Network Working Group
Internet Draft

Steven M. Bellovin
AT&T Labs Research

Expiration Date: May 1999

November 1998

Security Mechanisms for the Internet

draft-iab-secmech-00.txt

1. Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

2. Abstract

Many different mechanisms can be used to provide security for protocols. The precise one that is appropriate in any given situation can vary. We review a number of different choices, explaining the properties of each.

3. Introduction

A number of possible mechanisms can be used to provide security on the Internet. Which one should be selected depends on many different factors. We attempt here to provide guidance, spelling out the factors and the currently-standardized (or about-to-be-standardized) solutions, as discussed at the IAB Security Architecture Workshop [RFC2316].

Security, however, is an art, not a science. Attempting to follow a recipe blindly can lead to disaster. As always, good taste in protocol design should be exercised.

Finally, security mechanisms are not magic pixie dust that can be sprinkled over completed protocols. It is rare that security can be bolted on later. Good designs--that is, secure, clean, and efficient designs--occur when the security mechanisms are crafted along with the protocol.

4. Decision Factors

4.1. Threat Model

The most important factor in chosing a security mechanism is the threat model. That is, who may be expected to attack what resource, using what sorts of mechanisms? A low-value target, such as a Web site that "charges" demographic information only, may not merit much protection. Conversely, a resource that if compromised could expose significant parts of the Internet infrastructure--say, a major backbone router or high-level nameserver--should be protected by very strong mechanisms.

The value of a target to an attacker may depend on where it is located. A network monitoring station that is physically on a backbone cable is a major target, since it could easily be turned into an eavesdropping station. The same machine, if located on a stub net and used for word processing, would be at little risk.

One must also consider what sorts of attacks may be expected. At a minimum, eavesdropping must be seen as a serious threat; there have been too many such incidents since at least 1993. In many circumstances, active attacks--that is, attacks that involve insertion or deletion of packets by the attacker--are a risk as well.

Finally, of course, there is the cost to the defender of using cryptography. This cost is dropping rapidly; Moore's Law, plus the

easy availability of cryptographic components and toolkits, makes it relatively easy to use strong protective techniques. Although there are exceptions--public key operations are still expensive, perhaps prohibitively so if the cost of each public-key operation is spread over too few transactions--the default today should be to use the strong cryptography available.

4.2. Granularity of Protection

Some security mechanisms can protect an entire network. While this economizes on hardware, it can leave the interior of such networks open to attacks from the inside. Other mechanisms can provide protection down to the individual user of a timeshared machine, though perhaps at risk of user impersonation if the machine has been compromised.

When assessing the desired granularity of protection, protocol designers should take into account likely usage patterns, implementation layer (see below), and deployability. If a protocol is likely to be used only from within a secure cluster of machines (say, a NOC), subnet granularity may be appropriate. By contrast, a security mechanism peculiar to a single application is best embedded in that application, rather than inside TCP; otherwise, deployment will be very difficult.

4.3. Implementation Layer

Security mechanisms can be located at any layer. In general, putting a mechanism at a lower layer protects a wider variety of higher-layer protocols. The usual tradeoff is reach; lower-layer protocols terminate sooner. Thus, a link-layer encryptor can protect not just IP, but even ARP packets. However, its reach is just that one link. Conversely, a signed email message is protected even if sent across many store-and-forward mail gateways; however, only that one type of message is protected. Messages of similar formats, such as some Netnews postings, are not protected unless the mechanism is specifically adapted and then implemented in the news-handling programs.

5. Standard Security Mechanisms

5.1. Plaintext Passowrds

Plaintext passwords are the most common security mechanism in use today. Unfortunately, they are also the weakest. When not protected by an encryption layer, they are completely unacceptable. Even when used with encryption, plaintext passwords are quite weak, since they must be transmitted to the remote system. If that system has been compromised or if the encryption layer does not include effective authentication of the server to the client, an enemy can collect the passwords and possibly use them against other targets.

Another weakness arises because of common implementation techniques. It is considered good form [MT79] for the host to store a one-way hash of the users' passwords, rather than their plaintext form. However, that may preclude migrating to stronger authentication mechanisms, such as HMAC-based challenge/response.

The strongest attack against passwords, other than eavesdropping, is password-guessing. With a suitable program and dictionary (and these are widely available), 20-30% of passwords can be guessed in most environments.

5.2. One-Time Passwords

One-time password schemes, such as that described in [RFC2289], are very much stronger. The host need not store a copy of the user's password, nor is it ever transmitted over the network. However, there are some risks. Since the transmitted string is derived from a user-typed password, guessing attacks may still be feasible. (Indeed, a program to launch just this attack is readily available.) Furthermore, the user's login inherently expires after predetermined number of uses. While in many cases this is a feature, an implementation most likely needs to provide a way to reinitialize the authentication database, without requiring that the new password be sent in the clear across the network.

5.3. HMAC

HMAC [RFC2104] is the preferred shared-secret authentication technique. If both sides know the same secret key, HMAC can be used to authenticate any arbitrary message. This specifically includes random challenges, which means that HMAC is suitable for logins.

The disadvantage, of course, is that the secret must be known in the clear by both parties. In many situations, this is undesirable.

When suitable, HMAC should be used in preference to older techniques, notably keyed hash functions. Keyed hashes based on MD5 [RFC1321] are especially to be avoided, given the hints of weakness in MD5.

5.4. IPSEC

IPSEC [longlist] is the generic IP-layer encryption and authentication protocol. As such, it protects all upper layers, including both TCP and UDP. Its normal granularity of protection is host-to-host, host-to-gateway, and gateway-to-gateway. The specification does permit user-granularity protection, but this is comparatively rare.

Because IPSEC is installed at the IP layer, it is rather intrusive. Implementing it generally requires either new hardware or a new protocol stack. This makes it a poor choice for individual applications, at least until IPSEC is more widely deployed.

The key management for IPSEC can use either certificates or shared secrets. For all the obvious reasons, certificates are preferred; however, they may present more of a headache for the system manager.

There is strong potential for conflict between IPSEC and NAT [Hain98]. NAT does not easily co-exist with any protocol containing embedded IP address; with IPSEC, every packet, for every protocol, contains such addresses, if only in the headers. The conflict can sometimes be avoided by using tunnel mode, but that is not always an appropriate choice for other reasons.

5.5. TLS

TLS provides an encrypted, authenticated channel that runs on top of TCP. While TLS was primarily intended for use by Web browsers, it is by no means restricted to such. In general, though, each application that wishes to use TLS will need to be converted individually.

Generally, the server side is always authenticatd by a certificate. Clients may possess certificates, too, providing bilateral authentication. The reality, though, is that for most practical Web use, there is no authentication, since users do not check certificates [Bell98]. Designers should thus be wary of demanding plaintext passwords, even over TLS-protected connections.

5.6. SASL

SASL [RFC2222] is a framework for negotiating an authentication and encryption mechanism to be used over a TCP stream. As such, its security properties are those of the negotiated mechanism.

5.7. DNSSEC

DNSSEC [RFC2065] digitally signs DNS records. It is an essential tool for protecting against cache contamination attacks; these in turn can be used to defeat name-based authentication and to redirect traffic to or past an attacker. The latter makes DNSSEC an essential component of some other security mechanisms, notably IPSEC.

5.8. Security/Multipart

Security/Multipart [RFC1847] is the preferred mechanism for protecting email. More precisely, it is the MIME framework within which encryption and/or digital signatures are used. Conforming mail readers can easily recognize and process the cryptographic portions of the mail.

Security/Multipart represents one form of "object security", where the object of interest to the end user is protected, independent of transport mechanism, intermediate storage, etc. Currently, there is no general form of object protection available in the Internet.

5.9. OpenPGP and S/MIME

At this writing, two different secure mail standards, OpenPGP and S/MIME, have been proposed to replace PEM. It is not clear which, if either, will succeed. Both use certificates to identify users; both can provide secrecy and authentication of mail messages. Historically, the difference between PGP-based mail and S/MIME-based mail has been the style of certificate chaining. In S/MIME, users possess X.509 certificates; the certification graph is a tree with a very small number of roots. By contrast, PGP uses the so-called "web of trust", where any user can sign anyone else's certificate. The certification graph really is an arbitrary graph or set of graphs.

With any certificate scheme, trust depends on two primary characteristics. First, it must start from a known-reliable source -- either an X.509 root, or someone highly trusted by the verifier, often him or herself. Second, the chain of signatures must be reliable. That is, each node in the certification graph is crucial;

if it is dishonest or has been compromised, any certificates it has vouched for cannot be trusted. All other factors being equal (and they rarely are), shorter chains are preferable.

5.10. Firewalls and Topology

Firewalls are a topological defense mechanism. That is, they rely on a well-defined boundary between the good "inside" and the bad "outside" of some domain, with the firewall mediating the passage of information. While firewalls can be very valuable if employed properly, there are limits to their ability to protect a network.

The first limitation, of course, is that firewalls cannot protect against inside attacks. While the actual incidence rate of such attacks is not known (and is probably unknowable), there is no doubt that it is substantial, and arguably constitutes a majority of security problems. More generally, given that firewalls require a well-delimited boundary, to the extent that such a boundary does not exist, firewalls do not help. Any external connections, whether they are protocols that are deliberately passed through the firewall, links that are tunneled through, or direct external connections from nominally-inside hosts, weaken the protection. It should be noted that this phenomena can vitiate one oft-touted advantage of firewalls, that they hide the existence of internal hosts from outside eyes. Given the amount of leakage, the likelihood of successfully hiding machines is rather low.

In a more subtle vein, firewalls hurt the end-to-end model of the Internet and its protocols. Indeed, not all protocols can be passed safely or easily through firewalls [Freed98]. Sites that rely on firewalls for security may find themselves cut off from new and useful aspects of the Internet.

Firewalls work best when they are used as one element of a total security structure. For example, a strict firewall may be used to separate an exposed Web server from a back-end database, with the only opening the communication channel between the two. Similarly, a firewall that permitted only encrypted tunnel traffic could be used to secure a piece of a VPN. On the other hand, in that case the other end of the VPN would need to be equally secured.

6. Security Considerations

No security mechanisms are perfect. If nothing else, any network-based security mechanism can be thwarted by compromise of the endpoints. That said, each of the mechanisms described here have their own limitations. Any decision to adopt a given mechanism should weigh all of the possible failure modes. These in turn should be weighed against the risks to the endpoint of a security failure.

7. Acknowledgements

Brian Carpenter, Tony Hain, and Marcus Leech made a number of useful suggestions. Much of the substance comes from the participants in the IAB Security Architecture Workshop.

8. References

[more]

9. Author Information

Steven M. Bellovin
AT&T Labs Research
Shannon Laboratory
180 Park Avenue
Florham Park, NJ 07974
Phone: +1 973-360-8656
email: smb@research.att.com