Problems with Secure On-line Banking

A growing awareness of the commercial benefits of online banking have contributed to a sense of urgency among banks to deploy such systems. Electronic banking aims to provide easy access to banking services for customers. Both banks and customers stand to benefit from the introduction of electronic banking schemes, since the bank can offer its services at much lower cost, while the customer can access the services from any location at any time. Indeed, these benefits can obviate the need for branches or tellers altogether, resulting in the emergence of so called virtual banks, who conduct business purely on an electronic basis.

Internet-based electronic banking schemes rely on the existence of an Internet connection over which a customer can access a bank's services. There are a number of models that can be adopted to implement such a system. Customers can use existing "browser" software such as Netscape's Navigator or Microsoft's Internet Explorer as the client interface to the banks system. In this model, the bank's server provides HTML forms-based interface through which customers can make requests and conduct transactions. Communications security is provided by the SSL protocol which is built into the browser. Or else Customers can download Java applets from the bank-server's web site. The downloaded applet provides the interface through which customer transactions can take place. In this case, communication security is provided by the applet itself. Or Customers can download executable files from the bank-server's web site, which have been pre-compiled for a number of common platforms. The running executable provides the customer interface as well as the communications security.

Most Internet banking applications authenticate users based on an account/client number and a secret PIN. The account number is generally typed directly into a text field, while the "secret" PIN is entered via a GUI-keypad using the mouse. Account numbers are displayed in plain text, while an "*" is written to the screen as each keypad key is pressed. Once the user's information is complete, the application contacts the bank server and attempts to authenticate the user.

These systems all require privacy and integrity of transactions, and most importantly strong authentication. However, while these issues are usually well addressed by the communications protocols used in these systems, there are fundamental weaknesses in the security of the platform on which these applications run. It is well known that many popular operating systems do not provide sufficient protection from malicious programs such as viruses, which may be used to subvert authentication protocols by capturing authentication information such as a PINs or passwords that are entered by users.

Network security is well known in all the Internet banking applications, and is based on establishing a secure channel between the user's client application and the server running at the bank. The protocols used provide connection security that has three basic properties.

(1) The transaction is private. Public key encryption is used after an initial handshake to negotiate a session key. Symmetric cryptography is then used for subsequent data

encryption. (2)The peer's identity can be authenticated using public key cryptography. (3) Message transport includes message integrity check

The not-to-be-trusted computing base

The problem of ensuring privacy and authenticity between peers over a public network is well known and solutions are widely available to implementers. However, that in the development of Internet banking applications, general attention has been too strongly focused on the issues of network security, and not focused enough on the issues of security at the user's platform.

In trusted computing systems evaluation there is the concept of a Trusted Computing Base (TCB). This means protection mechanisms within a computer system, including hardware, firmware and software – the combination of which is responsible for enforcing a security policy. There are two main ways of implementing such a system. (i) A security kernel – This is a small part of a trusted system that mediates access to all objects according to a security policy. (ii)A front-end security filter – This is a process that filters incoming or outgoing data according to a specified security policy. Because, the TCB is a small and identifiable part of the system, it is easy to be confident in its ability to enforce the security policy correctly. In general, the existence of a TCB is a prerequisite for enforcing the security policy in any secure system.

However, in the case of an operating system such as Windows 95, there is no identifiable TCB on which a trusted system can be based. The operating system does not provide a security kernel to control access to resources. Any application can access any file on the system, and can observe (and perhaps alter) the flow of messages between hardware devices such as mice and keyboards and applications. Because there is no way for an application to prevent other applications from accessing resources and information, and this feature is explicitly not supported by the operating system, an application cannot enforce its security policy in this way.

When a virus runs on a platform that has no concept of ownership or access control, any application that runs on that platform may be compromised. Even if access control is enforced by the O/S or by additional hardware, the virus techniques can still work. While the O/S process management can ensure that other users' processes are safe from direct intervention by the malicious process, the processes owned by the 'owner' of the virus can still be compromised. Furthermore, techniques exist for eavesdropping on the traffic that flows between windowing server and client applications and such techniques can compromise any GUI-based application.

The lack of a TCB on which to base secure systems allows an attacker to subvert the security mechanisms of a banking application using viruses by attacking the way the user enters authentication information such as a PIN or password. This is done by having the virus lay dormant until it detects the critical points when such information is to be entered, then monitoring system events such as key presses and mouse clicks to capture authentication information for later use.

Possible Solutions.

The potential value to an attacker of hacking Internet banking applications means that they may go to extraordinary lengths in order to do so. Most of the banking applications are using simple passwords as the primary form of authentication. The following are possible solutions.

1.One-Time Passwords

In one-time password systems, the password entered is only valid for a single login, and then changes in a secure way. The benefit of such a system is that monitoring by an attacker is useless, as the information available to them can not be reused. However the disadvantages of this scheme are that administration is complex, and the user has to store a list of keys on a sheet of paper, creating the potential for theft and misuse. For these reasons, it is unlikely that a bank would adopt such a scheme, or that the public would accept it, even if it were implemented.

2.Token Based Authentication

Token based systems provide authentication of a user by requiring them to demonstrate the possession of a physical object or token which is unique to that user. There are basically three types of tokens. Memory tokens - These tokens do not contain any processing capacity, but contain authentication data stored in magnetic, electronic or optical form. Second one is Microprocessor tokens - These tokens contain a microprocessor in addition to memory. Such tokens may implement cryptographic algorithms for encryption on the card. Microprocessor tokens are commonly referred to as smart cards. Many smart cards have properties that make them resistant to tampering. Third one is Hand held password generators - This class of items includes both hardware calculators for the one-time password mechanisms as described above, and challenge-response calculators that allow a user to enter a challenge from the server and calculate the appropriate response. Unfortunately, all these schemes require the purchase of extra hardware, making them unattractive to both banks and their customers.

Conclusions

Internet banking has the potential to provide a useful service to customers and banks alike. However, unless more consideration is given to the design of secure applications that can operate within an untrusted environment, Internet banking will remain an activity that is associated with a significant level of risk.