

Remote administration software attacks on systems using public key security with and without smart cards

I. INTRODUCTION

Public key technology is increasingly used to protect against a wide range of potential security attacks on data and data transmissions and to prevent identity theft. At the same time, remote administration tools are becoming increasingly popular for centralized administration and troubleshooting. Remote administration tools offer both considerable savings for enterprises as well as considerable potential for bypassing security systems such as public key-enabled systems.

This white paper investigates the potential security impacts on public key enhanced security systems by remote administration tools. The general product class name was used, rather than specific vendor names, because the issues investigated are related to the technology and not to any particular vendor's implementation. The evaluation of the interaction of these two technologies was done using several remote administration applications, principally Back Orifice, and an application from a PKI solutions provider. The evaluation also included the impact of adding a smart card to the PKI to protect against security attacks using these remote administration and implementation tools.

Remote administration takes a wide range of forms, from the open and obvious to the intentionally clandestine. As an example, Back Orifice describes itself as, "a remote administration system which allows a user to control a Win95 machine over a network using a simple console or GUI application. On a local LAN or across the Internet, BO gives its

user more control of the remote Windows system than the person at the keyboard of that machine." (Back Orifice web site, www.cultdeadcow.com/tools). The most satisfying, or the most frightening, thing about these claims is that they are absolutely true.

Using Back Orifice or other remote control programs as administration or trouble-shooting tools is ideal. They allow an administrator to take complete control of a computer. The flip side of full control is that an improperly motivated individual can do essentially anything that the legitimate user can do. This has a serious impact on security that we will address in the remainder of this paper.

Back Orifice also advertises that it can get detailed system information, including:

- Current user
- CPU type
- Windows version
- Memory usage
- Mounted disks (including hard drives, CD-ROMs, removable drives and remote network drives) and information for those drives
- Screensaver password
- Passwords cached by the user (including those for dialups, web and network access, and any other password cached by the operating system)

Clearly password attacks are an area for considerable security concern, but it is not just password theft that



should worry potential users of these products. This paper will limit its discussion to client workstations, as there are obvious security issues if one gains illicit access to a workstation. Many applications grant and maintain access to networked data and computing resources based on an authenticated connection to a specific workstation. Because of this, we will look at two basic security attack types:

- Stealing passwords and/or credentials,
- Stealing legitimate sessions.

While access control systems are beginning to use PKI, many systems still rely on user names and passwords for authentication. In either case, a user's credentials are the keys that grant access to all local and networked resources. Therefore, if one can manage to steal a user's credentials, one has ready access to all of the resources granted to the legitimate user. The obvious threat of credential theft is typically addressed with enterprise security policies. Password policies that require regular changes of strong passwords offer good protection, however they are generally not enforced because of the difficulty for the user and high password recovery costs. META Group estimates a \$25 cost to an organization every time a password has to be reset.

While low-tech attacks such as shoulder surfing are very powerful, a wary user can avoid them. However, with rogue remote administration programs, an attacker can watch everything that happens in a computer without the user's knowledge. As a result, being wary is just not sufficient.

The second attack explored is session theft. In this case, it is not necessary to steal authentication credentials; it is only necessary for an attacker to wait for a legitimate authentication to take place. Then the attacker simply takes over the legitimate

session and has free access to all of the resources granted by the legitimate authentication. This attack can also be thwarted by vigilance, but rarely is. Computer users often leave their computers connected to authenticated sessions to get a cup of coffee, answer the phone, or take care of any one of hundreds of distractions encountered during the day.

With the increasing popularity and availability of user-friendly Single Sign On (SSO) solutions, the potential for a security breach is greatly expanded. Once a strong and legitimate root authentication is complete, the attacker would never again be challenged for further authentication (assuming the SSO solution does no further identity checks in the background). The balance between security and ease of use is always an issue that needs to be addressed in the design of security solutions.

II. IMPLICATIONS OF REMOTE CONTROL PROGRAMS IN SOFTWARE-BASED PKI SECURITY SYSTEMS

What are the security advantages offered by PKI? The suppliers of PKI products and solutions often claim PKI-based systems provide four advantages to the user. These are:

- Confidentiality
- Authenticity
- Integrity
- Non-Repudiation

Clearly, if credentials or sessions are stolen, the benefits above cannot be delivered. Therefore, we must look at how PKI can thwart the loss of credentials to prevent the loss of these four security advantages.



Identity authentication is at the root of all PKI-based applications. Public key infrastructures offer ready access to trusted information about an individual. This information may be limited to a public key/distinguished name relationship or include a wide range of access control and permission information. This public information is contained in a digital file called a certificate. This certificate contains a distinguished name of an individual and a public key for which the individual has demonstrated possession of the associated private key.

Given a certificate, systems can “easily” determine if the legitimate owner is actually requesting services such as access control to a protected resource. The proof of identity is based on the user’s ability to digitally sign data such that the signature can be verified with a valid certificate pointed to by a locally supplied “trial” certificate. If a signature can be verified, the system knows that the legitimate owner of the certificate is requesting the service. However, as in the case of user names and passwords, if public key credentials can be stolen, then anyone could successfully create a digital signature. Thus, there must be a way to protect the user’s private key credentials.

In the experiments outlined below, a PKI certificate was used with several PKI-ready applications. While one of the major PKI providers was used in this investigation, the same would hold true for any PKI system. There are no inherent weaknesses in the PKI provider selected and the system is, in fact, secure, very user friendly, and feature rich.

There are many ways that a session can be stolen. For example, people can always be a weak link in security—they want to be helpful and are willing to trust others. Helpfulness and trust are the number one and number two causes for security breaches. Thus, a policy-based prohibition against revealing

passwords will foil more security attacks than any other protection.

III. OVERVIEW OF EXPERIMENTS

Back Orifice was downloaded and was added to a security program according to Back Orifice directions. The security program was installed according to instructions. All of the typical operations associated with the PKI vendor’s installation continued normally. It is important to note that it was necessary to disable all anti-virus programs normally used in client workstations. This indicates the importance of having a good, strictly enforced anti-virus policy in place.

Each of the above installations depends on unquestioned confidence in the person with access to private keys. Software-based systems store private keys in the computer or re-compute them each time they are needed. Security systems essentially assume that the computer that is being secured is under the control of the person who is requesting access.

With the advent of Trojan programs (such as Back Orifice 2000 and SubSeven) it has become easier for hackers to infiltrate target systems and either deposit programs for later use or extract information from the target system. An investigation was undertaken to see if a hacker could extract the PKI credentials stored on the hard drive of a compromised system and the password for the credentials. In a second experiment, the credentials were stored on a smart card and the tests were repeated. The results of the experiments are discussed below.

The experiments used an isolated LAN environment, with computers running various operating systems. All anti-virus and desktop firewall software was disabled for the tests.



Experiment 1: Attack on Microsoft Windows™ 95 computer without a smart card

- The attack machine is a Windows 95 machine running the Back Orifice 2000 (BO2K) client.
- The victim machine is a Windows 95 machine running a BO2K server.
- The attack machine initiates a connection through BO2K with the victim machine and begins logging keystrokes.
- The victim performs a “Create PKI Credentials” command, which involves establishing a user name and password.
- The attack machine then stops logging keystrokes and views the log file. From the log file, the attacker can capture both the user name and password.
- The attacker uses Back Orifice to search for and retrieve all files that contain the captured user name as well as the pki.ini file. The required files (pkivendor credentials name extension, key files, etc.) are placed in the “PKI Credentials” folder on the attacker’s machine and the attacker can then log in using the captured credentials and password.

Experiment 2: Attack on Microsoft Windows™ 2000 Professional computer without a smart card

- The attack machine is a Windows 95 machine running SubSeven version 2.2.
- The victim machine is a Windows 2000 Professional machine running a SubSeven v. 2.2 server.
- The attack machine initiates a connection through SubSeven with the victim machine and begins logging keystrokes.
- The victim performs a “Create PKI Credentials” command, which involves establishing a user name and password.
- The attack machine then stops logging keystrokes and views the log file. From the

log file, the Attacker can capture both the user name and password.

- The attacker uses SubSeven to search for and retrieve all files that contain the captured user name as well as the pkivendor.ini file. The required files (PKI credentials files, key files, etc.) are placed in the “PKI Credentials” folder on the attacker’s machine and the attacker can then log in using the captured credentials and password.

Experiment 3: Attack on Microsoft Windows™ 2000 Professional Machine with a smart card

- The attack machine is a Windows 95 machine running SubSeven version 2.2.
- The victim machine is a Windows 2000 Professional machine running SubSeven v. 2.2 server. The machine has an external smart card reader running a certificate and token manager.
- The attack machine initiates a connection through SubSeven with the victim machine and begins logging keystrokes.
- The victim performs a “Create Credentials” command, which involves establishing a user name and password.
- The attack machine then stops logging keystrokes and views the log file. From the log file, the attacker can capture both the user name and password.
- **Attempt:** Try to remove the PKI credentials file from the smart card. The attacker knows that a smart card is being used because the PKI credentials file is not on the user’s hard drive and the certificate and token manager software can be found using SubSeven. However, there is no obvious way to manipulate the certificate and token manager software using SubSeven. Only with more powerful backdoor software like



pcAnywhere™ could this be achieved. Specifically, the attacker would have to use the GUI on the user's machine to export the PKI credential file to the hard drive. Most backdoor software like BO2K and SubSeven cannot perform GUI manipulation.

- **Result:** From the attack machine, the smart card cannot be seen or manipulated.
- **Attempt:** Try to capture the PKI credentials file when it is being sent to the PKI Certificate Authority. From the attack machine, install Snort (network sniffer) to run in silent mode on the victim machine. Configure Snort to perform verbose logging of all network activity. When victim machine logs into the PKI CA, stop logging. Analyze log files for signs of the PKI credentials file.
- **Result:** No signs of the PKI credentials file can be identified in the Snort log files.
- **Attempt:** Sabotage PKI software and force PKI credentials to be stored on the hard drive. The attacker creates and stores a copy of the victim's PKI vendor's .ini file. Then the attacker uninstalls the certificate and token manager and replaces the PKI vendor's .ini file. In this way, the PKI software still believes that the certificate and token manager is installed.
- **Result:** The PKI client still attempts to write to the card and, though it does not register any error, it does not write to the victim's hard drive. The system succeeded in thwarting the attack.

IV. CONCLUDING THOUGHTS

Remote control programs, either legitimate or rogue, offer considerable access to a user's computer. In the case of an administrator, this is useful. In the

case of a hacker, it is completely unacceptable. An attacker can easily gain access to a legitimate and authenticated session if credentials can be stolen. As a result, new rules should be put in place and written into policy. Not all of the following recommendations can be implemented with products available today; however they are technically possible and offer potential for products and/or services.

Anti-virus and firewall protection:

Always use a good anti-virus program and update it often. This is the best protection against rogue Trojan programs operating in your users' computers. A firewall also gives protection against Trojan programs, and, depending on the product, can alert the user immediately if an application is trying to communicate with an external system.

Password management:

Use difficult-to-guess passphrases and change them often. This is a very good rule, but impossible to manage with only policy. Password management programs that can help the user with proper passwords are a valuable addition to a desktop security environment. Such a program must also keep passwords from being stolen in mass.

It is important to remember that a password management program does not eliminate the need to change system passwords, and more importantly the root passwords, often. (The root password allows access to all the others.) This is particularly true if the root password is subject to attack. A particularly strong system would incorporate a smart card to hold password data, with the smart card's access control password not available to an attacker. Personal identification number (PIN) readers that deliver a password directly to the smart card without sending it through the operating system are available for such applications.



Session management:

Sessions should be established with strong authentication techniques and should require periodic re-authentication. A timeout of a few minutes that resets with each access should be adequate to allow a user essentially free access to a resource while limiting the window of opportunity to an attacker.

Session theft:

The review above does not address the issue of session theft or the potential of establishing a session if a password is stolen. If a smart card is used to store and use public key credentials, the credentials

cannot be stolen and used elsewhere. However, one must consider the possibility of an attacker using a stolen password to activate a public key-based session. Even with a smart card-based system, a new session can be established if the smart card password is stolen and the card is left in the reader.

Smart card security:

As demonstrated in the experiments outlined in this paper, smart cards combined with a public key security system do provide a greater level of protection against control software attacks than a standalone public key system.

The Smart Card Alliance Digital Security Initiative work group would like to thank Datakey, Inc. and Gordon Morrison, Senior Security Technical Specialist, Bank of Nova Scotia for their assistance in conducting the experiments and preparing this document. For more information, contact the Alliance at www.smartcardalliance.org; info@smartcardalliance.org.

