Kerberized Internet Negotiation of Keys

Kerberized Internet Negotiation of Keys (**KINK**) is a protocol defined in RFC 4430 used to set up an IPsec security association (SA), similar to Internet Key Exchange (IKE), utilizing the Kerberos protocol to allow trusted third parties to handle authentication of peers and management of security policies in a centralized fashion.[1]

Its motivation is given in RFC 3129 as an alternative to IKE, in which peers must each use X.509 certificates for authentication, use Diffie–Hellman key exchange (DH) for encryption, know and implement a security policy for every peer with which it will connect,[2] with authentication of the X.509 certificates either pre-arranged or using DNS, preferably with DNSSEC.[3] Utilizing Kerberos, KINK peers must only mutually authenticate with the appropriate Authentication Server (AS), with a key distribution center (KDC) in turn controlling distribution of keying material for encryption and therefore controlling the IPsec security policy.

Contents

Protocol description

KINK is a command/response protocol that can create, delete, and maintain IPsec SAs. Each command or response contains a common header along with a set of type-length-value payloads. The type of a command or a response constrains the payloads sent in the messages of the exchange.

KINK itself is a stateless protocol in that each command or response does not require storage of hard state for KINK. This is in contrast to IKE, which uses Main Mode to first establish an Internet Security Association and Key Management Protocol (ISAKMP) SA followed by subsequent Quick Mode exchanges.

KINK uses Kerberos mechanisms to provide mutual authentication and replay protection. For establishing SAs, KINK provides confidentiality for the payloads that follow the Kerberos AP-REQ payload. The design of KINK mitigates denial of service attacks by requiring authenticated exchanges before the use of any public key operations and the installation of any state. KINK also provides a means of using Kerberos User-to-User mechanisms when there is not a key shared between the server and the KDC. This is typically, but not limited to, the case with IPsec peers using PKINIT for initial authentication.

KINK directly reuses Quick Mode payloads defined in section 5.5 of IKE, with some minor changes and omissions. In most cases, KINK exchanges are a single command and its response. An optional third message is required when creating SAs, only if the responder rejects the first proposal from the initiator or wants to contribute the keying materials. KINK also provides rekeying and Dead Peer Detection.

Packet format

The KINK message includes the following fields:

KINK message

Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	type								version								length													
32	domain of interpretation (DOI)																													
64	transaction ID (XID)																													
96	next payload								A								checksum length													
128 ...	payloads ...																													
... ...	checksum ...																													

- type: CREATE, DELETE, REPLY, GETTGT, ACK, STATUS, or private use
- version: the major protocol version number
- length: length of the entire message
- domain of interpretation (DOI): a DOI as defined in the Internet Security Association and Key Management Protocol (ISAKMP)
- transaction ID (XID): identification the transaction, defined as a command, a reply, and an optional acknowledgement
- next payload: type of the first payload after the message header as KINK_DONE, KINK_AP_REQ, KINK_AP_REP, KINK_KRB_ERROR, KINK_TGT_REQ, KINK_TGT_REP, KINK_ISAKMP, KINK_ENCRYPT, or KINK_ERROR
- ACK or ACKREQ bit: 1 if responder requires an explicit acknowledgement that a REPLY was received otherwise 0
- checksum length: length in bytes of the cryptographic checksum of the message
- payloads: a list of Type/Length/Value (TLV) payloads
- checksum: Kerberos keyed checksum over the entire message excluding the checksum field itself

Payloads

KINK payloads are defined as:

KINK payload

Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	next payload																payload length													
32 ...	value ...																													

- next payload: type of the first payload
- length: length of the payload

The following payloads are defined:

- KINK_AP_REQ: a payload that relays a Kerberos AP-REQ to the responder
- KINK_AP_REP: a payload that relays a Kerberos AP-REP to the initiator
- KINK_KRB_ERROR: a payload that relays Kerberos type errors back to the initiator
- KINK_TGT_REQ: a payload that provides a means to get a TGT from the peer in order to obtain a User-to-User service ticket from the KDC
- KINK_TGT_REP: a payload that contains the TGT requested in a previous KINK_TGT_REQ payload of a GETTGT command
- KINK_ISAKMP: a payload to encapsulate the ISAKMP IKE Quick Mode (phase 2) payloads, to allow backward compatibility with IKE and ISAKMP if there are subsequent revisions
- KINK_ENCRYPT: a payload to encapsulate other KINK payloads and is encrypted using the session key and the algorithm specified by its etype
- KINK_ERROR: a payload that returns an error condition

Implementations

The following open source implementations of KINK are currently available:

- Racoon2 (http://www.racoon2.wide.ad.jp/w/) from the WIDE Project.

See also

- Internet Key Exchange

References

1. *RFC 3129: Requirements for Kerberized Internet Negotiation of Keys* (http://tools.ietf.org/html/rfc3129), Internet Engineering Task Force, June 2001, p. 2
2. *RFC 3129: Requirements for Kerberized Internet Negotiation of Keys* (http://tools.ietf.org/html/rfc3129), Internet Engineering Task Force, June 2001, p. 1
3. *RFC 4322: Opportunistic Encryption using the Internet Key Exchange (IKE)* (http://tools.ietf.org/html/rfc4322), Internet Engineering Task Force, June 2001, p. 5