White Paper

Passwords for access to secure Internet services are not enough

Expertron Group (Pty) Ltd

Tel: +27 (0) 12 349-0390
Fax: +27 (0) 12 349-0360
info@expertron.co.za
http://www.expertron.co.za

August 2001

Abstract: Many online Internet services provide access to secure systems by authorized users. These services typically include Internet banking, access to private medical information, financial portfolio management portals for investments and insurance, etc. Authorized users are required to authenticate their identity before access to these secure services is granted. This paper briefly reviews the three mechanisms that can be used for user authentication, and then discusses the weaknesses of the method most typically used, namely, password-based user authentication. A demonstration consisting of a computer virus will be described wherein the cited vulnerabilities of this authentication method can easily be exploited in a practical and feasible way so that authentication credentials (such as usernames and passwords) can be "leaked out".

Introduction

This paper deals with the security issues around access to secure Internet-based subscription services. Here, a subscription service refers to any kind of online service where an authorized user is required to authenticate his/her identity before being granted access to privileged information or services. These subscription services include access to Internet banking and other financial services, access to health services (such as sensitive/private medical information provided by medical funds), etc.

There are three fundamental requirements when users of computer-based systems, where these systems consist of client and server computers, gain access to secure services:

A. Authentication of the user (and/or client computer) making use of the secure service. This allows the server to confirm the identity of the user (and/or client computer).

B. Authentication of the server providing the secure service. This allows the user to confirm the identity of the server.

C. Encryption (or secrecy) of the communication channel between the server and the client computer. This may be necessary when a high degree of confidentiality is required such as during a private transaction, or when messages need to be digitally signed.

The first of the above three concerns, authentication of the user (A), is the most challenging, particularly where there are many users who are widely distributed. In this paper, the problems associated with user authentication will be briefly discussed, and it will be shown that, of all the available means for authenticating users, password-based authentication is the most practical (and hence, most widely used), but the least secure. Then, a demonstration will be discussed, consisting of a computer virus, wherein the vulnerabilities associated with password-based authentication can be exploited in a practical and feasible way, namely that it is easy to obtain a user's authentication credentials (username and password) in an unscrupulous manner.

User authentication in review

Users may identify themselves to servers usually by providing a “username” or “user number”. Since usernames and numbers are not secrets, it would be easy for an intruder to pose as a user and gain access to that person’s secure services. To prevent this from happening, the identity of the user must be authenticated. User authentication (proof of identity) can commonly be done in three ways:

What you know: Secrets

> If the user can show that he or she is in possession of a secret such as a password, PIN, cryptographic key or certificate that only the real user is supposed to know, it may act as proof of identity.

What you have: Hardware Tokens

> If the user can show that he or she is in possession of a hardware device, such as a magnetic card, smart card, cryptographic token or calculator, that only the real user is supposed to have, it may act as proof of identity.

What you are: Bio-metric Measurements

> If the user can show that a measurement of part of his or her body (such as a fingerprint, retina scan, photograph, etc.) matches that of the registered user, it may act as proof of identity.

User authentication based on a secret, particularly where this secret is a password or PIN that is managed by the user, is generally considered a **weak** authentication mechanism because

- users are known to choose weak (short, easy-to-guess) passwords that can easily be remembered;
- users may write down or share passwords;
- passwords are static; users do not usually change their passwords on a frequent basis, and if the secret “leaks out”, the user can never sure that his or her secret is not known by a third party unless this is changed on a frequent basis.

Authentication based on a secret such as a suitably long cryptographic key (i.e. where decipherment is infeasible) is considered a **strong** user authentication mechanism.

Authentication mechanisms based on hardware tokens are **strong** authentication mechanisms because identity of the user cannot be verified by guessing a secret. Furthermore, the user can be assured that as long as she is in possession of the hardware token, access to her secure services by a third party is impossible.

Authentication mechanisms using bio-metric measurements are also **strong**, since these are not based on a secret that the user knows, and therefore has to remember. However, bio-metric measurements of a particular user do not change (i.e. they are static, since there is an intrinsic association with “what” the user is), and hence, when the measurement is encoded into some electronic format that is transmitted over open communication channels, this information must be kept secret. In this sense, due to the static nature of bio-metric measurements and their transmission over open channels, these are essentially no different from authentication mechanisms based on secrets.

Hence, authentication credentials can either be

- **Static**, such as cryptographic keys that must be installed on computing equipment (both client and server); passwords that users must remember; measurements of bio-metric entities such as fingerprints, retinas, voice, etc;

- **Dynamic**, such as one-time passwords (OTP) generated by some hardware token, where possession of the token, and hence identity, is proved by offering the OTP. One-time passwords may either be generated by algorithmic processes based on some cryptographic key, or generated by random processes.

Dynamic credentials provide a significant advantage over static credentials: with dynamic authentication credentials, the validity of a one-time password for a particular authentication session expires after some time, and even if the password does “leak out” it cannot be used for future authentication sessions.

Strong user authentication methods are, in many cases, impractical and expensive. This impracticality (or the *distribution problem*) refers to the difficulty of “rolling out” the user authentication technology. In all cases, either secret keys, hardware tokens such as cryptographic tokens and calculators, software programs or devices such as smart cards, card readers and bio-metric scanners must be distributed to all the users. Usually there are many more users than servers, and where the servers may be centrally located, users are usually widely distributed.

Hence, password-based authentication, although fundamentally weak, is often the most practical and cost-effective way of authenticating large numbers of users.

Security model for secure Internet services

The architecture of the secure online Internet services considered here is shown in Figure 1. In this architecture, authentication of the server, as well as encryption (i.e. secrecy) of the data channel is maintained through a public key located on the server. Hence, the outstanding problem is that of authenticating the user. As cited above, of all the possible authentication mechanisms, password-based user authentication is the most practical, and therefore most frequently used.

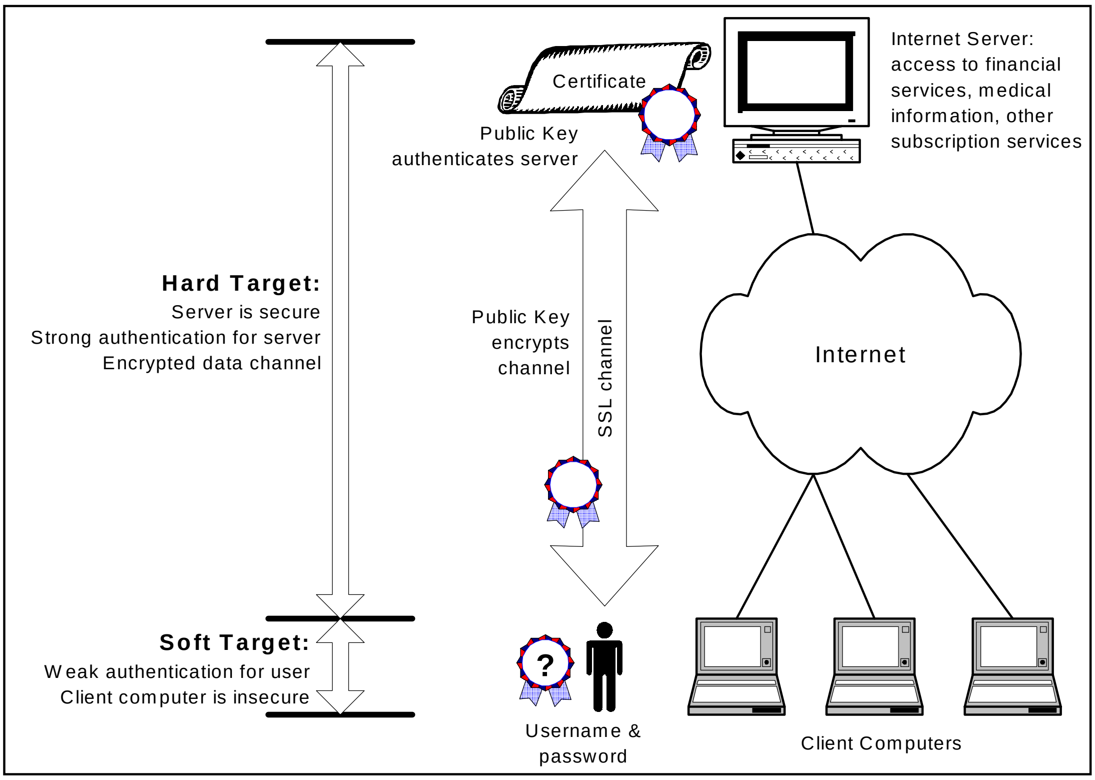
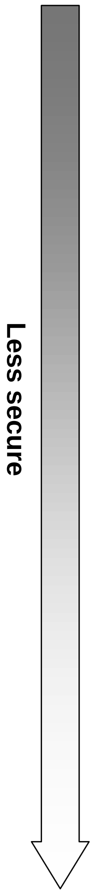


Figure 1: Architecture for secure Internet services

The security model that is implemented in the architecture in Figure 1 comprises four components:



1. The underlying **mathematics** describing the encryption algorithm. This forms the theoretical basis for authentication of the server and encryption of the data channel, and since this is security implemented in the purest form, it is the strongest component of the model.

2. The **program code** implementing the mathematics. This layer represents the real-world implementation of the mathematics, and may deviate from the strength of the pure mathematical foundation. Hence, security as this level may be weaker than the mathematical core.

3. The **server computer** that runs the program code. Adding greater complexity, the operating environment within which the program code is executed is exposed to greater vulnerabilities, and if this environment is compromised, the security implemented by the program code is undermined. Although this layer forms a weaker component than the program code, the risk of a security breach can be greatly reduced if this equipment is maintained by trained personnel within a controlled environment.

4. The **user** accessing his/her online subscription service. If at this level the authentication information is compromised, the entire security model is comprised. Here, the secrecy of the authentication information, the password, is the user's responsibility. Furthermore, online subscription services are accessed from a computer that is connected to a public network, and users are often not sufficiently trained in the subtleties and know-how of maintaining the security of computing equipment within this hostile environment. Hence, the user, and in particular the equipment that he/she makes use of, are the weakest components within the security model. It is at this layer where an attack is likely to be targeted to obtain and compromise authentication credentials. A method for demonstrating such an attack is the topic of the next section.

A method for stealing user passwords

In this section, it will be described how passwords can be stolen by infecting the user's computer with a virus that is executed by exploiting the curiosity or ignorance of the user, or the vulnerabilities of the user's computer.

This virus may comprise two components:

1. The Installer: software that downloads the virus Payload from the Internet and executes it on the user's computer.

2. The Payload: the actual virus; software that detects authentication activity and steals authentication credentials (usernames and passwords).

The Payload comprises software that monitors the user's keystrokes, implementing a filter for detecting user authentication activity, such as signing on to an online subscription service. A simple filter scans for a combination of keystrokes that corresponds to the URL of, for example, an Internet banking site (there are other methods for detecting such information if this is not explicitly typed in, for example where a URL is selected from a drop-down menu or cut-and-paste from a buffer). Once this has been detected, the filter will scan for sequences of numbers that are candidates for the user's authentication credentials, such as an account number and a PIN (these numbers are usually separated by either a particular keystroke event (a Tab key) or a mouse event). After this authentication activity has been detected, the authentication credentials (account name or number and password) is emailed to an anonymous email address, or any other public forum providing anonymity for the cracker where stolen passwords can easily be retrieved.

The following three methods for infecting the user's computer are proposed and can be demonstrated.

Method 1: Exploit user curiosity

Delivery mechanism: HTML-formatted email containing a hyperlink to an interesting executable file.

Here, the user receives an HTML-formatted email containing a hyperlink to an executable file. This is important, as no attachment is required (which may otherwise have been stripped off or rejected by mail transport software). When the user clicks on the hyperlink, the executable may either be run directly from the URL where it is located, or it may be saved to disc and run locally from there. This executable file is a Trojan Horse that has been infected with the virus Payload, i.e. no separate Installer is required. If the user executes the program, and hence the virus, authentication credentials leak out the next time she authenticates herself to a particular secure Internet service that the virus is programmed to detect. There is no defense against such an attack if the user co-operates.

Software for creating Trojan Horse programs is freely available, and may be modified so that the resulting Trojan Horse program containing the virus Payload is ignored by anti-virus software. In the demonstration referred to here, the Silkrope Trojan Horse program was used to hide the Payload inside a simple Microsoft Windows game. In this demonstration, the Trojan Horse program was not detected by Norton AntiVirus.

It should be noted that most viruses (e.g. “I Love You” worm, Melissa virus) are spread in this manner, i.e. where the user is sent email containing an attachment with malicious executable content which the user must explicitly execute, or directed to a site from where the malicious code can be saved and executed.

Method 2: Exploit user curiosity and browser vulnerability

Delivery mechanism: HTML-formatted email containing a hyperlink to an interesting Web page containing malicious active content

In this example, the user receives an HTML-formatted email containing a link to an interesting Web page. Even if the user is savvy to the dangers of executing programs of unknown origin, many users are unaware of the potential dangers of visiting Web sites that may contain active content (such as VBscript and JAVA) that may exploit a browser vulnerability and install a malicious program onto the user’s local hard drive. The user clicks on the hyperlink that opens a browser window to view the Web page at the URL. The malicious code that is embedded in the Web page either installs the virus Installer, or the Payload itself, to the Startup directory of the Windows computer. The next time the user reboots and accesses her secure online Internet service, her authentication credentials leak out.

In this example, the well-known Scriptlet.typelib vulnerability was exploited. This vulnerability affects Internet Explorer 5.0 and 5.5 (SP1), irrespective of the Internet Explorer security settings (unless scripting is disabled, which will make most Internet sites unusable). Although a patch is available for this vulnerability, this must be explicitly applied, and this is unlikely to be the case with most users. Anti-virus software, such as Norton AntiVirus will however detect activity that exploits this vulnerability. Hence, without anti-virus software, or where anti-virus software is present but Script Blocking has been disabled, the user is vulnerable. This vulnerability has been exploited by the BubbleBoy worm.

Method 3: Exploit lower browser security settings

Delivery mechanism: HTML-formatted email containing a malicious VBscript

Here, the user receives an HTML-formatted email that contains a malicious VBscript which implements the Installer. If the email is viewed in the preview pane of Outlook Express (in some cases this could even be opened automatically without any intervention from the user), the Installer is activated, then downloads and executes the Payload, and the user’s computer becomes infected. The next time he accesses his secure online service, authentication credentials leak out.

This example requires the user to activate ActiveX controls not marked as safe (via the Internet Explorer security settings). Furthermore, such activity will be detected by anti-virus software, unless Script Blocking is disabled. This example demonstrates the necessity for the user to ensure that the security settings of the computer from which secure online services are accessed are correctly configured, and exposes the vulnerability of accessing secure services from an untrusted computer, such as at an Internet Café.

There are other hazards associated with accessing secure services from an untrusted computer (such as at an Internet Café) where, for example, unknown to the user, passwords are cached by the browser software, or similar to the virus above, a keystroke monitor program may be installed. In each case, the authentication credentials can be retrieved by the criminal at a later stage.

Conclusion

For practical reasons, password-based authentication is most often used to authenticate users when accessing secure online subscription services. However, passwords, particularly where these are managed by the user, are the weakest authentication mechanism. Furthermore, both the user and the user's computer form the weakest component, and hence the softest target in the security model. In this paper, a demonstration was described where the user's computer is the target of an attack. Here, the attack consists of a simple virus that is executed on the user's computer. This virus monitors the user's keystrokes and filters out authentication credentials such as usernames and passwords. This authentication information is then "leaked out" by transferring it to an anonymous email account where it may be retrieved by an unscrupulous third party.

In conclusion, the following important points should be noted:

- The user can never be sure that her password has not "leaked out". If this password is static, or changed infrequently, access to the user's privileged information may be unknowingly gained by an unscrupulous third party at any time.

- The user should never access secure online services from a computer that is not under his direct management, and access to his computer must be controlled. Settings on the computer, or installed keystroke monitoring software, can result in passwords being stored or captured, and may be retrieved later by an unscrupulous third party.

- The user should take all measures to maintain the security of the computer from which secure Internet services are accessed. These include all operating system updates and patches, as well as anti-virus software with the latest virus definitions.

- Anti-virus software does not protect the user from future (unknown) attacks, only the past (known, existing attacks), and where the user's curiosity and ignorance are exploited there is often no defense. Hence, this will always leave the user vulnerable to the "next attack", against which there is little defense.

- Authentication mechanisms that are based on a static secret (such as a password that is infrequently changed or a bio-metric measurement) are vulnerable to future attacks if the secret is compromised. For this reason, an authentication mechanism that uses dynamic authentication credentials, such as a one-time password that changes for each authentication session, is much stronger within a hostile, open environment such as the Internet.