Closed Administrative Domains

by David G. Messerschmitt

Supplementary section for Chapter 1 of Understanding Networked Applications: A First Course, Morgan Kaufmann, 1999.

Copyright notice: Permission is granted to copy and distribute this material for educational purposes only, provided that this copyright notice remains attached.

The security techniques described thus far are well suited to a public environment, like consumer electronic commerce. The asymmetric encryption algorithm supports authentication, integrity, and confidentiality, without the hassles of distributing secret symmetrical keys. The price paid—which raises its own set of issues—is the need for a certificate authority and digital certificates.

In a closed administrative domain, the infrastructure of digital certificates can be avoided by distributing secret symmetrical keys directly to users or hosts. This enables a class of authentication, access, and confidentiality protocols based on a shared secret—the presumption that two or more entities (users or hosts) share a common secret. This shared secret can be used for both authentication (using a challenge-response protocol) and confidentiality.

Passwords

Authentication of users for access control in a closed administrative environment is often based on a password—a character string chosen by the user and shared with a host or application. Thus, both the user and the host or application are presumed to share this secret. Authentication is based on the user presenting an identification (user name) and associated password.

Passwords have serious weaknesses as a means for authentication:

• While a user can avoid writing down or storing his or her password by committing it to memory, thus preventing it from being stolen, a host or application must store the password, or equivalent information. This makes the password vulnerable to being stolen. A partial solution to this problem is to store not the password, but a one-way function computed on the password (like the message digest). The password itself cannot be inferred from the stored value, and the user can be authenticated by having him supply the password, computing the one-way function, and comparing with the stored value.

• A password authenticates only the user, and not the host or application. Thus, the user can easily be fooled by an imposter that pretends to validate a password, but actually ignores it.

• A password can be stolen by simply issuing a bogus request for that password—users are sometimes gullible.

• A password can be stolen as it crosses the network. Fortunately, this weakness can be avoided by using a challenge-response protocol, challenging the user to prove he has the password without revealing it.

• In pragmatic terms, the user faces a dilemma. He can establish a different password for each host or application, but he then has difficulty remembering this proliferation of passwords. Alternatively, he can use the same password, but this increases the vulnerability to theft.

As networked applications proliferate, passwords are rapidly losing their appeal. In a public envi-
vironment, the digital certificate and certificate authority are viable and secure alternatives. In a controlled administrative environment, a secure alternative that avoids certificates is the trusted authentication server.

**Authentication Servers**

Some of the weaknesses of passwords can be avoided by using shared secrets for authentication and confidentiality. However, in a typical multi-party environment, using shared secrets as a basis for authentication results in a proliferation of secrets. If \( n \) parties (users and/or hosts) need to authenticate one another, then they must each share a unique secret with \( n - 1 \) others. Thus, there are \( n \cdot (n - 1)/2 \) shared secrets required in total. Distributing so many secrets, insuring the secret is passed to the right parties, and protecting them all is an intolerable burden.

This problem is eased with the aid of a **trusted authentication server**. The idea is simple: If two parties are each authenticated by the server, and if they trust the server, then they have indirectly authenticated one another. In effect, each party asks the server to authenticate the other on its behalf, rather than doing it directly, and then trusts the server’s authentication. Each party must share a secret with the server, but not with the other party, reducing the total number of shared secrets to \( n \) — one for each party. This idea and its benefit are similar to the conversion among \( n \) representations using a common intermediate form described in "Two Ways to Convert Representations" on page 157.

At the same time as it authenticates the parties, an authentication server can also provide them a single-use random symmetric session key, so that they can communicate confidentially. Thus, the authentication server distributes symmetric keys in much the same fashion as a certificate authority distributes asymmetric public keys.

In practical terms, the authentication server becomes a point of serious vulnerability, since it stores all the shared secrets in one place. Thus, it must be physically secured and carefully administered by vigilant and trusted workers. It reduces the logistical problems by focusing security on a single server, trading that off against the much greater vulnerability should that server ever be compromised. A commonly used authentication server protocol is **Kerberos** (see the sidebar "Kerberos").

**Access Control**

Usually security features are built into software applications, and don’t disturb the user. However, they can be invasive, as for example when a user must obtain a digital certificate to participate in a secure application. As a result, strong security techniques are sometimes not used, in spite of the vulnerabilities that result. To minimize these problems, a critical element of an organization’s security is a set of **security policies** established as part of a comprehensive plan for achieving a desired trade-off between security and ease of use. Policies establish, for example, when confidentiality and authentication are required or not required. Policies are enforced primarily by system administrators, who configure hosts and applications in accordance with those policies.

One essential policy-based security measure is **access control**, which determines and limits user access to individual applications, to individual hosts, or to entire intranets. An important security policy regards access rules based on “need to know” and “need to use” criteria.

**Example…** If a publication style social application serves the citizenry or an interest group it may not require any access restrictions. However, if it serves a work group, restricting access to only the
members of that group may be appropriate if proprietary information is involved. Similarly, for information management applications, information that is sold over the network should restrict access to only those users who pay for it.

Access control requires several elements:
- A secure access database describing access authorizations for each user.
- Authentication protocols to authenticate each user requesting authorized access. This requires that authorized users must possess some attribute (location, secret, physical characteristic) that can be verified.

Kerberos

Kerberos is widely used in closed administrative environments for overcoming the limitations of passwords. It assumes that each party shares a secret (which can be a one-way function applied to a password) with the authentication server. The server can assist parties A and B to authenticate one another—and at the same time confidentially distribute a random session key to each—without sending secrets over the network. It uses a randomized challenge/response protocol, as illustrated in Figure 1.

The server, having been informed that ID_A and ID_B wish to authenticate one another, generates a random session key (ABSK) and a random challenge k that it will provide to both A and B. The server also shares secret symmetric encryption keys (ASK and BSK) with A and B. It sends \([k,ABSK,ID_A]\) to B (indirectly thorough A), encrypting it with BSK so that only B can read it. The server also sends \([k,ABSK,ID_B]\) to A, encrypting it with ASK so that only A can read it. Now A and B both possess k and ABSK and implicitly challenge one another to prove they possess k (without revealing it). A proves this to B by sending k encrypted by ABSK, and B proves this to A by sending \(k+1\) encrypted by ABSK. Subsequently, A and B can conduct a session using ABSK as a session key.

**Figure 1.** The Kerberos protocol allowing A and B to authenticate one another and obtain a shared secret key with the help of an authentication server.
• Some way to prevent access to unauthorized or unauthenticated users.

An alternative, and somewhat less effective form of access control, is the firewall discussed in "Intranet" (Section 4.5.1 on page 109). Rather than authenticating each user individually, the firewall bases access control on location (see the sidebar "Firewalls for Access Control").

Discussion

D1 What are some weaknesses and vulnerabilities of the security protocols and policies discussed here? Equipped with this understanding of available tools, do you believe you can trust Internet electronic commerce?

Firewalls for Access Control

A firewall is imposed on each network link connecting a protected enclave (the intranet) from the global Internet. It is configured to restrict communication passing through, and offers a focal point for enforcing access policies in either direction. Typically it prohibits access to certain resources for anybody outside the firewall, and permits access to anybody inside the firewall. Further, a firewall often restricts communication protocols and applications.

If hosts were intrinsically completely secure, they could be directly connected to the global Internet, but in practice hosts and their applications have security loopholes. The greatest loophole is the users with legitimate access to the host, who are often not as trained on or conscious of security issues as would be desirable. If an intruder gains access to a legitimate user account, for all practical purposes that host is compromised. Thus, it is better if intruders cannot reach the host at all. Unfortunately, it isn’t as simple as a blanket denial of access to all outsiders, because an organization typically want unrestricted access to some applications, such as Web servers targeted at the general public.

A n a l o g y ... Countries typically have checkpoints (analogous to a firewall) at their border crossings where laws relative to contraband, immigration and emigration, etc. are enforced. A country usually restricts its borders, but also allows other countries to operate embassies and consulates within its borders. That other country may be permitted free passage and unrestricted access to its own embassy.

Care must be exercised to avoid connections bypassing the firewall, such as dial-up lines or modems attached to individual phones. An intranet offers no protection against threats within the trusted enclave unless there are internal firewalls. The more protocols and applications that are admitted by a firewall, the easier it is for legitimate users, but the more vulnerable the organization’s data and applications.

Final Defense: Legal Sanctions

No matter how secure, it is possible for an intruder to gain access by surreptitious or fraudulent means. An extreme example would be breaking and entering a physically restricted facility, but there are other ways, such as a confidence game or blackmail. In these cases, the intruder may have broken federal or local laws, and can be apprehended and prosecuted.

E x a m p l e ... A particularly notorious individual who gained surreptitious access to many systems is Kevin Mitnick [Haf95], who was convicted of a felony.

Laws and legal sanctions are the ultimate protector of computer systems.
D2 The use of a secret key to maintain confidentiality has been emphasized. Another approach that would be feasible in a coordinated environment would be to keep an encryption algorithm secret, and perhaps not even have a key. Discuss the merits or problems of this approach.

D3 In terms of proving the authenticity of a signature in court, discuss the relative merits of a physical signature vs. a digital signature.

Concepts

Security policies:
- Access control
- Firewalls
- Authentication: passwords and authentication servers (Kerberos)

Exercises

E1. Government cryptography policy was discussed in Chapter 13 from a U.S. perspective. Discuss similar policy issues from the perspective of each of the following, emphasizing how they may differ from the U.S.
   a. The government of France
   b. The government of Israel
   c. The government of Russia
   d. Citibank

E2. Before you give your credit card number to a mail- or phone-order merchant, how do you “authenticate” them? (Your initial reaction may be that you don’t, but chances are you do, perhaps subconsciously.)

E3. In using a password for authentication, somehow a server has to learn the one-way function of the user’s password. You want to do this securely, i.e. without revealing the password in an untrusted domain.
   a. One method might to be capture the password in a client workstation and then send the one-way function value to the server rather than the password. Why does this method not work?
   b. Describe how this could be done securely.
   c. Can you define a second approach?

E4. Consider the following authentication procedure between a client and server, which are presumed in advance to share a secret. The server challenges the client by sending a random string. The client appends the shared secret to this string, applies a one-way function, and sends the resulting value back to the server. The server calculates the same one-way function and confirms a match.
   a. Assuming the shared secret is a password, does this procedure avoid sending the password over the untrusted domain?
   b. Does this procedure authenticate the server to the client?
   c. Compare this technique to the standard password algorithm with respect to the theft of the shared secret from the server.