RSA public/private key-pair and random salt. Thus, proxy can decrypt data received from C and send to S through MS with new encryption, on the other hand, proxy can decrypt data received from S and send it to C through MS. From user’s experience point of view, he can access S’s desktop and perform all operations just like directly access to it, but the proxy M can obtain all session data and records them to files.

In order to strengthen user access control, a secure web-based authentication is adopted. User access system’s WEB logon page through WEB browser with specific HTTPS URL, and authorized hosts (including RDP, VNC and X-window) will be listed in a result page after successful logon with correct username and password. Clicking on a target host in authorized hosts list, will activate WEB browser’s local ActiveX control named msrdp.ocx (a RDP client which can download from the Microsoft website), with a access token (generated by WEB authenticate module) and other connection parameters set, the ActiveX control automatically connect to RDP Proxy Server, RDP Proxy checks the client’s validity and determines which target host to connect to by client’s access token.

B. VNC Proxy

VNC (Virtual Network Computing) is widely used as free remote desktop access software with GPL license. VNC mainly consists of VNC Server and VNC Viewer, which can be installed on various operating systems such as Windows, Linux, and UNIX. It is transferable between VNC and RDP under bottom mechanism layer, since both of their central concept is to transmit the server’s graphics changes that have been protocol encapsulated to client and then render to windows interface. By taking advantages of xrdp [11] like protocol conversion technology, VNC Proxy in this paper makes RDP client access to VNC server.

VNC proxy consists of three modules: RDP server module, VNC client module and logon GUI module, as shown in Fig. 4.

When an authorized VNC user select a VNC server to logon through WEB browser, the browser will call RDP ActiveX control as a client to connect to VNC proxy, and the process is as follows:

1) RDP client establishes a secure connection with RDP server module;

2) RDP Server module parses the client’s access token and determines which target VNC server to connect to.

3) RDP server module calls logon GUI module and provides a simulated windows logon interface to RDP client with RDP graphical protocols;

4) The VNC user enters a correct password of the target VNC server in logon window;

5) VNC client module connects to the target VNC server with the password just inputted;

6) VNC proxy converts the protocols between VNC and RDP until session end.
converts RDP client’s mouse and keyboard events to VNC events and transmits them to VNC server. So Windows client can access VNC server without using VNC client, and VNC proxy can record all sessions’ data with RDP protocol formats.

C. XWIN Proxy

X-window is the Unix/Linux graphical user interface, whose core is called a series of X’s graphic interface protocol suite. X introduces the Client/Server architecture. X application is the client and X Server responsible for graphic display is the server. Combining with XDMCP protocol, X-window allows X desktop system on one host to display on X Server on another host via TCP session. The XWIN proxy technology in this paper is an extended application more than a common protocol proxy, which resolves how to access X-window graphic interface through RDP client.

XWIN proxy is implemented based on VNC proxy technology described above, which totally includes three modules: RDP server modules, VNC client module and VNC server module (Xvnc), as shown in Fig. 5.

![Figure 5. XWIN proxy schematic diagram](image)

VNC server module includes an Xvnc program, which is essentially a dual server, both a VNC protocol server and an X server. For each X-window session, XWIN proxy can direct X login form of target host to Xvnc through the following command call:

`Xvnc: <number> -localhost -query <host> securitytypes=none`

*number* is the number of local new virtual desktop, -query parameter means to inquire and accept the X-window graphical interface of a specified host, and *<host>* stands for IP address of a target host.

In Fig. 5, a RDP client can access X-window graphical interface by communication channel between VNC client module and Xvnc in XWIN Proxy. The major procedure is shown as follows:

1) RDP client establishes a secure connection with RDP server module;
2) RDP Server module parses the client’s access token and determines the target XWIN host.
3) RDP Server module forks an X Server with Xvnc command introduced above;
4) Xvnc establishes an X session with the target XWIN host;
5) VNC client module connects to the local Xvnc just forked;
6) VNC proxy converts the protocols between VNC and RDP until session end.

So the RDP client can transmit user input to the server, and the X Server returns screen updates to the client. By this way the client hosts don’t need to install any X Server software e.g. Xmanager or X-Win32.

D. Session Records Playback

The system provides capacity of using only RDP client to access Servers of RDP, VNC and X-window by proxy technology, so all sessions’ data can be stored in the unified format of RDP protocol. Although the transmission of RDP is encrypted, the cipher text can be decrypted by the proxy. Proxy creates a log file for each session, which consists of arbitrary length records, and each record consists of a 4-tuple of values. The first value of each 4-tuple specifies where the data is from, RDP client or server. The second value is the timestamp of the proxy recording the RDP packet. The third value specifies the length of the followed RDP packet length and the fourth value is the RDP packet with plaintext.

In RDP client/server architecture, once the RDP session is established with secure communication, the client waits and receives graphical rendering data from the server, and the server receives input events from client and responds with graphical rendering data of Windows, fonts, pictures, etc. So, if a special RDP client can receive data from simulated RDP server streams which read from RDP session log files, it will also display all the graphical operations of RDP sessions just like a video player.

RDP session playback program is developed on the basis of RDP client prototype, e.g. rdesktop [12]; it provides the capacity of reading RDP session log files and playing them like videos. As shown in Fig. 6, the log reader module simulates RDP session’s TCP streams (from both client side and server side) by reading and parsing records from log files. The server streams which contain graphical rendering data will be sent to RDP drawing module, and be interpreted into corresponding Microsoft Win32 graphics device interface (GDI) API calls [13]. The client streams which contain data of mouse moving events will be sent to mouse events process module, and a simulated mouse will be drawn in replaying window. By controlling the processing time interval of each packet, user
can fast-forward, pause or locate the scenes. So it can be used by the administrator to audit and backward key operations.

V. PERFORMANCE TESTING

The performance of security audit system for remote desktop access reflects in the capacity of number of concurrent sessions in proxy server, as well as every graphics operation’s smooth degree. In [14], by running a small java applet program automatically drawing pictures and characters, author proposed a simple evaluation methodology to evaluate graphics operation's smooth degree and client’s performance according to drawing speed rate.

We present our system performance test by comparing three various technologies for remote desktop access. The configuration of the proxy server is set as follows: CPU (Intel Pentium 4 3.0GHz), Memory (2GB), Network Interface Card (1Gbps).

Test Method: There are 10 Windows XP clients and 5 Windows 2003 Server servers and 5 Linux servers in our tested in total. The clients access the server through the proxy. And we test the performance of RDP, VNC and X-Window. In the test, remote desktop is configured in resolution of 1024*768 and 16bit color. We add 10 concurrent clients each time. At last we plot the CPU utilization ratio of proxy server. The test data is shown in Fig. 7.

![Figure 7. Performance test curve](image)

In our experiment, the traffic targeted to server through proxy server is collected for statistics: RDP’s average bandwidth is about 102 kbps, while VNC’s is 130 kbps, and X-window’s is about 990Kbps. The experiment results illuminate that: for common graphical window operation, the workload of RDP session carried to the proxy server is lowest, and the medium is VNC session, while X-window session has the highest workload on proxy server. Under the condition of graphical window operation without obvious delay in interactive operation, the proxy can handle more than 100 concurrent RDP or VNC sessions, while less than 50 X-Window sessions can be handled in the same condition, this is because for each X-window session, proxy server will instantiate an extra X-server process.

The results demonstrate that for most of small business, only one proxy server is enough to handle the routine auditing task of RDP sessions.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a proxy server technology to audit and monitor the RDP session for the mainline RDP technique, i.e. MS-RDP, VNC and X-window. A RDP-like format has been used to store in-flight and playback the RDP session. It highly reduces the difficulty to audit and maintain the server. This paper only focuses on the RDP stream in graphical level, and how to identify the user’s behavior via the session data remains further study.

ACKNOWLEDGMENT

This work is supported by Natural Science Foundation of China No. 90718040 with the title of “Trusted Computing Environment and Trusted Software Design”. It is also supported by High-tech Program No. 2007AA01Z468 with the title of “A Holistic UTM System Design and Implementation”. Xue-hai Peng’s research is supported by Distinguished Career Program with No.20081D0900200272.

REFERENCES