Public Key Infrastructure and authentication

Public Key Infrastructure and authentication

The degree of security of a system is largely governed by the quality of the authentication procedures that are employed. Authentication may be defined as the process by which proof of identity or of integrity is established in response to some form of challenge. This chapter examines authentication methodologies based on asymmetric algorithms and their application to VPNs.

During the initial negotiation phases between two security endpoints, each peer should authenticate the other by some means. Failure to implement such peer authentication could allow an unknown system to masquerade as the remote end of a VPN in order to acquire confidential data. For example, if a university has provided a remote user VPN facility, only recognised computers should be allowed to establish a tunnel with the remote access server, otherwise any Internet-connected computer could inject packets into the private campus LAN. Mutual authentication of the security endpoints is always the first stage of establishing a VPN connection. If this fails, all succeeding processes must stop so that the VPN connection is not established.

This form of device authentication is completely separate from the more familiar user authentication whereby a human submits their username and password in order to gain access to network resources. Device authentication is concerned with determining whether the basic network infrastructure of a VPN should be constructed. Only if these communication channels have been established can user authentication take place. Mistakenly deploying user-authentication instead of device authentication can result in catastrophic security flaws.

Once identities have been established, data can be transmitted over the VPN. All packets arriving at the security endpoints must be submitted to an integrity authentication procedure to ensure that the security peer (and not some other device) sent them and that they were not modified during the course of their transmission over the public infrastructure. This is accomplished by attaching a digital signature to each packet, which is then checked by the recipient endpoint.

Digital Signatures

Public Key cryptography can be used to authenticate, as well as encrypt, a message by transmitting a digital signature along with the message data. The previous chapter described how messages may be encrypted with a recipient’s Public Key. Only the holder of the

corresponding Private Key is able to decrypt the message. If these actions are performed in the opposite sense then a simple form of digital signature has been achieved, implying that the messages originate from the sender and not an impersonator. The sender encrypts messages with his or her Private Key and the recipient decrypts them with the sender's Public Key.

This simplistic attempt at authentication relies upon the transformation of encrypted gibberish into legible plaintext. While a human reader can tell the difference between ciphertext and plaintext, a machine cannot be expected to make this distinction and so something further is required to provide a worthwhile signature. The signatory passes the message through a one-way hash function that produces a unique fixed-length summary called the message *digest* as its output. This is encrypted with the sender's Private Key to produce a digital signature. A hash function produces a unique thumbprint that is characteristic of the message from which it is derived. It is safe to assume that a signed digest produced in this manner is a trustworthy digital signature.Furthermore, it is amenable to machine verification. The signature, in the form of an encrypted digest, is appended to the message it is authenticating. The recipient verifies the signature by passing the message through the same hash function and decrypting the signature. If the two digests are identical then the signature is valid.

Digital signatures constructed by encrypting a message digest with the sender's Private Key are extremely difficult to forge, cannot later be repudiated and are non-transferable. This makes them ideal vehicles for guaranteeing the authenticity of the sender.

Message Integrity

A digital signature can guarantee the integrity of the message it accompanies as well as the identity of the sender because any alterations to a signed message will produce an entirely different hash value, rendering the signature invalid. However, digital signatures are slow to calculate because they are generated by means of asymmetric ciphers. For an ongoing stream of data (such as a series of IP packets), signing each protected packet would be as onerous as encrypting the payload with the recipient's Public Key rather than using a symmetric block cipher.

Fortunately there are symmetric and asymmetric hash functions in the same way as there are symmetric and asymmetric ciphers. A digital signature uses a slow, highly secure asymmetric hash function. Once the peers have authenticated one another by these means, the sender of the data is trusted and no longer needs to be authenticated. A faster symmetric hashing function is sufficient to guarantee the integrity of any received packets. Symmetric hash functions where a single shared key is used to sign the input are known as MACs (Message Authentication Codes).

Two commonly implemented hash functions are SHA (Secure Hash Algorithm) and MD5 (Message Digest-5). These functions simply produce a fixed-length digest of a variable length input. It is important that the keyed variant of these functions is employed, otherwise an attacker could alter the payload of a packet and simply recompute the digest. A special type of keyed hash exists, known as HMAC (Hash-Keyed Message Authentication Code), that can be utilised with any existing hash function and so the keyed versions of the aforementioned hash functions are known as HMAC-SHA and HMAC-MD5.

It is worth noting that some of the security protocols mentioned in this guide may become

ineffective in the future. For example methods to forge MD5 hashes have been demonstrated.

Digital Certificates

Digital signatures prove conclusively that a message was received in an unmolested state from the peer holding a given Public Key. However, there is a strong element of trust still involved because the message's recipient has no proof of the sender's identity. A certificate is a device, issued and digitally signed by a trusted third party, that binds the holder's identity to a Public Key. There is an agreed standard called X.509 (which is a member of the family of X.500 directory standards) that governs the properties a digital certificate must implement [5] [1] . Certificates are non-transferrable, non-forgeable files that act as a digital identity badge or passport to help ensure that users or computers are who they say they are.

Certification Authorities (CAs)

A certificate may be obtained by applying to some issuing party, called a CA. This may be either a company that specialises in issuing digital certificates or a private organisation that requires users to submit a certificate it has previously issued as proof of digital identity. The CA should take reasonable steps to confirm the identity of applicants before issuing a certificate and should undertake various maintenance tasks throughout a certificate's lifetime. These include the revocation of any certificate whose Private Key has been compromised or whose holder has left the issuing organisation. Most importantly, the CA should ensure that its own Private Keys are never revealed as this would allow an attacker to issue bogus certificates under the issuer's name, thereby casting doubt on every certificate emanating from that CA.

Registration Authorities (RAs)

The purpose of the optional RA is to verify the information supplied by an applicant for a certificate. The CA may delegate this task to the RA or perform the checks itself. If the certificate confirms the identity of a person (as opposed to a computer), such verification might comprise checking the supplied e-mail address or, for more demanding applications, require a personal visit to the RA's offices to prove identity by displaying a document such as a driving license or passport. Once the various checks have been conducted, the RA approves the request for a certificate by submitting it to the CA which then issues the certificate. The certificate will usually contain an indication of the identity checks performed.

Anatomy of a Certificate

Version 3 of the X.509 standard is most commonly implemented. It allows more detailed identifying information and requires that a certificate bears the following items of information:

- the holder’s identifiers (name, organisation, address, etc.)
- the holder’s Public Key
- the certificate’s validity dates
- the certificate’s serial number
- the name and signature of the issuer
- the signature algorithm (usually SHA-1)
- the applicable X.509v3.

Trust Relationships

When two security endpoints exchange certificates they are delegating the process of confirming the peer’s identity to a CA. Acceptance of a certificate implies that the authenticator trusts the issuing authority, which in turn implies that the authenticator must have some existing knowledge of the CA.

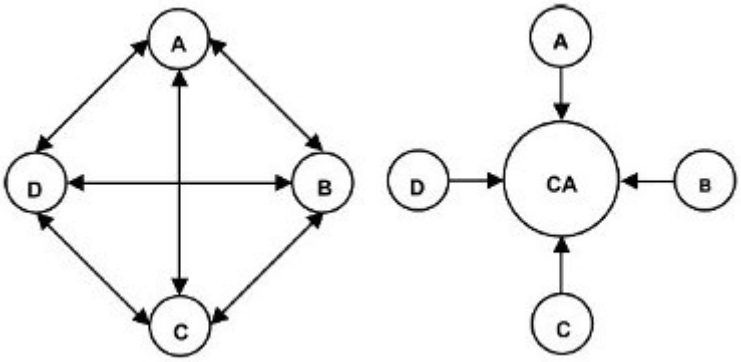
As well as confirming the identities of others, every CA publishes a special type of certificate that contains its own Public Key and is self-signed. This is tantamount to declaring one’s own identity without any supporting proof. A node wishing to authenticate a peer must first establish a trust relationship with the CA that issued thepeer’s certificate. Installing the CA’s self-signed certificate into the node’s *trusted root* certificate store achieves this end. The identity of any third party that subsequently presents a certificate from this now trusted CA would then be assumed valid. When a group of nodes establish individual trust relationships with a common CA, and are all issued certificates by that CA, the participants form a *Public Key Infrastructure* (PKI).

The participation of a CA greatly reduces the number of trust relationships that must be formed. Without a CA, a fully meshed arrangement in which each node establishes direct one-to-one trust relationships with every other node in the PKI would be required. The number of trust relationships can be calculated from:

(N2 – N)/2

where *N* is the number of nodes.

For a PKI consisting of just 25 members, no fewer than 300 direct trust relationships would be required, as opposed to 25 (the same as the number of nodes) with a CA at the centre of the PKI. Figure 2 graphically demonstrates two PKI structures with and without a central CA. The



ıst.

[2]

Figure 2. Graphic demonstration of two PKI structures with and without a central CA.

The direction of the arrows indicates the direction of the trust.

The left-hand diagram depicts a PKI based upon a fully meshed structure and the right-hand diagram shows how the presence of a CA reduces the number of trust relationships. The solid arrows indicate a trust relationship between either two nodes or between a node and the CA. In the right-hand diagram, even though there is no direct trust relationship between any two nodes, this is implied because they are members of a common PKI. Consequently, any two of the nodes will successfully authenticate one another.

Trust Models

The type of PKI hitherto described is a very simple system in which a node is only able to authenticate other members of the same PKI. Suppose that Alice and Bob wish to authenticate one another but are members of different PKIs. By establishing trust relationships between CAs, the trust model can be rapidly extended. Two different methods are recognised.

Cross-certification Method

A cross-certification system can exist where Alice's CA signs the certificate of Bob's CA and vice versa. Because Alice trusts her CA not to misuse its signature, when it signs a certificate bearing the identity and Public Key of another CA, Alice will then trust that CA in exactly the same manner as she would any other node in her PKI. By extension, therefore, she trusts any node whose certificate has been signed by the alien, but now trusted, CA.

Hierarchical Method

Another trust model is based on the familiar hierarchical method with the pinnacle represented by a 'root' CA that is implicitly trusted by all the other subservient CAs. Immediately below the root there is a level of CAs whose own certificates the root has signed. Further levels of CAs may be added whose own certificates a CA in the level immediately above has signed.

This type of hierarchical CA is suitable for constructing an overarching PKI that covers an entire organisation, such as a university, that consists of partially autonomous faculties and departments. The cross certification system is clearly better suited for use by a number of completely autonomous organisations such as separate universities.

Authentication by Certificates

Consider a situation where a server provides some form of network access to remote clients. While in practice both devices are peers of equal standing and check one another's certificates, it is convenient to consider just one of these transactions (the access 'server' authenticating the remote 'client') for illustrative purposes.

The client digitally signs (but does not encrypt) its certificate and sends it to the server. The server extracts the Public Key from the certificate and uses this to decrypt the signature. By

comparing the digest from the decrypted signature with a digest it computes, the server verifies that the sender of the certificate is also its holder. A similar check is now performed on the CA's digital signature that has been extracted from the client's certificate. This second check can only be performed if the server knows the CA's Public Key and this will only be the case if the server holds a copy of the CA's self-signed certificate. The server must trust the client's CA in order to verify the certificate that it issued to the client. It should now be evident why the two security peers must belong to a common PKI in order for the mutual authentication process between them to conclude successfully.

An additional optional step in the authentication procedure involves the peers consulting the CA's Certificate Revocation List. This is a mechanism by which a CA can revoke a certificate prior to its expiry date. Such a mechanism allows a college to issue certificates to staff members and then withdraw the certificate should a member of staff leave the college's employment.

When two security peers exchange certificates, it is worth noting that they obtain one another's Public Keys that can be used for the subsequent strong encryptions of, for example, a shared DES key.

Authentication Models

VPNs have already been classified according to whether they are used to connect a remote site or a remote user to the main campus. Different models of identity authentication are most appropriate for these two types of VPN.

Peer authentication within the *remote site* category of VPN can generally be accomplished by configuring a shared secret on both of the peers. Cisco® routers allow the secret phrase to be mapped to the peer endpoint's identity (usually its hostname or IP address). This is akin to the CHAP (Challenge Handshake Authentication Protocol) authentication method associated with PPP and may be considered secure providing both of the endpoint routers reside within the same management domain (for example the college's computer services department). Although local circumstances may dictate otherwise, it should not be necessary to resort to the expense and complexity of installing certificates on the peer routers.

An extensive *remote user* VPN requires a more rigorous approach because individual users, not the college authorities, will control the remote computers. It is important that, should the need arise, the remote access facility can be withdrawn immediately without requiring access to the device concerned. User authentication is insufficiently secure because suspension of an account cannot guarantee that the user would not be able to gain remote access to the network using a colleague's password. If the remote computer is authenticated using a digital certificate then access can be withdrawn by placing the certificate on the CA's revocation list.

Links
[1] https://community.ja.net/library/advisory-services/references-0
[2] http://community.ja.net/system/files/images/tg-vpn-02.jpg