Microsoft

Microsoft Windows 2000 Server

Operating System

Smart Card Logon

White Paper

Abstract

The Microsoft® Windows® 2000 operating system introduces smart card authentication as an alternative to passwords to achieve strong network authentication. A smart card can be used to authenticate to a Windows 2000 domain in three ways. The first is *interactive logon* involving Active Directory, the Kerberos version 5 protocol, and public key certificates. The second is *remote logon* that uses a public key certificate with the Extensible Authentication Protocol (EAP) and Transport Layer Security (TLS) to authenticate a remote user to an account stored in Active Directory. The third is *client authentication* where a user is authenticated using a public key certificate mapped to an account stored in Active Directory. By integrating public key technologies and smart cards with Windows 2000, Microsoft is helping customers to increase their level of security at a time when the convergence of the enterprise and Web computing models is driving companies to open up their corporate networks to stay competitive.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

0699

CONTENTS

INTRODUCTION

Smart cards are a key component of the public-key infrastructure (PKI) that Microsoft has integrated with the Windows® 2000 operating system. Smart cards enhance software-only solutions such as interactive logon, client authentication, and remote logon. Smart cards provide:

- Tamper-resistant storage for protecting private keys and other forms of personal information.
- Isolation of security-critical computations involving the private key from other parts of the system that do not have a “need to know.”
- Portability of credentials and other private information between computers at work, home, or on the road.

What is a Smart Card?

The term *smart card* has been used to describe a class of credit card-sized devices with varying capabilities: stored-value cards, contact-less cards, and integrated circuit cards (ICC). All of these cards differ in functionality from each other and from the more familiar magnetic-stripe cards used by standard credit, debit, and ATM cards. It is the ICC that is of most interest to the personal computer, and Windows 2000, because it is able to perform more sophisticated operations such as digital signature and key exchange.

A smart card is essentially a miniature computer, embedded in plastic in the form of a credit card, with limited storage and processing capability. The circuitry in a smart card derives power from a smart card reader after the card is inserted into the reader. Data communication between a smart card and an application running on a computer is performed over a half-duplex serial interface managed by the smart card reader and its associated device driver. Smart card readers are available in a variety of form-factors and can be connected to a computer using an RS-232, PCMCIA or USB interface.

What is Cryptography?

Cryptography is the science of protecting data or messages. Many cryptographic algorithms mathematically combine input plaintext data and an encryption key to generate encrypted data referred to as *ciphertext*. With a good cryptographic algorithm, it is computationally infeasible to reverse the encryption process and derive the plaintext data from the ciphertext. In order to decrypt the ciphertext some additional data, a decryption key, is needed to perform the transformation.

In traditional secret (or symmetric) key cryptography, encryption and decryption keys are identical and must be shared by multiple parties. Parties wishing to communicate with secret-key cryptography must securely exchange the encryption/decryption keys before they can exchange encrypted data.

Public Key Cryptography

In contrast, the fundamental property of public-key (PK) cryptography is that the encryption and decryption keys are different. Encryption with a public key is a one-way function; plaintext turns into ciphertext, but the encryption key is unrelated to the decryption process. A decryption operation requires a private key (related, but not identical, to the encryption key) to transform the ciphertext back into plaintext. Therefore, every public key user has a pair of keys consisting of a public key and a private key. By making the public key available to anyone, it is possible to enable someone to send encrypted data to another person (or persons) that can only be decrypted by the recipient using the private key. Separation of the public key from the private key has enabled new applications of cryptography such as digital signature, key agreement, and distributed authentication.

Authentication typically requires some type of challenge-response between authenticating parties. Public key cryptography provides a means by which a challenge-response can be accomplished between two parties who have never met because the public key and private key are distinct and separate. Separation of the private and public key enables distributed authentication because it does not require that the parties share a key *a priori*. Likewise, public key cryptography can also be used to generate a shared key without the parties having to meet in secret.

In addition, data can be transformed using a private key in such a way that recipients can verify the data originated from a specific sender and that the data has not been tampered with while in transit. This is known as digital signature and is quite powerful. Digital signatures are themselves just data so they can be transported along with the signed data that they protect*.* A digital signature ensures identity because only the owner of the private key could have signed the data, and integrity because modification of the data after it is signed invalidates the signature. Anyone can verify a signature because the public key can be published in a directory as part of a certificate.

PUBLIC KEY CONCEPTS

The following concepts are important to understanding how smart card logon works in Windows 2000. For more information on the public key technologies integrated with Windows 2000, read the Windows 2000 Public Key Infrastructure white paper.

What is a Public Key Infrastructure?

A public key infrastructure (PKI) is the set of components that manages certificates and keys used by encryption and digital signature services. A good PKI must provide services for cryptographic operations, certificate enrollment and renewal, certificate distribution and validation, certificate revocation, plus administrative tools and services for managing all of the above. Major components of the Windows 2000 PKI include the certification authority (CA) service, the Microsoft CryptoAPI for managing certificates and keys, and the policy infrastructure used to configure operational parameters that determine trust relationships and application behavior. A directory can also be considered a PKI component because it can store information such as CA location, certificates, and certificate revocation lists (CRLs).

Certificate

Certificates provide a mechanism for gaining confidence in the relationship between a public key and the entity that owns the corresponding private key. The most common form of certificates in use today is based on the ITU-T X.509 standard. The Internet Engineering Task Force (IETF) Request For Comment (RFC) 2459 profiles the X.509 version 3 certificate and the X.509 version 2 certificate revocation list (CRL) both of which are supported in Windows 2000.

A certificate can be thought of as similar to a driver’s license. A driver’s license is accepted by numerous businesses as a form of identification because the license issuer (a government institution) is accepted by the community as trustworthy. Because businesses understand the process by which someone can obtain a driver’s license, they can trust that the issuer verified the identity of the individual to whom the licensed was issued. Therefore, the driver’s license can be accepted as a valid form of identification.

Certification Authority

A certification authority issues certificates to requesters based on a set of established criteria. A CA acts as a guarantor of the binding between the subject public key and the subject identity information that is contained in the certificates it issues. The criteria that a CA uses when processing a request should be made public in a certificate practice statement enabling users of certificates to make informed trust decisions.

The Windows 2000 public key infrastructure assumes a hierarchical CA model chosen for its scalability, ease of administration, and consistency with a growing number of commercial and third-party products. In its simplest form a CA hierarchy consists of a single CA, though, in general, a hierarchy contains multiple CAs with clearly defined parent-child relationships. A child CA is referred to as a subordinate CA. There is no requirement that all subordinate CAs within a company share a common top-level parent CA.

The CA at the top of a hierarchy is generally referred to as a *root* CA whose certificate is self-signed. A self-signed certificate is a certificate whose subject name and issuer name are the same and whose public key can be directly used to verify the signature attached to the certificate.

A CA that issues end-entity certificates is typically called an *issuing* CA. An *intermediate* CA refers to a CA that is not a root CA, but that only certifies other CAs in a hierarchy. A hierarchy with only a root CA does not contain an intermediate CA. A two-level hierarchy with a root CA and an issuer CA also does not contain an intermediate CA. Hierarchies with more than two levels of CAs, by definition, contain at least one intermediate CA.

Windows 2000 supports two types of CA services: enterprise or stand-alone. The primary difference between the two CA services is in how certificates are issued. The stand-alone CA will issue certificates without authenticating the requestor and usually is configured to require a CA administrator to approve requests based on some out-of-band authentication. The enterprise CA requires the existence of a Windows 2000 domain and authenticates the requestor based on his or her domain logon information. The enterprise CA uses certificate templates to distinguish different types of certificates based on intended usage(s). Users may enroll for different types of certificates based on their access rights within a domain.

Certificate Revocation List

The CA is responsible for publishing status information about any certificates it has revoked so that applications can check whether a given certificate and key-pair is still trustworthy. Revoking a certificate means that the CA is withdrawing its statement about the allowed usage of the key-pair prior to the certificate’s normal expiration. In order to revoke a certificate a CA will publish a list of serial numbers identifying the revoked certificates. This list is called a certificate revocation list and is signed by the CA to allow an application to trust the information it contains based on the signature. Like certificates, certificate revocation lists have validity periods though their lifetimes are typically much shorter than certificates. Windows 2000 supports both automatic and manual publishing of CRLs with configurable lifetimes and to multiple locations.

Certificate Enrollment

Certificate enrollment is the procedure used to request and receive a certificate from a CA. The certificate request provides identity information to the CA that subsequently becomes part of the issued certificate. The CA processes the request based on a set of criteria that may require some out-of-band authentication. If the request is successfully processed, the CA then issues the certificate to the user, computer or another CA. In the case of a root CA, the CA issues itself a certificate. The Windows 2000 PKI supports certificate enrollment to the CA service or to third-party CAs like VeriSign. Enrollment support is implemented in a transport-independent manner and is based on use of industry standard PKCS-10 certificate request messages and PKCS-7 responses containing the resulting certificate or certificate chain. In Windows 2000 certificates that support RSA keys and signatures, Digital Signature Algorithm (DSA) keys and signatures, and Diffie-Hellman keys are supported. However, enrollment for Diffie-Hellman certificates is not supported in Windows 2000 due to the absence of an established standard on which to base an implementation.

Certificate Stores

Cryptographic keys and their associated certificates are stored and managed by the CryptoAPI subsystem. Keys are managed by Cryptographic Service Providers (CSPs) and certificates are managed by the CryptoAPI certificate stores.

The certificate stores are repositories for certificates and their associated properties. By convention, the PKI defines five standard certificate stores:

- **Personal**—contains a user's or computer's certificates for which the associated private key is accessible through CryptoAPI.
- **CA**—contains issuer or intermediate CA certificates used in building certificate chains.
- **Enterprise Trust**—contains Certificate Trust Lists . These are an alternate mechanism that allows an administrator to specify a collection of trusted CAs that must verify to a self-signed CA certificate in the trusted root store.
- **Trusted Root**—contains only self-signed CA certificates that are trust points in the PKI.
- **UserDS**—provides a logical view of a certificate repository that is stored in the Active Directory. For example, the Windows 2000 CA can publish user and machine certificates to the userCertificate property of the user or computer object for use by other users or services.

Certificate Chain Validation

The certificate verification code in CryptoAPI attempts to build a certification path (that is, certificate chain) from the end-entity certificate (for example, a user or computer certificate) up to a CA root as follows:

1. If a certificate chain is not delivered as part of the protocol, CryptoAPI uses the Authority Key Identifier (AKI) field information, if present, to find the parent (issuer) certificate(s) in the system certificate stores. If AKI contains issuer and serial number information, CryptoAPI will look for a specific parent certificate; otherwise it will use the key identifier value to find matching parent certificate(s).
2. If CryptoAPI does not find a parent certificate in the certificate chain delivered by the protocol, or in the system stores based on AKI, CryptoAPI will then look at the Authority Info Access (AIA) field in the certificate. CryptoAPI will use information in AIA, if present, to retrieve the parent certificate(s) from the location(s) specified (for example, HTTP, LDAP).
3. If neither of the above steps yields a parent certificate, then CryptoAPI will use the issuer name information in a certificate to find a parent in the system certificate stores.

This process is repeated for each certificate in a chain terminating in a self-signed certificate in the trusted root store for the user or the computer.

ACTIVE DIRECTORY AND KERBEROS CONCEPTS

The following concepts are important to understanding how smart card logon works in Windows 2000. For more information on Microsoft's implementation of the Kerberos version 5 protocol, read the Windows 2000 Kerberos Authentication white paper, available from http://www.microsoft.com/windows/server/Technical/. For more information on Active Directory read the Windows 2000 Active Directory white paper, available from http://www.microsoft.com/windows/server/Technical/.

Kerberos

The Kerberos authentication protocol provides a mechanism for mutual authentication between a client and a server, or between different servers. One benefit of mutual authentication using the Kerberos version 5 protocol is that trust between security authorities for Windows 2000 domains is by default two-way and transitive.

The Kerberos version 5 protocol relies heavily on an authentication technique involving shared secrets. The basic concept is quite simple: If a secret is known by only two people, then either person can verify the identity of the other by confirming that the other person knows the secret. Rather than sharing a password, communication partners share a cryptographic key, and they use knowledge of this key to verify one another's identity. For the technique to work, the shared key must be *symmetric*—a single key must be capable of both encryption and decryption. One party proves knowledge of the key by encrypting a piece of information, the other by decrypting it.

PKINIT

There is an extension to the Kerberos version 5 protocol proposed by the IETF called PKINIT that allows for the use of a public key certificate in place of a password during the initial authentication. The PKINIT extension is the basis for smart card logon support in Windows 2000 and does not change the requirement for a long-term symmetric key. Rather, the public key in the certificate is used to encrypt a symmetric key returned as a result of a successful authentication that then must be decrypted using the associated private key stored on the smart card.

Key Distribution Center

Windows 2000 implements the Key Distribution Center (KDC) as a domain service and uses the domain's Active Directory as its account database. The KDC is a single process comprised of two services: the Authentication Service (AS) and the Ticket-Granting Service (TGS). The AS issues Ticket Granting Tickets (TGTs) to authenticated principals (that is, users, machines, services) for admission to the TGS. The TGS issues tickets for admission to other services in the domain or to a TGS in another trusted domain. Each domain controller has a KDC that runs in the process space of the Local Security Authority (LSA). Any domain controller can accept authentication requests and ticket-granting requests addressed to the domain's KDC.

Active Directory

The Active Directory is primarily a *namespace* that is a bounded area in which a given name can be resolved. Name resolution is the process of translating a name into some object or information that the name represents. A telephone directory forms a namespace in which the names of telephone subscribers can be resolved to telephone numbers. The Active Directory forms a namespace in which the name of an object in the directory can be resolved to the object itself.

An *object* is a distinct, named set of *attributes* that represents something concrete such as a user, a printer, or an application. The attributes hold data describing the subject that is identified by the directory object. Attributes of a user might include the user’s common name, a certificate, or an e-mail address. All Active Directory objects are protected by an Access Control List (ACL). ACLs determine who can see the object and what actions each user can perform on the object. The existence of an object is never revealed to a user who is not allowed to see it.

Domains, Forests and Trust

A *domain* is a single security boundary in Windows 2000. Active Directory is comprised of one or more domains. On a stand-alone computer, the domain is the computer itself. A domain can span more than one physical location. Every domain has its own security policies and security relationships with other domains. When multiple domains are connected by trust relationships and share a common schema, configuration, and global catalog, you have a *domain tree*. Multiple domain trees can be connected together into a *forest.*

A *forest* is a set of one or more trees that do not form a contiguous namespace. All trees in a forest share a common schema, configuration, and Global Catalog. All trees in a given forest trust each other via transitive, hierarchical Kerberos trust relationships. Unlike a tree, a forest does not need a distinct name. A forest exists as a set of cross-reference objects and Kerberos trust relationships known to the member trees. Trees in a forest form a hierarchy for the purposes of Kerberos trust; the tree name at the root of the trust tree can be used to refer to a given forest

Kerberos uses the Active Directory as its account database from which it obtains information about security principals. When a domain is joined to a Windows 2000 domain tree, a Kerberos trust relationship is automatically established between the joined-from domain and its parent in the tree. Kerberos trust is transitive, so no additional trust relationships are required among tree members. The trust hierarchy is stored in *cross-reference* objects in the directory.

Each domain controller keeps a writeable copy of the directory so that accounts can be created, passwords reset, and group membership modified on any domain controller. Changes made to one replica of the directory are automatically propagated to other replicas. Windows 2000 does not, however, implement the Kerberos replication protocol. Instead it replicates the information store for Active Directory using a multi-master protocol over a secure channel established between replication partners.

HOW SMART CARD AUTHENTICATION WORKS

A smart card can be used to authenticate to a Windows 2000 domain in three ways. The first is *interactive logon* involving Active Directory, the Kerberos version 5 protocol, and public key certificates. The second is *client authentication* where a user is authenticated using a public key certificate that matches an account stored in Active Directory. The third is *remote logon* that uses a public key certificate with the Extensible Authentication Protocol (EAP) and Transport Layer Security (TLS) to authenticate a remote user to an account stored in Active Directory.

Interactive Logon

Interactive Logon using a smart card begins when a user inserts a smart card into a smart card reader that signals the Windows 2000 operating system to prompt for a Personal Identification Number (PIN) instead of a username, domain name and password. The card insertion event is equivalent to the familiar Ctrl-Alt-Del secure attention sequence used to initiate a password-based logon. However, the PIN the user provides to the logon dialog is used to authenticate only to the smart card and not to the domain itself. A public key certificate stored on the smart card is used to authenticate to the domain using the Kerberos version 5 protocol and its associated PKINIT extension.

Logon Request

After a user inputs a PIN to the logon dialog, the operating system begins a sequence of actions to determine whether the user can be identified and authenticated based on credential information the user has provided (PIN and smart card). The logon request first goes to the LSA that subsequently forwards it to the Kerberos authentication package running on the client. The Kerberos package sends an authentication service (AS) request to the KDC service running on a domain controller to request authentication and a Ticket Granting Ticket (TGT). As part of the AS request, the client-side Kerberos package includes the user’s X.509 version 3 certificate, retrieved from the smart card, in the pre-authentication data fields of the AS request. An *authenticator,* included in the pre-authentication data fields, is digitally signed by the user’s private key so that the KDC can verify the AS request originated from the owner of the accompanying certificate.

Certificate Verification

Before the KDC can satisfy the AS request, it must first verify the certification path of the user’s certificate to ensure that it can be trusted. The KDC uses CryptoAPI to build a certification path from the user’s certificate to a root CA certificate residing in the system root store. If the KDC fails to build a valid certificate chain for any reason (that is, root certificate is not trusted, cannot find parent certificates, revocation status cannot be determined) the KDC will return an error and fail the request.

The KDC must also verify that the issuing CA is authorized to issue certificates whose name information can be used as a basis for authentication within the domain. In Windows 2000, the issuing CA must be an enterprise CA published in the Active Directory in order to be trusted for authentication. This is required in order to prevent a rogue CA, trusted under one CA hierarchy, from issuing certificates into another domain's namespace. While this type of attack is extremely difficult and depends on the rogue CA obtaining issuance rights from a legitimate parent CA, the solution to require an enterprise CA published in Active Directory was implemented to remove the potential for an attack.

Digital Signature Verification

Upon successful verification of the user's certificate, the KDC then uses CryptoAPI to verify the digital signature on the authenticator that was included as signed data in the pre-authentication data fields. The signature verification is done using the public key from the user's certificate to prove that the request originated from the owner of the public key. Because the certificate was retrieved from the smart card and the authenticator was signed using the private key stored in the smart card, the digital signature must be legitimate because the user had to authenticate to the smart card in order for the private key to sign the authenticator. After verifying the signature, the KDC service must then validate the timestamp in the authenticator to ensure the request is not a replay attack.

Account Lookup and TGT

Upon verifying that a user is who they say they are and that the certificate can be used to authenticate to the domain, the KDC service then queries the domain's directory for account information. The KDC service retrieves user account information from Active Directory based on the User Principal Name (UPN) specified in the Subject Alternative Name field in the user's public key certificate. The account information that the KDC retrieves from the directory is used to construct a TGT. The TGT will include the user's Security ID (SID), the SIDs for any domain groups to which the user belongs, and potentially the SIDs for any universal groups in which the user is a member (in a multi-domain environment). The list of SIDs is included in the TGT's authorization data fields.

The KDC encrypts the TGT using a random key generated specifically for this purpose. The random key is itself encrypted using the public key from the user's certificate and the encrypted key is included in the pre-authentication data field of the KDC's response. The KDC signs the reply using its private key so that the client can verify the reply is from a trusted KDC. The client verifies the KDC's signature by first building a certification path from the KDC's certificate to a trusted root CA and then using the KDC's public key to verify the reply signature. The KDC also signs the TGT's authorization data using the server's key that is then signed with the KDC's secret key so that a rogue service cannot alter the authorization data after the TGT has been issued. The client, upon receipt of the KDC's response, will extract the encrypted random key, decrypt it, and use the resulting key to decrypt the TGT. Once in possession of the TGT, the standard Kerberos version 5 protocol is used to request tickets from the TGS for other domain resources.

It should be noted that supplemental credentials are generated as part of a Kerberos logon so that access to down-level servers such as those running the

Windows NT® 4.0operating system will still work. This is true even if the user has never used a password on the computer. When an account is created a one-way function (is generated and added as an attribute of the account for use as a supplemental credential for down-level authentication.

Offline Logon

When a user is disconnected from the network or the domain controller is unreachable due to failure somewhere along the network path, a user must still be able to logon to his or her computer. With passwords this capability is supported by comparing the hashed password stored by the LSA with a hash of the credential that the user supplied to the GINA during logon. If the hashes are the same then the user can be authenticated to the local machine.

In the smart card case, offline logon requires the user's private key to decrypt supplemental credentials originally encrypted using the user's public key. If the user has multiple smart cards then the supplemental credentials must be encrypted and referenced based on the hash of the certificate to ensure that the user can perform an offline logon regardless of what card is used.

Client Authentication

Because smart card support is integrated with CryptoAPI, the Secure Sockets Layer (SSL) and the Transport Layer Security (TLS) protocols do not need to know anything about a smart card in order for smart cards to work with these protocols. The role of the smart card in client authentication is to sign a part of the protocol during the initial SSL session negotiation. Because the private key corresponding to the public key certificate is stored on the smart card, the method of authentication is stronger because use of the private key requires the holder of the card to authenticate to the card and to the domain. In addition, the private key operation performed during the initial session negotiation is performed on the card such that the private key is never exposed to the host computer or network.

Mutual Authentication

Both the SSL 3.0 and TLS 1.0 protocols support mutual authentication meaning the client can authenticate the server and the server can authenticate the client. Server authentication is when the client authenticates the server by verifying the cryptographic signatures on the server's certificate, and any intermediate CA certificates, to a root CA certificate residing in the trusted root store on the client. Authentication of the client by the server is accomplished in the reverse manner as server authentication. The server verifies the cryptographic signatures on the client's certificate, and any intermediate CA certificates, to a root CA installed in the trusted root store on the server. Once the identity of the client is verified, the server needs to establish a security context with appropriate authorization that determines what resources the client is allowed to use on the server.

Authorization

Public-key client authentication in Windows 2000 uses information in the client's certificate to map to domain access control information. Windows 2000 implements a security service that uses certificate information to map to accounts stored in the Active Directory for the purpose of determining access rights for the authenticated client. This directory operation can be performed based on the UPN in the certificate or by searching the directory for an account that matches properties, issuer or issuer and subject, in the client certificate.

When an SSL or TLS session is established, the secure channel provider first attempts to find a user account in the domain's directory based on the UPN in the certificate. The UPN is included in the certificate in the Alternative Subject Name field and specifies the exact user account name (prefix) and the domain name (suffix) where the account is located. Just as in Kerberos-based smart card logon, the issuing CA must be authorized to issue certificates for the domain whose name information can be trusted for identification and authentication.

If there is no UPN match, or the issuing CA is not authorized to issue certificates for domain authentication, the provider then attempts to query the directory to find an account whose Alternate Security Identities attribute contains an explicit mapping to the client certificate. An explicit mapping between a certificate and a user account requires an administrator to perform an action to create the mapping. The mapping to the client certificate can be based on either the issuer name or the issuer and subject names in the certificate. An account can have one or more certificates associated with it to facilitate using one account for multiple external users. A certificate cannot be mapped to multiple accounts stored in the domain's directory. Authentication will fail if a certificate is mapped to more than one account.

Remote Access

Support for extensible authentication for remote users is supported in Windows 2000 by the remote access service (RAS). The remote access server supports EAP to allow vendor-supplied authentication modules to be added support for a variety of authentication methods ranging from smart card to one-time passwords to biometrics. Windows 2000 includes a built-in module for smart cards to enable strong authentication for remote users.

A remote logon actually involves two separate authentications: one to the RAS server and the other to the network. The first authentication results in the RAS server being authenticated by the client and establishment of a connection between the client and server. As a result of this connection, some RAS-specific policies and account attributes are applied to the client. Account attributes are applied on a per user basis and include properties such as access rights, callback options, static routes, and so forth. RAS policies are specific to how the server should interact with the client and are rules-based. This allows for management of users by matching on the properties of the connection, group membership, profile, and so forth.

The second authentication is to the domain and uses EAP over TLS instead of Kerberos or standard SSL as the authentication protocol. The authentication to the domain over EAP/TLS is very similar to client authentication using SSL described previously, except that the public key certificate must contain a UPN that matches an account stored in the domain's Active Directory.

Local versus Domain Logon

There is a subtle difference between logging onto a domain interactively using a dial-up connection selected from the logon dialog and logging into the computer locally and then using a dialer to establish a network connection to the RAS server. The former causes domain policy to be applied to the user and client and the latter does not. This subtle difference matters because if a user is logging in remotely for the very first time, the computer will not have domain policy unless the computer had been pre-configured as a member of the target domain. Without domain policy, the client will not be able to authenticate the RAS server causing the initial authentication to fail.

DEPLOYING SMART CARDS

The value to a corporation deploying smart cards is increased network security through stronger authentication methods. Windows 2000 makes it possible to deploy strong authentication using smart cards by leveraging operating system features such as Kerberos, Active Directory, and the variety of administrative tools used to manage a public key infrastructure. Windows 2000 Professional, Windows 2000 Server and Windows 2000 Advanced Server have integrated smart card and public key technologies to enable corporations to take immediate advantage of them in a cost-effective manner without the need to outsource certificate management or purchase expensive and proprietary application plug-ins.

For any business considering deployment of smart cards, there are some basic questions that must be answered first. While this paper is not intended to serve as a deployment guide, here are four simple questions to consider when planning a smart card deployment:

- Which user populations should be required to use smart cards?
- What policies and procedures should be established to manage the cards and certificates?
- How should smart cards be issued to users?
- What smart card hardware is available and compatible with Windows 2000?

Who Should Use Smart Cards?

It is recommended that users who do not perform advanced tasks such as joining computers to domains or promoting servers to domain controllers be issued smart cards and not passwords. This category of user should represent a significant portion of a company’s employee population. These users could be professional workers, suppliers, contractors, or anyone else who is not trusted to administer a computer or the network.

Windows 2000 supports organizing users based on their roles within a domain and defines three distinct categories: Administrator, Power User and User.

Administrators are all-powerful. The default Windows 2000 security settings do not restrict Administrative access to any registry or file system object. Administrators can perform any and all functions supported by the operating system. Any right that the Administrator does not have by default, they can grant to themselves.

Power Users are less powerful than administrators but are still able to install applications, configure system settings, and so forth. The default Windows 2000 security settings for Power Users are backward compatible with the default security settings for users in the Windows NT 4.0 operating system.

Users are the opposite of administrators. The default security settings are designed to prohibit Users from compromising the integrity of the operating system and installed applications if Windows 2000 is clean-installed on an NTFS partition. Users cannot modify machine-wide registry settings, operating system files, or program files. Users cannot install applications that can be run by other Users (preventing

Trojan horses). Users cannot access other users' private data.

Members of the Users group should use smart cards for authentication because their role within the enterprise should not require them to perform advanced tasks. This group of users should represent a significant portion within any corporation, making smart card deployment a worthwhile investment because managing passwords for this population will no longer be necessary. Because the smart card can also be used to authenticate over SSL, in addition to interactive and remote access logon, the value of deploying smart cards is high because large populations of users can migrated away from passwords that have proven difficult to manage.

However, it is not feasible to recommend that Power Users or Administrators use smart cards exclusively since they may have a need to perform operations that involve a secondary authentication requiring a username, domain name and password. In particular, smart card-based authentication cannot be used in the following scenarios:

- The user is required to join his or her computer to a domain.
- The user must perform administrative tasks such as promote a server to be a domain controller.
- The user needs to configure a network connection for remote access.

What Policies Are Needed?

Public key security policy is one aspect of security policy and is integrated with the Windows 2000 policy management infrastructure to provide a consistent model for administering public key policy alongside policies for other services. There are several types of policy that can be set to control the use of smart cards within a Windows 2000 domain.

Smart Card Required

Windows 2000 supports a per user account policy, *smart card required for interactive logon*, that requires a smart card to effect an interactive logon. What this means is that once the policy is set on an account, the user cannot use a password to log on to the account, interactively or from a command-line. The policy applies to interactive and network logon only, but not to remote access logon which uses a different policy configured on the remote access server. While setting the smart card required for interactive logon policy on an account is not recommended for every user in an enterprise, it should be set for those users who are members of the Users group that are using smart cards to log on to a Windows 2000 domain.

The *smart card required for interactive logon* policy is not recommended for the following scenarios:

- The user is required to join his or her computer to a domain.
- The user must perform administrative tasks such as promote a server to be a domain controller.
- The user needs to configure a network connection for remote access.

In each of these scenarios, the user will need to provide a username, domain name and password because these tasks do not support using public key-based authentication. In future releases, Windows 2000 will support the use of public key certificates for authentication in those scenarios.

On Smart Card Removal

When a user walks away from a computer with an active logon session, he or she is expected to either logoff or lock the computer. If the user fails to secure the computer, the screensaver program could lock the computer if it is configured to do so. Otherwise the computer is open for an attack by a malicious insider who can do various things such as send unfriendly email as the logged on user.

The *on smart card removal* policy is a local computer policy administered on a per machine basis on not on a per user account basis like the smart card required for interactive logon policy. The decision to set the *on smart card removal* policy depends on the needs of the corporation and how users interact with computers. In situations where users interact with computers in an open floor or kiosk environment, the use of such a policy is highly recommended. In situations where users have a dedicated computer or multiple computers that only they use, it may not be necessary to set this policy if other means of locking the computer are enabled such as a screensaver program. As with many security policy decisions, the trade-off is increased security versus usability.

Left Card at Home

One issue that must be considered when deploying smart cards is what happens when an employee forgets his or her card. Being able to gain access to a building using a temporary badge is quite different than needing to logon to a computer to perform one’s job function. There are several options available to handle this situation. One is to issue a temporary smart card with a certificate that has a short expiration such as one day. Another is to not set the smart card required for interactive logon policy and instead set a long password that is only shared with the employee when the situation arises, then subsequently reset.

Personal Identification Numbers

While passwords are inherently weak, and are made weaker by users choosing easy-to-remember pass-phrases, smart card PINs do not have to follow the same rules as "strong passwords" because the cards are not open to classic dictionary attacks. An easy-to-remember PIN is not a problem because a smart card will lock when too many wrong PIN inputs are attempted in a row. Since the PIN itself is never transmitted over-the-network in any form, a replay attack is extremely difficult because it also requires possession of the physical card as opposed to a sniff of a network packet as in a password-based attack.

Much has been made of PIN management and the need to periodically change the PIN for the smart card, like what is done with password policy. Frequent PIN changes are really not necessary because of the design of the smart card and the type of attack that can be mounted against it (as pointed out above). However, there is a usability issue with regards to how or when a user can change his or her PIN because this capability is only exposed to the user when a private key operation is being performed. This is due to the lack of standards for how PINs are managed across card operating systems forcing PIN management to be done at the CSP layer and not at the operating system layer. Hence, there is no change PIN functionality available through the standard desktop logon interface like there is for passwords.

How Should Smart Cards Be Issued?

Because a smart card is a trusted ID like an employee badge, most corporations will want to integrate smart cards with the employee badge rather than issue a second ID which must be managed separately. Obtaining a badge typically requires a visit to a security office where the employee must prove identity and then have his or her picture taken to create the badge used to gain access to a building or facility. The only delta to the above procedure is the badge would now also contain a certificate issued to the employee by the corporation.

In this scenario, the security office acts as a registration authority by performing an enroll-on-behalf operation to request and then install the certificate onto the employee’s smart card-based badge. The certification authority service that ships with Windows 2000 Server and Windows 2000 Advanced Server supports this functionality as part of its Web enrollment interface.

No matter how a corporation chooses to deploy smart cards, the decision to integrate with the employee badge or not should be based on business needs that must balance the need for increased security with overall usability across the employee population.

Smart Card Enrollment Station

A smart card enrollment station is included as part of the enterprise CA service available with Windows 2000 Server and Windows 2000 Advanced Server. This enrollment station supports the issuance of smart cards from a central location. Like the enterprise CA, the smart card enrollment station uses certificate templates to determine what information to include in a certificate such as intended usage. The default install of an enterprise CA does not enable the smart card certificate templates for issuance; instead, a CA administrator must enable these templates for issuance by an enterprise CA.

For smart card there are two certificate templates of interest: Smart Card Logon and Smart Card User. The Smart Card Logon certificate and Smart Card User certificates are very similar except the Smart Card Logon certificate cannot be used for secure email while the Smart Card User certificate can. Both certificate types have specific extended key usage properties that are used to determine the intended purpose of the certificate. For example, only these two certificate types can be used to log on interactively to a domain because each contains an extension specifying smart card logon.

In order to issue a smart card certificate, a smart card enrollment station must exist somewhere in the corporation to perform enroll-on-behalf operations. This requirement is necessary because, by default, domain users cannot enroll for smart card certificates issued by the Windows 2000 enterprise CA service. Access to the smart card certificates is restricted to domain administrators unless the access permissions on a template has been modified by a domain administrator to allow other user groups the ability to enroll. This is required to prevent an attack where a user leaves his or her workstation without logging off or locking it and someone uses the active logon session to enroll for a smart card certificate as the (unaware) user.

To operate the Windows 2000 smart card enrollment station, someone within the corporation must be authorized to be an enrollment agent. To support this role, the enterprise CA can issue an Enrollment Agent certificate for the explicit purpose of enroll-on-behalf operations. This certificate is the most powerful of all certificates because an employee with an Enrollment Agent certificate has the ability to enroll for smart card certificates for any domain user, including Administrator. It is therefore recommended that the default access permissions on the Enrollment Agent certificate template be set to allow only select employees the ability to enroll for one. In addition, it may be desirable to disable issuance of this certificate type at the CA except when specifically needed or to take the CA offline. By default, the access permissions for the Enrollment Agent certificate is set to domain administrators.

What Hardware Is Available?

Smart Cards

Companies deploying smart cards must not assume a card can store multiple certificates because there is no support to distinguish multiple certificates on a card. Cryptographic smart cards have a limited amount of storage capacity; typically between 2 kilobytes and 8 kilobytes of available storage, with 16 kilobytes coming in the future. Because storage is at a premium and expensive, card vendors often restrict the amount of storage available to a single application in order to support multiple applications or services on the card.

In Windows 2000, smart cards and their associated cryptographic service provider (CSP) are assumed to only support one certificate per card. Windows 2000 will work with any cryptographic smart card that has an associated CryptoAPI CSP. Gemplus and Schlumberger both have CryptoAPI cryptographic service providers that are included with Windows 2000 while companies like Bull and Siemens Nixdorf offer their smart card CSPs separately.

Smart Card Readers

Smart card readers come in a variety of types depending on the application. The majority of smart card readers connect to the PC through an RS-232 port, a Type II

PCMCIA slot, or USB port. Windows 2000 includes native support for several smart card readers from the major vendors that include Bull, Gemplus, Litronic, Rainbow Technologies, Schlumberger, and SCM Microsystems.

To help customers determine which smart cards readers are compatible with the Windows platform, Microsoft has developed a logo program for smart card readers, administered by the Windows Hardware Quality Lab, to ensure computer compatibility and card and reader interoperability. The smart card logo program is similar to what Microsoft has done for many other device types such as sound cards, network cards, and graphics cards. The biggest win for customers is that they can purchase a reader that has the Windows-compatible logo knowing that the device meets all of the requirements for compatibility and interoperability including device driver quality, power management, and Plug and Play. For a list of Windows-compatible devices, go to http://www.microsoft.com/hwtest/hcl and search on Smart Card Reader.

INTEROPERABILITY AND STANDARDS

Interoperability is always bi-directional between vendor products. The Microsoft Windows 2000 PKI is designed to be compliant with the major public key standards. Microsoft continues to support interoperability by testing with a variety of third-party products from Netscape, VeriSign, and Entrust among others.

PKIX

The IETF Internet X.509 Public Key Infrastructure Certificate and CRL Profile specification, RFC 2459, can be found at http://www.ietf.org/rfc/rfc2459.txt.

PKCS

The Public Key Cryptography Standards (PKCS) specifications can be found at http://www.rsa.com/rsalabs/pubs/PKCS.

TLS

Internet drafts from the IETF Transport Layer Security (TLS) working group can be found at http://www.ietf.org/html.charters/tls-charter.html.

Kerberos

The IETF Kerberos Network Authentication Service (V5) specification, RFC 1510, can be found at http://www.ietf.org/rfc/rfc1510.txt.

The IETF draft Public Key Cryptography for Initial Authentication in Kerberos that updates RFC 1510 can be found at http://search.ietf.org/internet-drafts/draft-ietf-cat-kerberos-pk-init-07.txt.

The PC/SC

The Personal Computer/Smart Card specifications can be found at http://www.smartcardsys.com/.

ADDITIONAL REFERENCES

Resources

The latest security information available from Microsoft can be obtained at http://www.microsoft.com/security/.

Information for software developers can be obtained through the Microsoft Developer Network (MSDN) at http://msdn.microsoft.com/.

Documents

U.S. Government

The National Institute of Standards and Technologies (NIST), in conjunction with the Department of Defense (DOD) and the National Security Agency (NSA), have published information on the U.S. government's Federal PKI efforts. Published documents can be found at http://csrc.nist.gov/pki/.

Canadian Government

The Canadian government has a similar PKI effort under the auspices of the Treasury Board of Canada Secretariat. Published information on Canada's PKI efforts can be found at http://www.cio-dpi.gc.ca/pki/splash_e.html.

For More Information

For the latest information on Windows NT Server, visit our World Wide Web site at http://www.microsoft.com/ntserver or the Windows NT Server Forum on the Microsoft Network (GO WORD: MSNTS).