Towards Ubiquitous Computing via Secure Desktop Service

Pan-Lung Tsai, *Student Member, IEEE*, and Chin-Laung Lei, *Member, IEEE*

***Abstract*—Among all the approaches enabling remote access to the private information stored at some static locations, one approach, often referred as remote desktop control, provides the most intuitive form of such operations. With remote desktop control, people are able to access their personal desktops remotely and operate on them in the same way as they do when sitting in front of the consoles. A side effect of this approach is that the potential damages caused by misuses get severer, and the security issues hence become more important. In this paper, a scheme of authentication via mobile phones is proposed to make the approach of remote desktop control securer. Another benefit of this scheme comes from the globally deployed telecommunication network, which makes the application truly ubiquitous.**

***Index Terms*—Computer network security, distributed computing, interactive computing, internetworking.**

I. Introduction

As the territories of the Internet have been broadened in an explosive manner, the dream of a world in which human users are able to access personal information from every corner on the planet comes closer and closer to the reality. While the connectivity is being achieved, another issue comes to our mind: in what way shall I make use of the data residing in my personal computer 1,000 kilometers away? Of course the answers diverse along with your imagination. However, one common thing is likely to happen in each of these first thoughts: it must not be the same way when I sit in front of the computer touching my own keyboard and clicking the buttons on my favorite mouse, as I did in the past decade. But, why not?

By forwarding the input events such as key presses and button clicks to the remote machine as well as retracting the output events such as screen rendering and audio feedback from the same machine, a user is allowed to operate on her/his own desktop remotely. This kind of operations is often referred as remote desktop control, and the machine under remote control is said to be providing remote desktop service. In this case, the actions taken by the user are treated as the commands issued directly from the console and are usually the most privileged. Therefore, the damages caused by the misuses of remote desktop control are potentially severer than others caused by typical network applications.

Pan-Lung Tsai is currently a Ph.D. candidate in the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, Republic of China. (e-mail: charles@fractal.ee.ntu.edu.tw).

Chin-Laung Lei is currently with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, Republic of China. (e-mail: lei@cc.ee.ntu.edu.tw).

Because the machine providing remote desktop service is so vulnerable, it is reasonable to deploy the remote desktop server behind a firewall and thus isolate it from the public network. The firewall will normally rejects all inbound connection requests to reduce the chance of successful intrusions. However, since the machine acts as a server, there must be some way for a client to connect it from outside world. This is often implemented by leaving a backdoor, typically a secret TCP (Transmission Control Protocol) port allowing inbound connection requests to pass through, on the firewall and hence compromises the security [1].

In this paper, a secure scheme with two variants is proposed to eliminate the security deficiency caused by remote desktop service. The rest of the paper is organized as follows. Section II introduces some related works, and Section III describes the proposed scheme in great details, followed by the security analysis. At the end of this paper, some conclusions are given in Section IV.

II. Related Works

In this section, various implementations of remote desktop control are reviewed at first. Then the idea of parasitic authentication is introduced. For the remaining of this section, an overview of GSM (Global System for Mobile Communications) technology is given along with some discussions on the security mechanisms it provides.

A. Remote Desktop Control

As GUI (Graphical User Interface) has revolutionized the operating model of computers, people are able to make use of their machines in a convenient and intuitive way. In most GUI systems, a human user starts from a screen called *desktop*, which is an analogy to the surface of a desk in the real world. Like the personal stuffs scatter on the user' s own desk in her/his real life, all the resources of the computer are symbolized as small icons spread on the desktop. By moving the pointers over the icons and clicking on the buttons of the pointing device, the user is able to operate on the resources as easily as she/he grabs a pencil on the desk and starts writing.

Since twenty years ago, a number of GUI systems are developed by competing companies as well as research groups. All of these implementations, including Mac® OS, X Window System™, and Windows®, share the common characteristics described above. The population of personal computers also makes such intuitive and convenient user interface indispensable. For a large number of users, it is unbearable to type a series of mysterious keywords under text-mode command

IEEE Catalogue No. 01CH37239
0-7803-7101-1/ 01 / $10.00 © 2001 IEEE.

prompt only to browse the content of the hard drives.

In this age of the Internet, the demand of accessing private information remotely arises. Skilled people are able to make use of the computing resources from remote sites via traditional network applications such as *telnet* and *ftp*. These text-based utilities restrict their users to a relatively small group of technicians that are trained to be familiar with complicated command sequences. In order to grant such privilege to the public, technologies that enable transmitting graphics and interactive events over the network have been developed to make GUI remotely accessible as well. This leads to remote desktop control.

Based on X Window System, Olivetti Research Laboratory of AT&T has developed a technique known as Teleporting [2], which further evolves into VNC (Virtual Network Computing) [3]. VNC employs Web browsers as its front end, and permits users to access remote desktop through a Java™ applet. VNC is famous for its high performance and interoperability with operating systems and existing applications. Another well-known applet-based implementation of remote desktop control is Desktop On-Call [4] from IBM Corporation. Taking advantage of the wide-deployed Web browsers, VNC and Desktop On-Call require no installation on the client machines and are virtually available everywhere.

There are still other proprietary software implementations that enable remote desktop control on personal computers. NetMeeting® [5] and Windows 2000 Terminal Services [6] from Microsoft Corporation are recently added features that help Windows users to access their desktop remotely. Before these features are built into the operating system, pcAnywhere® [7] from Symantec Corporation might be the most popular remote desktop control software on the Windows platform.

B. *Parasitic Authentication*

Preventing unauthorized access to private resources while retaining authorized use is regarded quite difficult to achieve. Traditional schemes of authorization are either too complicated to be realized or so inconvenient that can not be accepted by the users. If the authorization procedure itself requires even more efforts than the transaction following it, the whole thing becomes less meaningful.

In [8], the idea of parasitic authentication is proposed to provide an effective solution to the authorization and authentication problems of e-wallets. By temporarily delegating the responsibility of authorization to a secondary device, the need for the user to enter the PIN (Personal Identification Number) or to prove their identity every time the transaction is being carried out is successfully eliminated. Besides, the security is enhanced because it is less possible for the user to lose her/his e-wallet and the secondary device simultaneously.

In the proposed scheme, the spirit of parasitic authentication is exploited so that the session of remote desktop control can be authorized through a secondary device. As a result, the remote desktop service mentioned previously can thus be improved to be secure desktop service.

C. *GSM Technology*

GSM is known as the second generation of wireless communication technology. Compared with its predecessor, GSM is not only more versatile but also securer. In addition to the voice service it is originally designed to provide, GSM also support various data services such as SMS (Short Message Services) and WAP (Wireless Application Protocol). To the end of January 2001, there are 457.8 million GSM subscribers in the world [9].

Among all the data services available in GSM technology, SMS is the most flexible one. GSM operators are currently offering dozens of consumer applications and corporate applications based SMS. SMS can be regarded as wireless data transport capable of carrying various application-specific information without limit on distance. To the end of January 2001, there are 15 billion SMS messages sent per month.

GSM security primarily comes from confidentiality techniques and protocol definition. The security services provided by GSM include anonymity, authentication, signaling protection, and user data protection [10]. The service of anonymity prevents unauthorized tracking of phone calls, and the service of authentication is designed for billing purpose. The service of signaling protection conceals sensitive information such as telephone numbers from the outside world, and the service of user data protection makes eavesdroppers fetch no useful information from the radio channel. In general, communications via GSM are safer than most data transmission through the Internet, and can be considered as a secure channel.

III. The Proposed Scheme

When setting up conventional remote desktop service, the server machine is deployed in the protected domain of a firewall. For the clients to be served, there will be a backdoor left on the firewall so that inbound connection requests can be passed to the server without difficulty. However, this backdoor will become the security hole of the system because most successful intrusions start from scanning for such exposed ports.

The problem can be solved by delegating the requesting of the service to a secondary device through another channel. The selection of such secondary device must satisfy an important criterion: it must be deployed at least as widely as the remote desktop client is; otherwise, the whole scheme will become meaningless. Besides, the alternative channel for the secondary device to request the remote desktop service must offer basic level of security, or allow the implementation of security mechanisms upon it. Taking all these into account, there is a perfect candidate for the secondary device: the mobile phone.

The extensive use and growth of mobile phones can be easily perceived by the world-wide devotion to the mobile communication technology such as GPRS (General Packet Radio Service) and 3G (3rd Generation), as well as the proliferation of television commercials promoting a wide variety of handsets from competing phone companies (e.g., Nokia Corporation and Motorola, Inc.). In addition, the GSM standard also promises certain level of security and thus makes

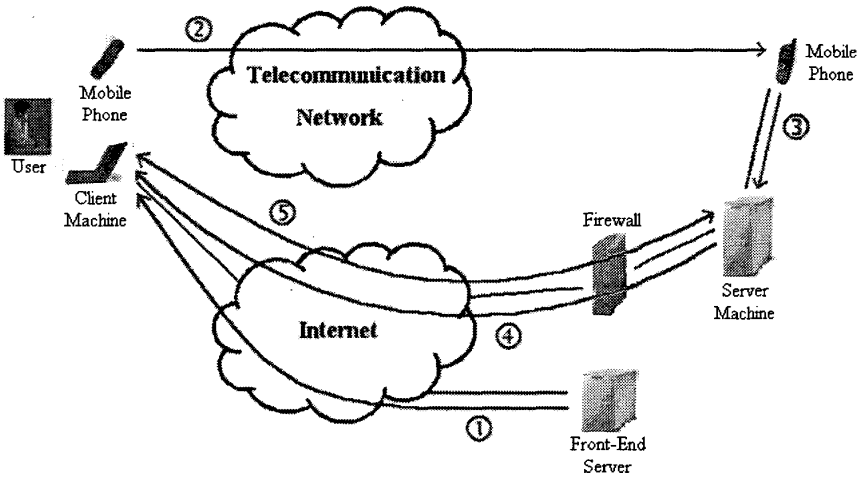

Fig. 1. Architecture of secure desktop service.

the telecommunication network a relatively secure channel. Based upon these characteristics, secure desktop service is designed to remedy the shortcomings of remote desktop service.

A. Secure Desktop Service

Fig. 1 shows the architecture of secure desktop service. The secure desktop server resides in the protected domain guarded by a firewall, which rejects all inbound connection requests with no exception. The server is also equipped with a mobile phone to accept requests. An additional server (needs not to be protected) is responsible for providing the front-end software to the client machine.

To acquire the secure desktop service, the following steps are carried out in sequence:

1. The user opens a Web browser of any kind and downloads the secure desktop client via HTTP (HyperText Transfer Protocol). After the client software is executed, it will present a randomly generated session key and a randomly selected TCP port number to the user and start listening on the selected TCP port.
2. The user composes a well-formatted message consists of the IP address of the client machine, the session key and the TCP port number generated in step 1 and her/his identification information (usually the account/password pair) to be authenticated by the secure desktop server. Then she/he sends the message to the secure desktop server using SMS through the mobile phone.
3. The mobile phone at the server side receives the SMS message and passes it to the secure desktop server.
4. The secure desktop server authenticates the user according to the identification information included in the SMS message. If the access is granted, the server initiates a connection to the specified IP and TCP port and sends the session key as the first message.
5. The secure desktop client accepts the connection and verifies the session key. If nothing goes wrong, a secure desktop session is started.

When the user travels with her/his own computing device (e.g., a notebook or a handheld PC), there is a great chance that the client software has been installed beforehand. Under such circumstances, there is no need to perform the initial download of the secure desktop client, and the deployment of a front-end server becomes unnecessary. However, if the user does not carry any computing devices and still wishes to access her/his personal stuffs ubiquitously, then the front-end software will require some special treatment as described below.

B. Secure Desktop Service with Plug-in Client

In order to achieve the goal of ubiquitous access to the secure desktop service, the semi-Web-based procedures depicted previously must be modified to be a fully Web-based solution. By rewriting the front-end software (the secure desktop client) as a plug-in to the browser, users will never suffer from repetitive installation again. In this case, the front-end server is just a normal Web server and the first step of the procedure is automated and becomes:

1. The user opens a Web browser of some supported brands and navigates to her/his home site (the front-end server). Then the plug-in is downloaded and started automatically. A randomly generated session key and a randomly selected TCP port number is presented to the user in the browser window, and the secure desktop client starts listening on the selected TCP port.

As some of perfectionists may notice that this modified scheme exhibits a minor but unpleasant drawback. Plug-in software is not only platform dependent but also browser dependent. This means there will be several versions of the secure desktop client to accommodate various browsers for true ubiquitous access.

C. Secure Desktop Service with Applet Client

Software portability is never an easy problem. However,

there has been a solution to the portability problem on the Web since 1995. As the promoters like to say, "Java comes to the rescue, again"! If the secure desktop client is implemented as a signed Java applet, then the user can still activate it in a browser (without installation) and the drawback of maintaining multiple versions of the front-end is also eliminated. The Java applet must be digitally signed so that it is trusted by the browser and hence is capable of listening for a TCP connection request on the remote machine [11]. The initial step turns out to be:

1. The user opens a Java-enabled Web browser and navigates to her/his home site (the front-end server). Then the Java applet is downloaded and started automatically. A randomly generated session key and a randomly selected TCP port number is presented to the user in the browser window, and the secure desktop client starts listening on the selected TCP port.

Since nearly all of modern Web browsers are Java-enabled, this approach potentially satisfies more users than the previous one does.

D. Security Analysis

The proposed scheme of secure desktop service incorporates little assumptions more than generally adopted in a secured system protected by a firewall on the Internet. The secure desktop server behind the firewall, the mobile phone, and the link between them are regarded as a single trusted component. As discussed in Section II, the wireless telecommunication network employing GSM technology can be safely considered as a secure channel separated from the Internet.

The denial-of-service attack is excluded from this analysis because all network-based applications are naturally vulnerable to such attack and there exists no effective countermeasure yet.

Under these assumptions, three security issues need to be examined. The first one is the acquirement of the client software. For the basic scheme of secure desktop service, the client software can be installed on the computing device beforehand. In this case, the on-line acquirement does not happen at all, so the situation is considered temper-proof. If the client software is going to be downloaded on demand, it is likely to be tempered by malicious attackers, unless the client software is implemented as a signed Java applet, as in the second variant of the proposed scheme. Therefore, the client software should be either installed on the computing device beforehand, or implemented as a signed Java applet.

The second issue is the benefits obtained by the firewall. As mentioned above, the initiation of a secure desktop session starts from an outbound request rather than an inbound one, the firewall now just rejects all the inbound requests without a hesitate. The firewall can be implemented to discard all TCP connection requests silently and hence prevent port scanning, which is the first step of most successful intrusions. Besides, discarding such packets blindly also reduces the amount of computation and hence improves the performance of the firewall observably.

The last security issue regards the information exchange during the secure desktop session. Since there is a secure channel (the telecommunication network), it is feasible to exchange some shared secret via the secure channel before the connection is established. Afterwards, the shared secret can be used to encrypt and decrypt the information exchanged during the session. From the viewpoint of the secure desktop session being established, this is equivalent to the one-time pad scheme, which is computationally unbreakable.

IV. CONCLUSIONS

In this paper, a secure scheme is proposed to complement the weakness of ordinary remote desktop service. With secure desktop service, the session between the secure desktop client and the secure desktop server actually starts with an outbound connection request and the need for a backdoor on the firewall is therefore eliminated. This is accomplished by delegating the requesting of the service to a secondary device through an alternative secure channel. Besides, the Web-based solution along with the use of mobile phones (and hence the global telecommunication network) makes perfect sense on the idea of ubiquitous computing, because Web browsers are installed nearly everywhere while mobile phones are carried by almost everyone.

With a little bit modification, the proposed scheme can also be applied to virtually all client/server applications to change the direction of connection initiation. Doing this will get rid of the numerous backdoors (one per service) on a firewall and reduce the chance of successful intrusions.

REFERENCES

[1] D. Brent Chapman and Elizabeth D. Zwicky, *Building Internet Firewalls*, First Edition, O' Reilly & Associates, Inc., September 1995.

[2] Tristan Richardson, Frazer Bennett, Glenford Mapp, and Andy Hopper. "Teleporting in an X Window System Environment." *IEEE Personal Communications*, Vol. 1, No. 3, 3rd Quarter 1994, pp. 6-12.

[3] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper, "Virtual Network Computing," *IEEE Internet Computing*, Vol. 2 No. 1, January/February 1998, pp. 33-38.

[4] "Desktop On-Call," IBM Corporation, http://www-6.ibm.com/jp/esbu/E/dtoc/index.html.

[5] "NetMeeting Home," Microsoft Corporation, http://www.microsoft.com/windows/netmeeting/.

[6] "Exploring Terminal Services," Microsoft Corporation, http://www.microsoft.com/windows2000/guide/server/features/terminalsvcs.asp

[7] "pcAnywhere 10.0", Symantec Corporation, http://www.symantec.com/pcanywhere/.

[8] Tim Ebringer, Peter Thorne, and Yuliang Zheng, "Parasitic Authentication to Protect Your E-Wallet," *IEEE Computer*, Vol. 33, No. 10, October 2000, pp. 54-60.

[9] "GSM World," GSM Association, http://www.gsmworld.com/.

[10] Charles Brookson, "GSM Security: A Description of the Reasons for Security and the Techniques," *IEE Colloquium on Security and Cryptography Applications to Radio Systems*, 1994, pp. 2/1-2/4.

[11] Scott Oaks, "Java Security," First Edition, O' Reilly & Associates, Inc., February 1999.