Network Working Group
Request for Comments: 3129
Category: Informational

M. Thomas
Cisco Systems

June 2001

Requirements for Kerberized Internet Negotiation of Keys

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

The goal of this document is to produce a streamlined, fast, easily managed, and cryptographically sound protocol without requiring public key.

Motivation

The IPsec working group has defined a number of protocols which provide the ability to create and maintain cryptographically secure security associations at layer three (i.e., the IP layer). This effort has produced two distinct protocols:

1) a mechanism to encrypt and authenticate IP datagram payloads which assumes a shared secret between the sender and receiver

2) a mechanism for IPsec peers to perform mutual authentication and exchange keying material

The IPsec working group has defined a peer to peer authentication and keying mechanism, IKE (RFC 2409). One of the drawbacks of a peer to peer protocol is that each peer must know and implement a site's security policy which in practice can be quite complex. In addition, the lack of a trusted third party requires the use of Diffie Hellman (DH) to establish a shared secret. DH, unfortunately, is computationally quite expensive and prone to denial of service attacks. IKE also relies on X.509 certificates to realize scalable authentication of peers. Digital signatures are also computationally expensive and certificate based trust models are difficult to deploy

in practice. While IKE does allow for pre-shared symmetric keys, key distribution is required between all peers -- an O(n^2) problem -- which is problematic for large deployments.

Kerberos (RFC 1510) provides a mechanism for trusted third party authentication for clients and servers. Clients authenticate to a centralized server -- the Key Distribution Center -- which in turn issues tickets that servers can decrypt thus proving that the client is who it claims to be. One of the elements of a Kerberos ticket is a session key which is generated by the KDC which may be used by the client and server to share a secret. Kerberos also allows for both symmetric key authentication, as well as certificate based public key authentication (PKinit). Since the authentication phase of Kerberos is performed by the KDC, there is no need to perform expensive DH or X.509 certificate signatures/verification operations on servers. While clients may authenticate using X.509 certificates, the authentication phase can be amortized over the lifetime of the credentials. This allows a single DH and certificate exchange to be used to key security associations with many servers in a computationally economic way. Kerberos also support clients with symmetric keys but unlike IKE, the symmetric keys are stored in the KDC making the number of keys an O(n) problem rather than O(n^2). Kerberos also allows security policy to be managed in a more centralized fashion, rather than expecting each potentially untrustworthy peer to abide by stated security policies of an organization.

The KINK working group takes these basic features of Kerberos and uses them to its advantage to create a protocol which can establish and maintain IPsec security associations (RFC 2401). It should be noted that KINK is not a replacement for IKE. IKE has one property which KINK cannot reproduce: the ability for two peers to mutually authenticate and exchange keys without the need for an actively participating third party. However, there are many situations where a trusted third party which proxies authentication is viable, and in fact desirable.

While Kerberos specifies a standard protocol between the client and the KDC to get tickets, the actual ticket exchange between client and server is application specific. KINK is intended to be an alternative to requiring each application having its own method of transporting and validating service tickets using a protocol which is efficient and tailored to the specific needs of Kerberos and the applications for which it provides keying and parameter negotiation.

Given the above, a new general keying protocol which leverages the scalability of Kerberos is desirable. The working group's first task is to define this protocol and define an domain of interpretation for

IPsec to establish and maintain IPsec security associations. The protocol must be able to take full advantage of the features of RFC 2401 but in the context of a centralized keying authority.

Requirements

KINK must meet the following requirements at a minimum:

- The protocol must use the session keys found in Kerberos tickets as the basis of the keying material used for IPsec security association keys.
- The protocol must be able to integrate into security architecture of IPsec (RFC 2401).
- The protocol must be able to start up SA's regardless of any client/server disposition in the keying protocol. In other words, either IPsec peer can be the initiator or responder, regardless of whether it's a Kerberos 'client' (TGT-only) or Kerberos 'server'(has a keytab).
- The protocol must support Kerberos using either secret key, or public key (PKINIT) initial authentication.
- The protocol must support Kerberos User-to-User mode for cases in which the initiator cannot obtain an AP_REQ for the responder (i.e. the responder is a PKINIT client) or the responder cannot decrypt and AP_REQ from the initiator (i.e., the responder doesn't have a Kerberos Keytab, just a TGT).
- The protocol must be able to allow a peer to authenticate and participate in many realms.
- The protocol must handle absolute time skew gracefully.
- The protocol must be able to create, modify, rekey, and delete security associations.
- The protocol must be capable of setting up both transport and tunnel modes of IPsec.
- The protocol must be capable of setting up both AH and ESP security associations.
- The protocol must be capable of negotiating cipher suites.
- The protocol must be capable of setting up IPsec flow selectors.

- The protocol must be capable of rekeying without the assistance of the KDC if the Kerberos session ticket is still valid.

- The protocol must make an effort to mitigate third party Denial of Service attacks (aka Zombies attacks).

- The protocol must be able to be used for more than IPsec keying.

- The protocol must support both IPv4 and IPv6.

Security Considerations

These requirements lay out input to define a protocol which allows the keying of IPsec security associations using Kerberos as the key distribution mechanism. As such, the security associations that will be created by the new protocol will inherit the union of IPsec and Kerberos's existing security weaknesses. There is no requirement to address those weaknesses unless in combination they produce a new weakness which is not inherent in other keying protocols.

Acknowledgments

The original KINK Kabal was:

Michael Thomas (Cisco)
David McGrew (Cisco)
Jan Vilhuber (Cisco)
Jonathan Trostle (Cisco)
Matt Hur (Cybersafe)
Mike Froh (Cybersafe)
Sasha Medvinsky (GI)
Derek Atkins (Telcordia)

It must also be acknowledged that the Packetcable Security specification PKT-SP-SEC-I01-991201 provided the raw fodder for this effort in its Kerberized IPsec section, and all of the focus team members who played a part in the spec. We must also acknowledge Nancy Davoust of Cablelabs for keeping order in our normally disorderly proceedings.

References

[1] Kohl, J. and C. Neuman, "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

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

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

Author's Address

Michael Thomas
Cisco Systems
375 E Tasman Rd
San Jose, Ca, 95134, USA

Phone: +1 408-525-5386
EMail: mat@cisco.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.