Internet Security Standards: Past, Present, and Future

Stephen Kent
BOLT BERANEK AND NEWMAN, INC

■ **Security is a topic of great interest as the Internet transitions from the R & D environment to the commercial sector and the home. This article traces the evolution of security standards in the Internet and previews work now underway.**

The Internet protocol suite (e.g., IP, TCP) has been criticized as having been designed with no thought of security. People point to the ease with which IP addresses can be spoofed; the lack of security for name and address mappings provided by the Domain Name System (DNS); the lack of accounting facilities; the difficulty of operating some protocols across "firewall gateways," and similar characteristics, as evidence of failure to anticipate security requirements. These observations, while generally true, do not fully support the criticism. For example, IP was designed to operate over lower network layer protocols such as X.25, and it was assumed that these lower network layer protocols would enforce network-specific charging policies. The construction of networks from IP routers without the use of a lower network layer protocol was not part of the IP model, which also explains the lack of congestion control facilities in IP. Contrary to popular belief, IP was designed with a security model in mind [Kent 1993a]. The model assumes the use of end-to-end cryptographic protection at the network layer for most user-oriented security services and the use of link layer cryptography for traffic-flow confidentiality. TCP/IP was developed initially for use by the U.S. Department of Defense (DoD). In the DoD environment, the threats are such that the only accepted means of providing high-quality security in a large, geographically distributed network is through the application of cryptography.

Appropriately designed, IP-layer cryptographic devices offer (connectionless) confidentiality and integrity, data-origin authentication, and enforcement of identity and rule-based access control through automated key distribution. Uniform use of such cryptographic security technology addresses many of the concerns cited above. Prototype devices implementing these services in the TCP/IP environment were developed, tested, and deployed on a limited basis in the late 1970s as part of DoD-sponsored R&D programs—well before security became a common concern for many Internet users, before the term "information superhighway" became a buzzword, and before the advent of the Internet standards process.

Work is underway to develop a multivolume security architecture document for the Internet.

This model for security was not all-encompassing. For example, electronic mail was not addressed explicitly, despite the fact that IP-layer security cannot afford complete protection to email, due to the use of application layer relays. The Domain Name System (DNS) was not initially part of the Internet design, and no explicit security features were envisioned for protecting the name and address mappings, beyond the use of trusted computers and (IP-layer) secure communication paths to these servers. So even if IP layer, end-to-end cryptography were widely employed in the Internet, there would still be need for additional security standards.

Overview

The development of Internet standards for security (i.e., RFCs published as standards track documents) has been slow. The only explicit, security-oriented aspect of the Internet Protocol from the beginning has been a security label facility, the IP Security Option (IPSO),[1] which was present in the IP specification published in 1980 [DoD 1980]. The first RFC to introduce a security-oriented protocol into the TCP/IP suite did not appear until 1987 [Linn 1987], with the publication of the first in a series on email security protocol specifications.

In the intervening seven years, a number of security-oriented Internet protocols have been developed and are at various stages in the Internet standardization process [IAB 1994]. Internet standards are formally developed under the auspices of the Internet Society. The technical arm of the Internet Society is the Internet Architecture Board (IAB). There are two task forces under the IAB: the Internet Research Task Force (IRTF) and the Internet Engineering Task Force (IETF). The IRTF consists of a small number of research groups that explore advanced topics in Internet R&D, whereas the IETF consists of a large number of working groups (WGs)[2], where the bulk of standards development takes place.

The work of the IETF WGs is managed by the Internet Engineering Steering Group (IESG), the membership of which consists primarily of technical area directors.[3] All Internet standards are approved by the IESG, although the IAB can become involved to arbitrate disputes. Security standards fall under the purview of the security area director, although security-relevant standards are sometimes developed in WGs in other areas, with the assistance and concurrence of the security area director.

Despite the greatly increased activity in security standards in the IETF, the Internet lacks a security architecture. There is no published framework in which to evaluate proposed security standards, no standard technology for describing security services or mechanisms, and no mapping of services to protocol layers or of mechanisms to services. The Internet lacks a document analogous to ISO 7498-2 [Inf. Tech. 1987], the security addendum to the OSI reference model.

Work is underway to develop a multivolume security architecture document for the Internet. One volume will address all of the architectural aspects covered in 7498-2; another will be a "living" document that describes security mechanisms and will, therefore, be updated to encompass new mechanisms as they arise. The last volume will also be extensible and will describe security requirements for protocols used in the Internet and, where applicable, the security features associated with these protocols. This document, being developed by the Privacy and Security Research Group (PSRG) of the IRTF, when completed, will be submitted to the IESG area director for evaluation as an Internet standard.

In the remainder of this article, we'll briefly review major Internet security protocols, those that are already on the standards track, and those still under development within the WGs. The protocols are grouped by layer, starting with the lower layers and working up to the application layer, followed by security standards for the overall Internet infrastructure. For protocols on the standards track, the relevant RFCs are cited. For ongoing efforts within IETF WGs, the protocols are described in Internet drafts, but are not cited because such documents are intentionally short-lived (they expire in six months) and thus are not suitable for an archival journal.

IP and Lower Layer Security

For most purposes, the Internet architecture begins with IP at the OSI upper network layer. Internet standards addressing lower layers deal primarily with conventions for transport of IP packets over various WAN, LAN, and serial line protocols. Thus security standards below the IP layer tend to address requirements associated with protocols used to support IP transport—for example, link-by-link authentication.

1 The original version of the IPSO was part of the IP specification in the late 1970s. Subsequent versions were published during the 1980s, culminating with the publication of a revised standard IPSO in 1991 [RFC 1108].

2 The number of working groups varies, as new ones are formed and old ones are terminated; currently there are about 50.

3 The number of directorates varies, but more slowly than the number of working groups; currently there are ten permanent directorates.

IP security labels, in the absence of measures to verify their integrity and authenticity in the global Internet environment, may be of questionable utility.

PPP AUTHENTICATION

The Point to Point Protocol (PPP) [Simpson 1993] was developed as a common interface for communication over point-to-point links (versus the transmission of IP over packet-switched networks). PPP is a very flexible protocol that supports transmission of not only IP but OSI, DECnet, and other protocol suites. It can even carry packets from multiple protocol suites over the same link, to facilitate operation of multiprotocol internets.

PPP is often used on dialup links—enabling users to gain intermittent access to an Internet service provider via low-speed lines—and is also being used in interrouter links. Authentication of the dialup user, or of the neighbor router, is an important security concern, and is the focus of the security extensions to PPP, as described in RFC 1334 [Lloyd and Simpson 1992]. PPP allows negotiation of the (bind-time) authentication technique, with the default being no authentication. Two forms of authentication are supported currently: simple passwords and a challenge-response scheme. No provision is made for other security features that might be applicable at this layer (e.g., confidentiality of traffic or continuous authentication and integrity).

The challenge-response scheme described in the RFC is based on the use of a shared secret quantity—a unique challenge—and a one-way hash function (specifically, MD-5 [Rivest 1992]) to transform the challenge under control of the secret quantity. In principle, the challenge could be repeated during the course of a PPP association for greater assurance of association authenticity, but this facility is not usually employed. Also, in principle, this protocol can be used to provide two-way authentication so that each party verifies the identity of the other. Implementations of authenticated PPP used for interrouter communication typically do perform two-way authentication, but dialup users access via PPP typically involves only one-way authentication. PPP authentication in many respects has been one of the successes among Internet security standards to date.

IPSO

At the IP layer itself, there is currently only one security-relevant protocol standard, the Internet Protocol Security Option (IPSO) [Kent 1991]. This option carries security label information on a per-IP packet basis, typically for use with rule-based, administratively-directed, access control policies. The DoD origins of IP are evident in this option, which has undergone minor revisions over the course of a decade; RFC 1108 defines the current version of this option.

The IPSO encompasses two suboptions: the Basic Security Option (BSO) and the Extended Security Option (ESO). The BSO makes provisions for identifying the authorities responsible for security policies relative to which labels are to be interpreted. Although the field for identifying the authorities is variable in length, its one-bit-per-authority format and the limited space available for all options in an IP header effectively limit the protection authorities to a small set, with the intention that the authorities be U.S. government agencies.

The ESO top-level format allows for specification of additional security information and accommodates various nonhierarchic security labels, including multiple representation options for such labels (for efficiency). Unlike the BSO, the ESO is very extensible. The intent is that government agencies define specific label formats for use in different communities—registering the format identifiers with the Internet, but not necessarily disclosing the details of the syntax or semantics of the agency-specific labels. Of course, this approach makes it hard for the government to acquire commercial, off-the-shelf products that can manage and process ESO labels, since the requisite information would not be generally available to the vendor community.

Ultimately, the BSO is a (U.S.) DoD-specific labeling option. Its inclusion as an IP option is a holdover from the days when the Internet was funded primarily by the U.S. DoD. It is not sufficiently extensible to accommodate identification of a large number of different government or commercial organizations, and thus the option is of limited utility to the global Internet community. This has motivated the development of more readily extensible security label options outside the Internet community. However, IP security labels, in the absence of measures to verify their integrity and authenticity in the global Internet environment, may be of questionable utility.

Finally, IPSO is a good example of a security standard that suffers from the lack of an Internet security architecture. Although end systems can make use of IPSO, using IPSO with routers in a general topology is more difficult. For example, there are no provisions in OSPF to convey security label ranges associated with routes, so that a router could maintain distinct spanning trees reflecting different sensitivity levels. Similarly, BGP makes no provision for relaying label ranges as part of reachability data.

IPSP

The DoD secures TCP/IP traffic through the use of network-layer cryptography. Government cryptographic devices have been developed and deployed for over 15 years to provide just such security in packet network environments. The most recent (publicly disclosed) version of a DoD-developed network layer security protocol is SP3 [Nelson 1987]. The IP Security Protocol (IPSP) is the Internet community's approach to this sort of security in the open (versus classified) Internet environment. The goal is to provide a single security protocol that can serve the needs of many applications in the Internet. IPSP is not yet an Internet standard (it is currently under development within the IP Security WG of the IETF, so there is no definitive specification yet). Still, many of the features of IPSP are likely to parallel those of other network layer security protocols developed over the last several years, for example, SP3 and NLSP [Inf. Tech. 1993]. IPSP will be a protocol capable of encapsulating either transport layer or IP packets, implementable in either end systems or in routers, and both types of implementations will be able to interoperate.

Connectionless confidentiality will be an optional service, as will the combination of data origin authentication and connectionless integrity. IPSP may optionally include integrity-protected sequence numbers to counter replay attacks that might result in denial of service. Security Association Identifiers (SAIDs) will be used to link IPSP packets to the security attributes negotiated for each association, including the security services, algorithms, and key(s).

IPSP will make use of several forms of key management, including prearranged symmetric keys and certificate-based public-key exchanges. A companion security attribute negotiation protocol and a set of key management protocols will be defined by the same WG—but work on this task is just beginning.

Perhaps the greatest impediment to widespread use of IPSP is the difficulty of integrating an IP layer security protocol in a wide range of operating systems. Unlike application-layer security solutions, a network-layer security protocol requires careful integration into the existing network-layer protocol software (the software is part of the kernel of most Unix operating systems). Integration can be difficult, although at least one vendor of an analogous security protocol product (Hughes' NLSP-based Netlock) has addressed the problem by providing OS-specific "patches" as a means of distributing its software.

On the positive side, the developers of the next generation of IP (IPv6) have announced that security facilities will be an integral part of IPv6. The IPSP WG has somewhat redirected its efforts to produce a pair of protocols: one embedded in the IPv6 header, providing highly efficient authentication and integrity services, and one encapsulation protocol, providing services and confidentiality for either IPv4 or IPv6. This work may yield an IPSP standard by early 1995, with one or more key management protocols to follow.

Application Layer Security

The application layer has been the site of the most extensive security standards activity in the IETF, represented by two distinct approaches to application security. Privacy Enhanced Mail is an application-specific approach, whereas the output of the Common Authentication Technology WG is decidedly generic, and represents an application program interface (API) approach, rather than a concrete protocol. The results of both are described below.

PEM

Privacy Enhanced Mail (PEM) is represented by a set of four RFCs (1421–1424) that define message processing, certificate management, details for using specific cryptographic algorithms, and ancillary certificate management services. The protocol [Linn 1993a] adopts an encapsulation approach and is oriented primarily toward secure transport of messages in the RFC 822 format. However, provisions exist for extending PEM to carry messages in other formats, and work has been underway in the PEM WG for some time to extend it for use with MIME (Multipurpose Internet Mail Extensions [Borenstein 1992]).

PEM focuses on end-user security services, specifically data-origin authentication, connectionless integrity, and message-content confidentiality. All PEM-protected messages are afforded the first two services, while the last service is optional. PEM is somewhat algorithm-independent, incorporating algorithm identifiers for confidentiality, integrity, and key management; public-key cryptography is recommended for key management in support of confidentiality and for digital signatures, which provide support for nonrepudiation. Despite the focus on and the preference for use of public-key cryptography, PEM also defines means for using symmetric cryptography for the primary security services. Proposed extensions to PEM will increase algorithm independence by supporting use of separate public keys (even different algorithms) for signatures versus encryption key management.

When public-key cryptography is used with PEM, the current set of RFCs call for use of X.509 certificates and certificate revocation lists (CRLs), drawing on the work of the CCITT/ISO Directory Recommendations [Data Commun. 1988]. The PEM work proposes a certification system to support not only PEM, but also other security protocols in the Internet. In the absence of widespread deployment of directories in the Internet, the PEM RFCs also specify means for acquiring and distributing CRLs, services that are also necessary for more general use of public-key certificates in the Internet. This certification infrastructure is discussed in more detail below.

To date, PEM has not become a widely used security protocol. Several reasons contribute prominently to this lack of success. The certification infrastructure designed for PEM has been much delayed in its deployment, which has hampered growth. The vast majority of PEM implementations exhibit poor user interfaces, further discouraging use. Finally, many email users who are interested in security appear to want their existing email systems to be secure and are not willing to change email software to acquire security services. Thus the lack of PEM implementations in shrinkwrap email products further limits deployment.

GSS-API

The Generic Security Service Application Program Interface (GSS-API) [Linn 1993] is unlike most other security standards developed for the Internet. Rather than specifying security facilities for an existing protocol or a new, security-specific protocol, GSS-API is a set of interface specifications to be used by protocol developers who want to make use of integrity, authentication, and optional confidentiality services. Underneath the GSS-API, various cryptographic key management mechanisms can be employed, including symmetric systems such as Kerberos [Kohl 1993] and public-key systems such as SPX [Kaufman 1993]. The intent of GSS-API is to insulate protocol developers from the details of the underlying security mechanisms, e.g., key management systems, encryption and integrity algorithms, and even stream integrity techniques. GSS-API is itself independent of underlying communication protocols, permitting flexibility throughout the communication protocol hierarchy.

The GSS-API embodies an association-oriented flavor, and there are explicit facilities for establishing a security association and authenticating the peer entity at the other end of the association. Either one-way or two-way authentication can be required by the caller as part of initializing the security association. After a security association (called a security context in GSS-API) has been established, each packet can be afforded data origin authentication and integrity (using the Sign[4] and Verify operations) or confidentiality may be included as part of the security processing (using the Seal and Unseal operations).

All of these operations hide the underlying mechanisms from the caller by encapsulating returned arguments in "opaque" tokens that the caller passes to the other peer entity via a communication channel. The receiving peer passes the tokens to its local GSS-API implementation for processing data and status information is returned to the caller. There are provisions for a GSS-API caller to request detection of apparent packet replays or out-of-sequence delivery. These stream integrity provisions reinforce the notion that this interface is well-suited to support connection-oriented protocols, though it can also help datagram protocols dealing with replay problems. GSS-API does not include a facility for negotiation of security association parameters such as the underlying security mechanisms or even the security services. Instead, it is up to the protocol designer to determine the parameters, either through use of defaults, directory lookups based on peer identity, and so on. GSS-API makes few assumptions about the form of names used to identify peer entities, again avoiding any choices that might limit the applicability of the API. In a similar vein, GSS-API makes no assumptions about the mapping between communication channels and security associations, allowing for maximum flexibility in its use and placement in the protocol stack.

Each protocol that makes use of GSS-API involves a number of design decisions: which security services to employ, how the API is linked to the protocol environment, security attribute negotiation techniques, choice of name space, and so on. These design decisions have to be codified in individual (per-protocol) standards (RFCs). GSS-API provides the protocol designer with a framework in which to make decisions, through the standard set of interfaces for interaction with the underlying security mechanisms.

Perhaps the greatest benefits of using GSS-API accrue when a security protocol migrates from one set of underlying security mechanisms to another. In such circumstances, use of GSS-API should minimize the impact on the protocol software. However, two protocol implementations based on GSS-API, but using different mechanisms, are not made interoperable through use of GSS-API. In some respects, the GSS-API approach competes with the use of IPSP as a generic IP-layer security protocol. GSS-API may be used to create a variety of application-specific security solutions, each of which would require separate protocol specifications and result in separate development efforts. In circumstances where these applications are equally well served by reliance on a single, lower-layer security protocol, the GSS-API approach may be less desirable.

Security and Infrastructure

In addition to providing security for user data, several efforts are underway within the Internet to improve the security of the infrastructure, or to create a security infrastructure that would benefit a range of security protocols. This section examines both types of Internet security and infrastructure standards.

DNS SECURITY

The Domain Name System (DNS), which maps between host names and IP addresses, is a critical element of the Internet infrastructure. It is also an obvious point for launching attacks by spoofing the mapping information returned by a query to the DNS. For example, a source host queries the DNS to translate the name of a destination host for a Telnet

4 The choice of the name for this operation is unfortunate, since, in most cases, the security mechanism is not a digital signature but a more limited form of integrity checking.

connection. If the response is not the IP address of the intended target, but rather the address of another host (one operated by an attacker), then the source host will establish a connection to the wrong target. After this initial misdirection, the user on the source host might be tricked into providing his or her login identifier and authentication information for the legitimate target host, as well as other sensitive data. The major requirement here is that the records returned by DNS servers be verifiable in terms of integrity and data origin authentication and that some degree of timeliness be afforded. The goal is not to require trust in all DNS servers, since the set of servers is operated by a very wide range of organizations in various countries with no common administrative oversight. Rather, the primary goal is to prevent any DNS server from promulgating records that purport to be from another server, one that is authoritative for a different part of the DNS name space.

To achieve this goal, work is underway in the IETF to develop a set of security extensions to the DNS. The essence of these extensions is the application of digital signatures to the records contained in DNS servers, to permit DNS clients to verity the integrity and authenticity of the records. The signatures are not applied on behalf of DNS servers, but on behalf of the DNS domains (zones) themselves, completely removing the requirement to trust any server with regard to the integrity and authenticity of the records. It is expected that the records for a given domain will be signed off-line and then transferred to DNS servers for that domain, reducing the risk of exposure of the private key for the domain.

There is also a need to store the public keys needed to verify the signatures applied to the data records. This is essentially a form of public-key certificate, binding the domain identifier to its key, and signed by an entity that is trusted to perform binding. The obvious entity to sign a domain's certificate is the parent domain—the domain that controls the name space one level higher in the DNS tree. However, at least from the perspective of initial deployment of the system, it may be necessary to allow "cross-certification," so that domains not hierarchically related can be linked through digitally signed records.[5] In any case, this requirement gives rise to several new types of DNS records, including one that is analogous to a public-key certificate. The current DNS security proposals do not envision using the CCITT/ISO standard for these records.

The preceding makes the task sound simple, but in reality this is a complex design task. First, there are various types of records already used in the DNS, and the design must take into account the semantics of protecting each of these record types. Second, there is a strong requirement for backwards compatibility with the existing DNS infrastructure, which motivates not modifying the existing record types in any way, but defining new record types to hold signature information as well as public-key information. There is even a need to provide authenticated responses that attest to the nonexistence of data, e.g., when the response to a query indicates that the name in question does not exist in the domain.

A different form of backward compatibility problem arises because the User Datagram Protocol (UDP) is employed for many DNS queries and responses, even though TCP access is also supported. UDP is commonly configured to send and receive datagrams that are a maximum of 576 bytes, which might be too small to carry signed records. Since DNS servers are queried extensively in the Internet environment, the possibility of transforming single packet responses into multipacket ones, or of requiring TCP access rather than UDP access, is of great concern. Design efforts have focused on how to minimize the added data that will have to be transferred when DNS records are signed. However, recent analysis suggests that the 576-byte limit may not be as severe a problem as originally feared.

As noted above, work on a standard for securing DNS is in progress in the IETF and, at the time of this writing, there are two different proposals being considered. But no standards document is likely to exist until late 1994.

INTERNET CERTIFICATION INFRASTRUCTURE

As part of the PEM standards development, a public-key certification system was specified in RFC 1422 [Kent 1993b]. This system is primarily directed toward email security, but is more generally useful for applications that require public key certificates. The system makes use of the X.509 certificate and Certificate Revocation List (CRL) formats.[6] However, the Internet system goes beyond X.509 in imposing constraints on certification paths, a strict tree structure, and in defining semantics for various tiers of the tree.

Specifically, RFC 1422 calls for an agent of the Internet Society, dubbed the Internet PCA Registration Authority (IPRA), to be the root of the certification tree for the Internet and to act as the enforcer of a minimal, common policy that all lower tiers of the tree must adopt. Below the IPRA are Policy Certification Authorities (PCAs), each of which defines and publishes a certification policy that specifies security-relevant characteristics of the behavior of users and certification authorities (CAs). For example, a PCA policy specifies the procedural and technical security measures that a CA must implement in certifying users, the frequency with which CRLs are issued, and so on.

5 Use of cross certificates requires tightly constrained validation rules by DNS clients, to minimize the damage associated with compromise of a cross-certified domain. It also requires introduction and maintenance of these cross certificates by domain administrators, which may pose a significant burden for some domain administrators.

6 RFC 1422 uses the 1988 version of the X.509 certificate and defines a CRL format that was adopted by CCITT in 1993. It is likely that the newer certificate format defined in 1993 will be adopted by the Internet community in a future revision of this RFC.

On the third tier are organizational or geopolitical CAs, representing companies, universities, professional societies, states or provinces. Starting at the third tier, a strict name subordination rule is invoked, requiring that the subject name in any certificate issued by a CA at this tier (or at a lower tier) be subordinate to the name of the CA (i.e., the subject name must be lower in the X.500 directory information tree). This simple rule syntactically precludes abuses such as Company A issuing certificates for a user whose name is affiliated with Company B.

As noted above, this infrastructure is designed to produce certificates for individual email users, but could also be used to support any Internet security protocol requiring public-key certificates. The ongoing work in the DNS security arena will likely yield an alternative certification system to provide keys bound to domains and to hosts, and the names associated with these keys will be DNS names, rather than the distinguished names required in X.509 certificates. Thus the two certification systems are similar in some respects, different in others. The DNS certificates may be most appropriate for host identification, e.g., at the network layer with IPSP, when DNS names are used. The greater descriptiveness provided by well-formed distinguished names and the rich policy offered by PCAs may make the PEM-motivated certificates most appropriate for individual user identification in applications such as email and for EDI support.

SNMP SECURITY

Version 2 of the Simple Network Management Protocol (SNMPv2) incorporates several security facilities [Galvin 1993a; 1993b] designed to improve the security afforded to remotely managed devices in the Internet. Most of the SNMP implementations deployed in the Internet today are based on the predecessor protocol (SNMP), which did not initially contain any substantive security provisions. The lack of security caused SNMP to be used only for reading values from the Management Information Base (MIB) entries in these devices. Writing values to MIB entries was generally not supported due to fears that attackers would exploit the lack of security to seize control of managed devices. Security features were added to SNMP [Galvin 1992] to address these concerns, but these features were not widely implemented, perhaps because they were defined late in the lifetime of SNMP and were superseded with the advent of SNMPv2.

The security facilities of SNMPv2 are provided by means of two protocols: one for data origin authentication and integrity and one for confidentiality. The former is the base security protocol, required by the standard for all secure access to MIBs, while use of the latter is optional. Though optional, use of the second protocol is not independent; that is, confidentiality is employed only in conjunction with authenticity and integrity. Both protocols may be applied to both GET and SET (read and modify) transactions.

Each management entity interacting with a managed device is viewed as a "party" and vice versa. The access control facilities in SNMP allow different parties to be granted different access privileges for each MIB entry—which makes authentication of most parties a prerequisite for effective access control. However, one might often define an unauthenticated party and grant limited read access to some MIB entries to that party.[7] Parties that are required to be authenticated can safely be granted access to a greater range of MIB entries. Parties that are required to employ both authentication and confidentiality protocols might be granted an even broader range of access; this later configuration is necessary for a management station capable of changing private key values for parties.

There are provisions in the syntax, for the MIB associated with each party, for storage of a public and a private key for use with the authentication protocol and another pair for use with the confidentiality protocol. However, the key management described in the standard, for both the authentication and the confidentiality protocols, is based solely on the use of shared secret values, e.g., symmetric cryptoalgorithm keys. In fact, the protocol advocates use of the "public" key value as a flag to indicate when a new private key value has been set—a use of a MIB entry hardly in keeping with the entry's name. These protocols are intended to be algorithm-independent, and, to that end, the algorithms are values in the party MIB. However, the standard specifies exactly one authentication algorithm, based on the use of the MD5 algorithm, and one confidentiality algorithm, based on use of the DES algorithm. Thus, in practice, only one set of algorithms is likely to be implemented in fielded products.

The semantics of SNMP transactions do not require strict sequencing of transactions, nor detection of missing transactions, but do require an ability to match paired transactions (i.e., queries and responses and to reject replays of old transactions). The message stream integrity features of the SNMP security protocols are designed to accommodate these semantics. Protection against replay attacks or excessive delays is effected through the use of timestamps, based on clocks maintained by both the managed devices and the device managers. Each endpoint on an SNMP association maintains two clocks, representing time perceived by the local entity and by the remote entity. These clocks may have different values (due to skew), but both values are included in each message that makes use of the authentication protocol. A "lifetime" value is associated with each party, which is used to reject old messages, based on the timestamp values in the message. Receipt of an authentic, timely message is used to update the source and destination

7 In fact, clock values need to be readable without use of the authentication protocol because a remote management station needs to be able to determine the clock value of a managed device in composing a message that can be processed using the authentication protocol, e.g., initially or after loss of state by the management station.

clock values maintained by the receiver, ensuring monotonicity.

The standard provides "suggestions" for procedures to maintain clock synchronization, manage private keys, and recover from crashes. However, these suggestions are not, strictly speaking, requirements, and thus compliant implementations do not necessarily implement all of these features. Providing security in the management environment, with the goal of not losing control of managed devices due to communication errors, device failures, and so on, is a daunting goal. The many facets of this standard attest to the difficulty of the task.

ROUTING PROTOCOLS

OSPF (v2) [Moy 1994] and BGP-3 [Lougheed 1991] are the current Internet standards for intradomain and interdomain routing, respectively. Both protocols embody place holders for security features. For example, OPSF-2 includes a 64-bit field to carry authentication data. This might be used to carry the value of a keyed integrity function applied to the remainder of the packet (e.g., a DES MAC or MD-5 hash). Such a facility provides both integrity and authentication for received packets, since OSPF explicitly states that provision should be made to support different values for the authentication data for each interface. However, Appendix D of the OSPF specification defines only a password authentication function, hardly a serious authentication capability. Work is underway now to add (keyed) MD5 as an authentication option, which will increase the credibility of the place holder, but the key management required to make use of the feature remains to be defined. BGP-3 contains analogous facilities, but it does not define any authentication functions, not even passwords.

Provision of place holders for authentication data, and the corresponding processing descriptions that call for computation of this authentication data, represent a good first step toward securing routing protocols and an improvement over earlier routing protocol designs. However, a lack of detailed specification of authentication mechanisms, much less specification for corresponding key management facilities, suggests that the Internet is a long way from providing meaningful security in its routing protocols. Until these gaps are filled in the specifications, it is unlikely that implementations will be available with good authentication and integrity functions for users.

Summary

The Internet community has developed a handful of security standards since 1987, but only a few have had any significant impact on widely distributed implementations of security technology. Today, the IETF is actively engaged in the development of several additional standards that may have very widespread impact on Internet security, specifically IPSP and the DNS security extensions. Additionally, work is underway to develop a security architecture document that will provide both a framework for further development of Internet security standards and a guide to security mechanisms and security requirements for specific protocols.

The existing set of standards, as well as several of those under development, sometimes overlap or offer alternative approaches to fundamental security requirements—but there are many gaps. Ultimately, the Internet vendor and user communities will decide which standards best meet their needs; but to date these diverse communities have not been very aggressive in adopting any Internet security technology. **SV**

Bibliography

BORENSTEIN, N., and FREED, N. 1992. MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies, RFC 1341, June.

DATA COMMUNICATION NETWORKS 1988. Directory, Recommendations X.500-X.521.

DEPARTMENT of DEFENSE. 1980. DoD Standard Internet Protocol, RFC 760, Jan.

GALVIN, J., MCCLOGHRIE, K, and DAVIN, J. 1992. SNMP Security Protocols, RFC 1352, July.

GALVIN, J. and MCCLOGHRIE K. 1993a. Administrative Model for Version 2 of the Simple Network Management Protocol (SNMPv2), RFC 1445, April.

GALVIN, J. and MCCLOGHRIE, K. 1993b. Security Protocols for Version 2 of the Simple Network Management Protocol (SNMPv2), RFC 1446, April.

INFORMATION TECHNOLOGY. 1987. Open System Interconnection—Basic Reference Model, Part 2: Security Architecture. ISO 7498-2, Feb.

INFORMATION TECHNOLOGY. 1993. Open System Interconnection—Network Layer Security Protocol. ISO/IEC 11577, Nov.

INTERNET ARCHITECTURE BOARD. 1994. The Internet Standards Process—Revision 2, RFC 1602, March.

KAUFMAN, C. 1993. DASS—Distributed Authentication Security Service, RFC 1507, Sept.

KENT, S. 1991. U.S. Department of Defense Security Options for the Internet Protocol, RFC 1108, Nov.

KENT, S. 1993a. Architectural security. In *Internet System Handbook*, Jan.

KENT, S. 1993b. Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management, RFC 1422, Feb.

KOHL, J. 1993. The Kerberos Authentication Service, RFC 1510, Sept.

LINN, J. 1987. Privacy Enhancement for Internet Electronic Mail: Part I: Message Encipherment and Authentication Procedures, RFC 989, Feb.

LINN, J. 1993a. Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures, RFC 1421, Feb.

LINN, J. 1993b. Generic Security Service Application Program Interface, RFC 1508, Sept.

LLOYD, B. and SIMPSON, W. 1992. PPP Authentication Protocols, RFC 1334, Oct.

LOUGHEED, K. AND REKHTER, Y. 1991. A Border Gateway Protocol 3 (BGP-3), RFC 1267, Oct.

MOY, J. 1994. OSPF Version 2, RFC 1583, March.

NELSON, R. 1987. SDNS services and architecture. In *Proceedings of the 10th National Computer Security Conference*, Sept.

RIVEST, R. 1992. The MD5 Message Digest Algorithm, RFC 1321, April.

SIMPSON, W. 1993. The Point-to-Point Protocol (PPP), RFC 1548, Dec.