Internet Key Exchange Protocol

Overview

This module introduces the IKE (Internet Key Exchange) protocol in detail and provides an in-depth description of key management in IPsec VPNs. Detailed protocol characteristics are discussed, as well as different protection mechanisms and peer authentication schemes. Peer authentication schemes protect the key management system, and are vital to the proper operation of a secure and interoperable VPN. In order to build scalable IPsec VPNs, scalable key management is needed. This module provides the student with a strong knowledge of IKE, the key management and policy agreement protocol used in IPsec VPNs.

Objectives

Upon completing this module, you will be able to:

- Identify the main purposes of the IKE protocol
- Explains how IKE interacts with IPsec
IKE Technology Introduction

Objectives

Upon completing this lesson, you will be able to:

- Describe how IKE provides key management for IPsec
- Describe two main functions of IKE—key management and policy negotiation
- Describe how IKE interacts with IPsec and its security associations (SAs)
Internet Key Exchange (IKE)

- Internet Key Exchange (RFC 2409)
- The protocol used for key management in IPsec networks
- Allows for automatic negotiation and creation of IPsec SAs between IPsec peers

The Internet Key Exchange (IKE) protocol, described in RFC 2409, is a key management protocol standard which is used in conjunction with the IPsec standard. IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard.

As mentioned in the T_IPsec chapter, IPsec security associations (SAs) must exist in order for IPsec to protect network traffic. IKE manages those SAs on behalf of IPsec, and automatically negotiates protection policies between IPsec peers.
IKE is a hybrid protocol based on:

- **ISAKMP** (RFC 2408), the protocol for negotiated establishment of security associations
- **Oakley** (RFC 2412), a key agreement/exchange protocol
- **SKEME**, another key-exchange protocol

IKE is a hybrid protocol based on the Internet Security Association and Key Management Protocol (ISAKMP), described in RFC 2408. The IKE protocol implements parts of two other key management protocols—Oakley, described in RFC 2412, and SKEME. The protection policy within SAs is negotiated and established with the help of the ISAKMP protocol, and keying material (session keys for encryption and packet authentication) is agreed on and exchanged with the use of Oakley and SKEME protocols.

**ISAKMP**—The Internet Security Association and Key Management Protocol is a protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association. ISAKMP is implemented according to the latest version of the "Internet Security Association and Key Management Protocol (ISAKMP)" standard.

**Oakley**—A key exchange protocol that defines how to derive authenticated keying material.

**Skeme**—A key exchange protocol that defines how to derive authenticated keying material, with rapid key refreshment.
The Internet Security Association and Key Management Protocol (ISAKMP) establishes a secure management session between IPsec peers, which is used to negotiate IPsec SAs. ISAKMP provides the means to do the following:

- Authenticate the remote peer
- Cryptographically protect the management session
- Exchange information for key exchange
- Negotiate all traffic protection parameters using configured security policies

Therefore, the goal of ISAKMP is the establishment of an independent security channel between authenticated peers in order to enable a secure key exchange and the negotiation of IPsec SAs between them.
Oakley is originally a free-form protocol that allows each party to proceed with the exchange at its own speed. IKE borrowed this idea from Oakley, and defines the mechanisms for key exchange in different modes over the IKE (ISAKMP) session. Each protocol produces a similar result—an authenticated key exchange, yielding trusted keying material used for IPsec SAs. Oakley, within IKE, determines AH and ESP keying material (authentication and encryption session keys) for each IPsec SA automatically, and by default uses an authenticated Diffie-Hellman algorithm to accomplish this.
Diffie-Hellman algorithm was discovered in 1976 by Whitfield Diffie and Martin Hellman. It gets its security from the difficulty of calculating the discrete logarithms of very large numbers. The Diffie-Hellman algorithm is used for secure key exchange over insecure channels and is used a lot in modern key management to provide keying material for other symmetric algorithms, such as DES or keyed-MD5 (HMAC).
In order to start a Diffie-Hellman exchange the two parties must agree on two non-secret numbers. The first is \( g \) (generator) and the second is \( p \) (modulus). Those numbers can be made public and are usually chosen from a table of known values. The generator is usually a very small number (for example, 2, 3,...), and \( p \) is a very large prime number. Every party then generates its own secret value. Then, based on \( g \), \( p \) and the secret value of each party, each party calculates its public value. The public value is computed according to the following formula:

\[
Y = g^x \mod p
\]

where \( x \) is the entity’s secret value, and \( Y \) is the entity’s public value. After that, the two parties exchange their public values. Each party then exponentiates the received public value to its secret value to compute a common shared secret. When the algorithm completes, both parties have the same shared secret which they have computed from their secret value and the public value of the other party. No one listening on the channel can compute that value, as they only know \( g \), \( p \), \( Y_A \) and \( Y_B \), and at least one secret value is needed to calculate that shared secret. Unless the attacker can compute the discrete algorithm of the above equation to recover \( x_A \) or \( x_B \), they cannot obtain the shared secret.
To follow through the algorithm in steps, here is a sequential description of the calculations involved in a Diffie-Hellman exchange:

Alice and Bob agree on generator $g$ and modulus $p$.

Alice chooses a random large integer $x(A)$ and sends Bob its public value, $Y_A$.

$$Y_A = g^{x(A)} \mod p$$

Bob chooses a random large integer $x(B)$ and sends Alice his public value, $Y_B$.

$$Y_B = g^{x(B)} \mod p$$

Alice computes:

$$k = Y_B^{x(A)} \mod p$$

Bob computes:

$$k' = Y_A^{x(B)} \mod p$$

Both $k$ and $k'$ are the equal to:

$$g^{x(A)x(B)} \mod p$$

Alice and Bob now have a shared secret ($k = k'$) and even if someone has listened on the untrusted channel, there is no way they could compute the secret from the captured information (assuming that computing a discrete logarithm of $Y_A$ or $Y_B$ is practically unfeasible, which is currently the case).
To properly protect network traffic, the IPsec process needs established security associations (SAs). If no SAs are present for a certain destination peer, IPsec will ask IKE to negotiate and create IPsec SAs on its behalf.

In order to negotiate and create IPsec SAs, the two IKE processes on both peers must first establish a secure IKE key-management session over which they will negotiate and instantiate IPsec protection policy. Because IKE negotiations must be protected, each IKE negotiation begins by each peer agreeing on a common (shared) IKE protection policy. This IKE protection policy states which security parameters will be used to protect subsequent IKE negotiations.

After the two peers agree upon an IKE protection policy, the security parameters of the policy are identified by an IKE security association (IKE SA) established at each peer. These IKE security associations apply to all subsequent IKE traffic during the negotiation.

In this protected session, IPsec SAs are then negotiated and established. With a traffic protection (IPsec SAs) policy established and proper keying material exchanged using the Diffie-Hellman method, IPsec can start to protect the network traffic. After the IPsec SAs’ lifetime expires, IKE is invoked again, and fresh IPsec SAs are created.

It is important to differentiate between the two kinds of protection policies used in IKE/IPsec networks:

- The IKE protection policy resulting in IKE SAs, defines the protection of the IKE key management session only.

The IPsec protection policy resulting in IPsec SAs, defines the protection of network traffic. These IPsec SAs are usually negotiated over IKE sessions.
This figure shows the relationship between IPsec and IKE.

When a packet to a remote peer should be protected, and no SAs for that traffic flow exist in the local SA database (SADB), IKE steps into action.

The sequence of events on the left IPsec peer is as follows:

- Alice wants to send a packet to Bob. Because no SAs are established for this specification of traffic, IPsec asks IKE to provide IPsec SAs.
- Alice’s IKE process begins the negotiation with Bob’s IKE process. This negotiation includes the establishment of a secure IKE session, the authentication of peers, and an exchange of keys for protection of this IKE session.
- IKE starts negotiating IPsec SAs. When negotiation completes, IKE uses built-in key exchange methods, such as the Diffie-Hellman algorithm, to create keying material for new IPsec SAs and to create the SAs in the local SADB. Alice and Bob now have complete SAs in place, as IKE provided them with a negotiated policy and the needed keying material.
- The packet can now be securely sent to Bob since it is protected by the IPsec SAs that are negotiated with the help of IKE.
Summary

After completing this lesson, you should be able to:

- Describe how IKE provides key management for IPsec
- Describe two main functions of IKE: key management and policy negotiation
- Describe how IKE interacts with IPsec and its security associations (SAs)

Lesson Review

1. What protocols is IKE based on?
2. When does IPsec require the assistance of IKE?
3. When is IKE invoked again after IPsec SAs have been established?
IKE Technology Description

Objectives

Upon completing this lesson, you will be able to:

- Describe the IKE protocol
- Describe various peer authentication schemes with IKE
- Describe various phases and modes of the IKE exchange and how they relate to IPsec policies
An IKE session runs over the UDP protocol with source and destination ports set to 500. When the IKE negotiation begins, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its configured policies to the remote peer. The remote peer will try to find a match by comparing its highest priority policy against the other peer's received policies. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime less than or equal to the lifetime in the policy being compared.

If an acceptable match is not found, IKE refuses negotiation and IPsec SAs will not be negotiated and established.

If a match is found, IKE will complete negotiations, create a secure IKE session based on the agreed-upon policy, and negotiate IPsec security associations over the secure IKE session.
IKE sessions are protected by cryptographic algorithms/protocols.

The peers need to agree exactly on a bundle of algorithms and protocols to protect the IKE session.

These bundles are called IKE protection suites.

IKE sessions are protected by cryptographic algorithms. IKE provides peer authentication, session integrity, and session privacy for its management session. The IKE policy defines how the IKE session should be protected, and has various parameters that are agreed upon in the initial negotiation between peers. Since some IKE messages are encrypted and authenticated, the peers must agree upon a way to encrypt and authenticate messages. Since each peer must authenticate the identity of the other, they must also agree on a way to do this. For all these negotiated parameters, IKE defines attributes and the range of values that these attributes may have. The peers must agree exactly on a bundle of algorithms and protocols to protect the IKE session.

Those bundles (encryption, hash algorithm, authentication method and Diffie-Hellman group) are called IKE protection suites.
Protection suites define bundles used to secure the IKE session

- Encryption algorithm
- Hashing MAC algorithm
- Peer authentication procedure
- DH group for initial key exchange
- SA lifetime

IKE protection suites define bundles of protection methods used to secure the IKE session.

Those methods are:

- An encryption algorithm
- A hashing MAC (HMAC) algorithm
- A Peer authentication method
- The Diffie-Hellman group used for initial key exchange
- The IKE SA lifetime (IKE session lifetime)
IKE Phases and Modes

IKE has two phases:

- **IKE phase 1**
  - Uses *main* or *aggressive* mode exchange
  - Negotiates IKE SA
- **IKE phase 2**
  - Uses *quick* mode exchange
  - Negotiates IPsec SAs

IKE protocol has two phases of operation, each of which can run in a particular *mode*:

- **IKE phase 1**
  - Uses a main or aggressive mode exchange. Used to negotiate the IKE SA (establish a secure IKE session)
- **IKE phase 2**
  - Uses a *quick* mode exchange. Negotiates IPsec SAs (negotiates and creates IPsec protection policy)
In this figure, Alice and Bob want to talk IKE. Therefore they must agree on a common IKE protection suite. The initiator (Bob) proposes several protection suites and the responder (Alice) chooses one of the offered protection suites. The selection of security policy is made by the responder according to its priorities in the configuration. In the example, Bob proposes three protection suites, and Alice chooses the second one (based on her local policy configuration). Peers must agree exactly on the protection suite. If they do not, no common policies exist between peers, and the IKE session will be terminated.
IKE phase 2 is used to negotiate and establish SAs of other protocols (IPsec’s AH and ESP, IP PCP (payload compression protocol), etc). Phase 2 needs an established IKE SA (produced in IKE phase 1 to protect the IKE session) to operate, and only operates in one defined mode, the quick mode.

The IKE initiator presents a list of (IPsec) policy proposals and the IKE responder chooses an acceptable proposal according to its locally defined policy. When the policy between peers is agreed upon, the keying material is agreed upon, and IPsec SAs are established.

In this figure, Alice and Bob want to protect their traffic with IPsec and an IKE SA is already established between them. The initiator (Bob) proposes several IPsec security policies, and the responder (Alice) chooses one of the offered policies. The selection of security policy is made by the responder according to its priorities in the configuration. In the example Bob proposes two IPsec security policies and Alice chooses one of them (the one that has the highest priority in her configuration). After successful negotiation, keying material is exchanged, and the IPsec SAs are established to protect network traffic.
IKE Peer Authentication

- To establish the IKE SA, peers have to authenticate each other (two-way)
- Three defined mechanisms
  - Pre-shared keys
  - RSA encrypted nonces
  - RSA signatures

One of the most important factors in the IKE SA negotiation is the mutual authentication of peers. Each peer must be sure that it is talking to the correct peer, before negotiating traffic protection (IPsec) policies with it. This mutual authentication is accomplished via IKE’s two-way authentication methods.

IKE provides three defined methods for two-way authentication:

- Authentication using a pre-shared secret
- Authentication using RSA encrypted nonces
- Authentication using RSA signatures

The details of these authentication methods are beyond the scope of this course.
Besides peer authentication, IKE also provides privacy and integrity of the IKE session. The IKE session can be encrypted with the use of DES or 3DES. The keying material for encryption is generally derived from the initial Diffie-Hellman exchange that occurs between peers at the start of the IKE session. When IKE is negotiating in main mode, peer identity is also encrypted, whereas aggressive mode does not hide peer identities.
IKE uses the HMAC functions to guarantee the integrity of an IKE session. Usually, a choice between keyed SHA-1 and MD5 is available. The keying material for HMAC functions is also generally derived from the initial Diffie-Hellman exchange.

IKE initially performs an ephemeral Diffie-Hellman exchange at the start of the IKE session. From that exchange, peers get shared keying material, which is then used for IKE encryption and integrity functions. The strength of that keying material can be traded off for faster performance, by choosing lower key sizes for Diffie-Hellman exchanges. The key length (strength) of Diffie-Hellman exchanges can be changed with the use of different DH groups.

When an IKE session’s lifetime expires, a new Diffie-Hellman exchange is performed between peers and the IKE SA is re-established.
Summary

After completing this lesson, you should be able to:

- Describe the IKE protocol
- Describe various peer authentication schemes with IKE
- Describe various phases and modes of the IKE exchange and how they relate to IPsec policies

Lesson Review

1. What does the IKE protocol look like on the network (transport, ports)?
2. How is the IKE protocol protected against intruders?
Summary

After completing this module, you should be able to:

- Identify the main purposes of the IKE protocol
- Explain how IKE interacts with IPsec