Network Working Group
Request for Comments: 5406
BCP: 146
Category: Best Current Practice

S. Bellovin
Columbia University
February 2009

Guidelines for Specifying the Use of IPsec Version 2

Status of This Memo

This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Copyright Notice

Abstract

The Security Considerations sections of many Internet Drafts say, in effect, "just use IPsec". While this is sometimes correct, more often it will leave users without real, interoperable security mechanisms. This memo offers some guidance on when IPsec Version 2 should and should not be specified.

1. Introduction

The Security Considerations sections of many Internet Drafts say, in effect, "just use IPsec". While the use of IPsec is sometimes the correct security solution, more information is needed to provide interoperable security solutions. In some cases, IPsec is unavailable in the likely endpoints. If IPsec is unavailable to -- and hence unusable by -- a majority of the users in a particular protocol environment, then the specification of IPsec is tantamount to saying "turn off security" within this community. Further, when IPsec is available, the implementation may not provide the proper granularity of protection. Finally, if IPsec is available and appropriate, the document mandating the use of IPsec needs to specify just how it is to be used.

The goal of this document is to provide guidance to protocol designers on the specification of IPsec when it is the appropriate security mechanism. The protocol specification is expected to provide realistic, interoperable security. Therefore, guidance on the configuration of the various IPsec databases, such as the Security Policy Database (SPD), is often required.

This document describes how to specify the use of IPsec Version 2 [RFC2401] including the ESPv2 (Encapsulating Security Payload version 2) [RFC2406], AHv2 (Authentication Header version 2) [RFC2402], and IKEv1 (Internet Key Exchange version 1) [RFC2409]. A separate document will describe the IPsec Version 3 suite [RFC4301] [RFC4302] [RFC4303] [RFC4306].

For further guidance on security considerations (including discussion of IPsec), see [RFC3552].

NOTE: Many of the arguments below relate to the capabilities of current implementations of IPsec. These may change over time; this advice is based on the knowledge available to the IETF at publication time.

2. WARNING

The design of security protocols is a subtle and difficult art. The cautions here about specifying the use of IPsec should NOT be taken to mean that you should invent your own new security protocol for each new application. If IPsec is a bad choice, using another standardized, well-understood security protocol will almost always give the best results for both implementation and deployment. Security protocols are very hard to design; rolling out a new one will require extensive theoretical and practical work to confirm its security properties and will incur both delay and uncertainty.

3. The Pieces of IPsec

IPsec is composed of a number of different pieces. These can be used to provide confidentiality, integrity, and replay protection; though some of these can be configured manually, generally a key management component is used. Additionally, the decision about whether and how to use IPsec is controlled by a policy database of some sort.

3.1. AH and ESP

The Authentication Header (AH) [RFC2402] and the Encapsulating Security Payload (ESP) [RFC2406] are the over-the-wire security protocols. Both provide (optional) replay protection. ESP typically is used to provide confidentiality (encryption), integrity, and authentication for traffic. ESP also can provide integrity and authentication without confidentiality, which makes it a good alternative to AH in most cases where confidentiality is not a required or desired service. Finally, ESP can be used to provide confidentiality alone, although this is not recommended [Bell96].

The difference in integrity protection offered by AH is that AH protects portions of the preceding IP header, including the source and destination address. However, if ESP is used in tunnel mode (see Section 3.2) and integrity/authentication is enabled, the IP header seen by the source and destination hosts is completely protected anyway.

AH can also protect those IP options that need to be seen by intermediate routers, but must be intact and authentic when delivered to the receiving system. At this time, use (and existence) of such IP options is extremely rare.

If an application requires such protection, and if the information to be protected cannot be inferred from the key management process, AH must be used. (ESP is generally regarded as easier to implement; however, virtually all IPsec packages support both.) If confidentiality is required, ESP must be used. It is possible to use AH in conjunction with ESP, but this combination is rarely required.

All variants of IPsec have problems with NAT boxes -- see [RFC3715] for details -- but AH is considerably more troublesome. In environments where there is substantial likelihood that the two endpoints will be separated by a NAT box -- this includes almost all services involving user-to-server traffic, as opposed to server-to-server traffic -- NAT traversal [RFC3948] should be mandated and AH should be avoided. (Note that [RFC3948] is for ESP only, and cannot be used for AH.)

3.2. Transport and Tunnel Mode

AH and ESP can both be used in either transport mode or tunnel mode. In tunnel mode, the IPsec header is followed by an inner IP header. This is the normal usage for Virtual Private Networks (VPN) and is generally required whenever either end of the IPsec-protected path is not the ultimate IP destination, e.g., when IPsec is implemented in a firewall, router, etc.

Transport mode is preferred for point-to-point communication, though tunnel mode can also be used for this purpose.

3.3. Key Management

Any cryptographic system requires key management. IPsec provides for both manual and automatic key management schemes. Manual key management is easy; however, it doesn’t scale very well. Also, IPsec’s replay protection mechanisms are not available if manual key management is used. The need for automatic key exchange is discussed in more detail in [RFC4107].

The primary automated key exchange mechanism for IPsec is the Internet Key Exchange (IKE) [RFC2409]. A new, simpler version of IKE has been approved [RFC4306], but many existing systems still use IKEv1. This document does not discuss IKEv2 and IPsecv3. A second mechanism, Kerberized Internet Negotiation of Keys (KINK) [RFC4430], has been defined. It, of course, uses Kerberos and is suitable if and only if a Kerberos infrastructure is available.

If a decision to use IKE is made, the precise mode of operation must be specified as well. IKE can be used in main mode or aggressive mode; both support digital signatures, two different ways of using public key encryption, and shared secrets for authentication.

Shared secret authentication is simpler; however, it doesn’t scale as well in many-to-many communication scenarios since each endpoint must share a unique secret with every peer with which it can communicate. Note, though, that using shared secrets in IKE is far preferable to manual keying.

In most real-world situations where public key modes of IKE are used, locally issued certificates are employed. That is, the administrator of the system or network concerned will issue certificates to all authorized users. These certificates are useful only for IPsec.

It is sometimes possible to use certificates [RFC5280] from an existing Public Key Infrastructure (PKI) with IKE. In practice, this is rare. Furthermore, not only is there no global PKI covering most

Internet endpoints, there probably never will be. Designing a structure that assumes such a PKI is a mistake. In particular, assuming that an arbitrary node will have an "authentic" certificate, issued by a mutually trusted third party and vouching for that node’s identity, is wrong. Again, such a PKI does not and probably will not exist. Public key IKE is generally a good idea, but should almost always be used with locally issued certificates as opposed to certificates from an existing PKI.

Note that public key schemes require a substantial amount of computation. Protocol designers should consider whether or not such computations are feasible on devices of interest to their clientele. Using certificates roughly doubles the number of large exponentiations that must be performed, compared with shared secret versions of IKE.

Today, even low-powered devices can generally perform enough computation to set up a limited number of security associations. Concentration points, such as firewalls or VoIP servers, may require hardware assists, especially if many peers are expected to create security associations at about the same time.

Using any automated key management mechanism can be difficult when trying to protect low-level protocols. For example, even though [RFC2461] specified the use of IPsec to protect IPv6 Neighbor Discovery, it was impossible to do key management: nodes couldn’t use IKE because it required IP-level communication, and that isn’t possible before Neighbor Discovery associations are set up.

3.4. Application Programming Interface (API)

It is, in some sense, a misnomer to speak of the API as a part of IPsec since this piece is missing on many systems. To the extent that APIs exist, they aren’t standardized. The problem is simple: there is no portable way (and often no way at all) for an application to request IPsec protection, or to tell if it was used for given inbound packets or connections.

There are additional problems:

- Applications rarely have access to such APIs. Rather, IPsec is usually configured by a system or network administrator.

- Applications are unable to verify that IPsec services are being used underneath.

- Applications are unaware of the specific identities and properties of the protected channel provided by IPsec. For instance, the IPsec key management mechanisms may be aware of the identity and authorization of the peer, but this information cannot be used by the application nor linked to application-level decisions, such as access to resources reserved to the entity identified by this identity.

Router- or firewall-based IPsec implementations pose even greater problems since there is no standardized over-the-wire protocol for communicating this information from outboard encryptors to hosts.

By contrast, higher-layer security services, such as TLS, are able to provide the necessary control and assurance.

4. Availability of IPsec in Target Devices

Although IPsec is now widely implemented and is available for current releases of most host operating systems, it is less available for embedded systems. Few hubs, network address translators, etc., implement it, especially at the low end. It is generally inappropriate to rely on IPsec when many of the endpoints are in this category.

Even for host-to-host use, IPsec availability (and experience and ease of use) has generally been for VPNs. Hosts that support IPsec for VPN use frequently do not support it on a point-to-point basis, especially via a stable, well-defined API or user interface.

Finally, few implementations support multiple layers of IPsec. If a telecommuter is using IPsec in VPN mode to access an organizational network, he or she may not be able to employ a second level of IPsec to protect an application connection to a host within the organization. (We note that such support is, in fact, mandated by Case 4 of Section 4.5 of [RFC2401]. Nevertheless, it is not widely available.) The likelihood of such deployment scenarios should be taken into account when deciding whether or not to mandate IPsec.

5. Endpoints

[RFC2401] describes many different forms of endpoint identifier. These include source addresses (both IPv4 and IPv6), host names (possibly as embedded in X.500 certificates), and user IDs (again, possibly as embedded in a certificate). Not all forms of identifier are available on all implementations; in particular, user-granularity identification is not common. This is especially a concern for multi-user systems, where it may not be possible to use different certificates to distinguish between traffic from two different users.

Again, we note that the ability to provide fine-grained protection, such as keying each connection separately and with per-user credentials, was one of the original design goals of IPsec. Nevertheless, only a few platforms support it. Indeed, some implementations do not even support using port numbers when deciding whether or not to apply IPsec protection.

6. Selectors and the SPD

Section 4.4 of [RFC2401] describes the Security Policy Database (SPD) and "selectors" used to decide what traffic should be protected by IPsec. Choices include source and destination addresses (or address ranges), protocol numbers (i.e., 6 for TCP and 17 for UDP), and port numbers for TCP and UDP. Protocols whose protection requirements cannot be described in such terms are poorer candidates for IPsec; in particular, it becomes impossible to apply protection at any finer grain than "destination host". Thus, traffic embedded in a Layer 2 Tunneling Protocol (L2TP) [RFC2661] session cannot be protected selectively by IPsec above the L2TP layer, because IPsec has no selectors defined that let it peer into the L2TP packet to find the TCP port numbers. Similarly, the Stream Control Transmission Protocol (SCTP) [RFC4960] did not exist when [RFC2401] was written; thus, protecting individual SCTP applications on the basis of port number could not be done until a new document was written [RFC3554] that defined new selectors for IPsec, and implementations appeared.

Furthermore, in a world that runs to a large extent on dynamically assigned addresses and often uses dynamically assigned port numbers as well, an all-or-nothing policy for VPNs can work well; other policies, however, can be difficult to create in any usable form.

The granularity of protection available may have side effects. If certain traffic between a pair of machines is protected by IPsec, does the implementation permit other traffic to be unprotected or protected by different policies? Alternatively, if the implementation is such that it is only capable of protecting all traffic or none, does the device have sufficient CPU capacity to encrypt everything? Note that some low-end devices may have limited secure storage capacity for keys, etc.

Implementation issues are also a concern here. As before, too many vendors have not implemented the full specification; too many IPsec implementations are not capable of using port numbers in their selectors. Protection of traffic between two hosts is thus on an all-or-nothing basis when these non-compliant implementations are employed.

7. Broadcast and Multicast

Although the designers of IPsec tried to leave room for protection of multicast traffic, a complete design wasn’t finished until much later. As such, many IPsec implementations do not support multicast. [RFC5374] describes extensions to IPsec to support it. Other relevant documents include [RFC3830], [RFC3547], and [RFC4535].

Because of the delay, protocol designers who use multicast should consider the availability of these extensions in target platforms of interest.

8. Specifying IPsec

Despite all of the caveats given above, it may still be appropriate to use IPsec in particular situations. The range of choices makes it mandatory to define precisely how IPsec is to be used. Authors of standards documents that rely on IPsec must specify the following:

a. What selectors should the initiator of the conversation (the client, in client-server architectures) use? What addresses, port numbers, etc., are to be used?

b. What IPsec protocol is to be used: AH or ESP? What mode is to be employed: transport mode or tunnel mode?

c. What form of key management is appropriate?

d. What form of identification should be used? Choices include IP address, DNS name with or without a user name, and X.500 distinguished name.

e. If the application server will switch user IDs (i.e., it is a login service of some sort) and user name identification is used, is a new security association negotiated that utilizes a user-granularity certificate? If so, when?

f. What form of authentication should be used? Choices include pre-shared secrets and certificates.

g. How are the participants authorized to perform the operations that they request? For instance, are all devices with a certificate from a particular source allowed to use any application with IPsec or access any resource? (This problem can appear with any security service, of course.)

h. Which of the many variants of IKE must be supported? Main mode? Aggressive mode?

Note that there are two different versions of IKE: IKE and IKEv2. IKEv2 is simpler and cleaner, but is not yet widely available. You must specify which version of IKE you require.

i. Is suitable IPsec support available in likely configurations of the products that would have to employ IPsec?

9. Example

Let us now work through an example based on these guidelines. We will use the Border Gateway Protocol (BGP) [RFC4271] to show how to evaluate and specify the use of IPsec for transmission security, rather than the mechanism described in [RFC2385]. Note carefully that we are not saying that IPsec is an appropriate choice here. Rather, we are demonstrating the necessary examination and specification process. Also note that the deeper security issues raised by BGP are not addressed by IPsec or any other transmission security mechanism; see [Kent00a] and [Kent00b] for more details.

Selectors

BGP runs between manually configured pairs of hosts on TCP port 179. The appropriate selector would be the pair of BGP speakers, for that port only. Note that the router’s "loopback address" is almost certainly the address to use.

Mode

Transport mode would be the proper choice if IPsec were used. The information being communicated is generally not confidential, so encryption need not be used. Either AH or ESP can be used; if ESP is used, the sender’s IP address would need to be checked against the IP address asserted in the key management exchange. (This check is mandated by [RFC2401].) For the sake of interoperability, either AH or ESP would need to be specified as mandatory to implement.

Key Management

To permit replay detection, an automated key management system should be used, most likely IKE. Again, the RFC author should pick one.

Security Policy
Connections should be accepted only from the designated peer. (Note that this restriction applies only to BGP. If the router -- or any IPsec host -- runs multiple services with different security needs, each such service requires its own security policy.)

Authentication
Given the number of BGP-speaking routers used internally by large ISPs, it is likely that shared key mechanisms are inadequate. Consequently, certificate-based IKE must be supported. However, shared secret mode is reasonable on peering links or (perhaps) on links between ISPs and customers. Whatever scheme is used, it must tie back to a source IP address or Autonomous System (AS) number in some fashion, since other BGP policies are expressed in these terms. If certificates are used, would they use IP addresses or AS numbers? Which?

Availability
For this scenario, availability is the crucial question. Do likely BGP speakers -- both backbone routers and access routers -- support the profile of IPsec described above? Will use of IPsec, with its attendant expensive cryptographic operations, raise the issue of new denial-of-service attacks? The working group and the IESG must make these determinations before deciding to use IPsec to protect BGP.

10. Security Considerations

IPsec provides transmission security and simple access control only. There are many other dimensions to protocol security that are beyond the scope of this memo, including most notably availability. For example, using IPsec does little to defend against denial-of-service attacks; in some situations, i.e., on CPU-limited systems, it may contribute to the attacks. Within its scope, the security of any resulting protocol depends heavily on the accuracy of the analysis that resulted in a decision to use IPsec.

11. Acknowledgments

Ran Atkinson, Lakshminath Dondeti, Barbara Fraser, Paul Hoffman, Russ Housley, Stephen Kent, Eric Fleischman, assorted members of the IESG, and a plethora of others have made many useful suggestions.

12. References

12.1. Normative References

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC3554] Bellovin, S., Ioannidis, J., Keromytis, A., and R. Stewart, "On the Use of Stream Control Transmission Protocol (SCTP) with IPsec", RFC 3554, July 2003.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5374] Weis, B., Gross, G., and D. Ignjatic, "Multicast Extensions to the Security Architecture for the Internet Protocol", RFC 5374, November 2008.

12.2. Informative References

[Bell96] Bellovin, S., "Problem Areas for the IP Security Protocols", Proc. Sixth Usenix Security Symposium, pp. 205-214, 1996.

[Kent00a] Kent, S., Lynn, C., and K. Seo, "Secure Border Gateway Protocol (Secure-BGP)", IEEE Journal on Selected Areas in Communications, 18:4, pp. 582-592, 2000.

[Kent00b] Kent, S., Lynn, C., Mikkelson, J., and K. Seo, "Secure Border Gateway Protocol (Secure-BGP) -- Real World Performance and Deployment Issues", Proc. Network and Distributed System Security Symposium (NDSS), 2000.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC3715] Aboba, B. and W. Dixon, "IPsec-Network Address Translation (NAT) Compatibility Requirements", RFC 3715, March 2004.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4430] Sakane, S., Kamada, K., Thomas, M., and J. Vilhuber, "Kerberized Internet Negotiation of Keys (KINK)", RFC 4430, March 2006.

[RFC4535] Harney, H., Meth, U., Colegrove, A., and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol", RFC 4535, June 2006.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

Author's Address

Steven M. Bellovin
Columbia University
1214 Amsterdam Avenue
MC 0401
New York, NY 10027
US

Phone: +1 212 939 7149
EMail: bellovin@acm.org