



Figure 9.12 IKE Formats

Figure 9.12a shows the header format for an IKE message. It consists of the following fields.

- **Initiator SPI (64 bits):** A value chosen by the initiator to identify a unique IKE security association (SA).
- **Responder SPI (64 bits):** A value chosen by the responder to identify a unique IKE SA.
- **Next Payload (8 bits):** Indicates the type of the first payload in the message; payloads are discussed in the next subsection.
- **Major Version (4 bits):** Indicates major version of IKE in use.
- **Minor Version (4 bits):** Indicates minor version in use.
- **Exchange Type (8 bits):** Indicates the type of exchange; these are discussed later in this section.
- **Flags (8 bits):** Indicates specific options set for this IKE exchange. Three bits are defined so far. The initiator bit indicates whether this packet is sent by the SA initiator. The version bit indicates whether the transmitter is capable of using a higher major version number than the one currently indicated. The response bit indicates whether this is a response to a message containing the same message ID.
- **Message ID (32 bits):** Used to control retransmission of lost packets and matching of requests and responses.
- **Length (32 bits):** Length of total message (header plus all payloads) in octets.

**IKE PAYLOAD TYPES** All IKE payloads begin with the same generic payload header shown in Figure 9.12b. The Next Payload field has a value of 0 if this is the last

payload in the message; otherwise its value is the type of the next payload. The Payload Length field indicates the length in octets of this payload, including the generic payload header.

The critical bit is 0 if the sender wants the recipient to skip this payload if it does not understand the payload type code in the Next Payload field of the previous payload. It is set to 1 if the sender wants the recipient to reject this entire message if it does not understand the payload type.

Table 9.3 summarizes the payload types defined for IKE and lists the fields, or parameters, that are part of each payload. The **SA payload** is used to begin the establishment of an SA. The payload has a complex, hierarchical structure. The payload may contain multiple proposals. Each proposal may contain multiple protocols. Each protocol may contain multiple transforms. And each transform may contain multiple attributes. These elements are formatted as substructures within the payload as follows.

- **Proposal:** This substructure includes a proposal number, a protocol ID (AH, ESP, or IKE), an indicator of the number of transforms, and then a transform substructure. If more than one protocol is to be included in a proposal, then there is a subsequent proposal substructure with the same proposal number.
- **Transform:** Different protocols support different transform types. The transforms are used primarily to define cryptographic algorithms to be used with a particular protocol.
- **Attribute:** Each transform may include attributes that modify or complete the specification of the transform. An example is key length.

**Table 9.3** IKE Payload Types

Type	Parameters
Security Association	Proposals
Key Exchange	DH Group #, Key Exchange Data
Identification	ID Type, ID Data
Certificate	Cert Encoding, Certificate Data
Certificate Request	Cert Encoding, Certification Authority
Authentication	Auth Method, Authentication Data
Nonce	Nonce Data
Notify	Protocol-ID, SPI Size, Notify Message Type, SPI, Notification Data
Delete	Protocol-ID, SPI Size, # of SPIs, SPI (one or more)
Vendor ID	Vendor ID
Traffic Selector	Number of TSs, Traffic Selectors
Encrypted	IV, Encrypted IKE payloads, Padding, Pad Length, ICV
Configuration	CFG Type, Configuration Attributes
Extensible Authentication Protocol	EAP Message

The **Key Exchange payload** can be used for a variety of key exchange techniques, including Oakley, Diffie–Hellman, and the RSA-based key exchange used by PGP. The Key Exchange data field contains the data required to generate a session key and is dependent on the key exchange algorithm used.

The **Identification payload** is used to determine the identity of communicating peers and may be used for determining authenticity of information. Typically the ID Data field will contain an IPv4 or IPv6 address.

The **Certificate payload** transfers a public-key certificate. The Certificate Encoding field indicates the type of certificate or certificate-related information, which may include the following:

- PKCS #7 wrapped X.509 certificate
- PGP certificate
- DNS signed key
- X.509 certificate—signature
- X.509 certificate—key exchange
- Kerberos tokens
- Certificate Revocation List (CRL)
- Authority Revocation List (ARL)
- SPKI certificate

At any point in an IKE exchange, the sender may include a **Certificate Request** payload to request the certificate of the other communicating entity. The payload may list more than one certificate type that is acceptable and more than one certificate authority that is acceptable.

The **Authentication** payload contains data used for message authentication purposes. The authentication method types so far defined are RSA digital signature, shared-key message integrity code, and DSS digital signature.

The **Nonce** payload contains random data used to guarantee liveness during an exchange and to protect against replay attacks.

The **Notify** payload contains either error or status information associated with this SA or this SA negotiation. The following table lists the IKE notify messages.

Error Messages	Status Messages
Unsupported Critical Payload	Initial Contact
Invalid IKE SPI	Set Window Size
Invalid Major Version	Additional TS Possible
Invalid Syntax	IPCOMP Supported
Invalid Payload Type	NAT Detection Source IP
Invalid Message ID	NAT Detection Destination IP
Invalid SPI	Cookie
	Use Transport Mode

Error Messages	Status Messages
No Proposal Chosen	HTTP Cert Lookup Supported
Invalid KE Payload	Rekey SA
Authentication Failed	ESP TFC Padding Not Supported
Single Pair Required	Non First Fragments Also
No Additional SAS	
Internal Address Failure	
Failed CP Required	
TS Unacceptable	
Invalid Selectors	

The **Delete** payload indicates one or more SAs that the sender has deleted from its database and that therefore are no longer valid.

The **Vendor ID** payload contains a vendor-defined constant. The constant is used by vendors to identify and recognize remote instances of their implementations. This mechanism allows a vendor to experiment with new features while maintaining backward compatibility.

The **Traffic Selector** payload allows peers to identify packet flows for processing by IPsec services.

The **Encrypted** payload contains other payloads in encrypted form. The encrypted payload format is similar to that of ESP. It may include an IV if the encryption algorithm requires it and an ICV if authentication is selected.

The **Configuration** payload is used to exchange configuration information between IKE peers.

The **Extensible Authentication Protocol (EAP)** payload allows IKE SAs to be authenticated using EAP, which was discussed in Chapter 5.

## 9.6 CRYPTOGRAPHIC SUITES

The IPsecv3 and IKEv3 protocols rely on a variety of types of cryptographic algorithms. As we have seen in this book, there are many cryptographic algorithms of each type, each with a variety of parameters, such as key size. To promote interoperability, two RFCs define recommended suites of cryptographic algorithms and parameters for various applications.

RFC 4308 defines two cryptographic suites for establishing virtual private networks. Suite VPN-A matches the commonly used corporate VPN security used in older IKEv1 implementations at the time of the issuance of IKEv2 in 2005. Suite VPN-B provides stronger security and is recommended for new VPNs that implement IPsecv3 and IKEv2.

Table 9.4a lists the algorithms and parameters for the two suites. There are several points to note about these two suites. Note that for symmetric cryptography,

**Table 9.4** Cryptographic Suites for IPsec

	VPN-A	VPN-B
ESP encryption	3DES-CBC	AES-CBC (128-bit key)
ESP integrity	HMAC-SHA1-96	AES-XCBC-MAC-96
IKE encryption	3DES-CBC	AES-CBC (128-bit key)
IKE PRF	HMAC-SHA1	AES-XCBC-PRF-128
IKE Integrity	HMAC-SHA1-96	AES-XCBC-MAC-96
IKE DH group	1024-bit MODP	2048-bit MODP

(a) Virtual private networks (RFC 4308)

	GCM-128	GCM-256	GMAC-128	GMAC-256
ESP encryption/ Integrity	AES-GCM (128-bit key)	AES-GCM (256-bit key)	Null	Null
ESP integrity	Null	Null	AES-GMAC (128- bit key)	AES-GMAC (256-bit key)
IKE encryption	AES-CBC (128-bit key)	AES-CBC (256-bit key)	AES-CBC (128-bit key)	AES-CBC (256-bit key)
IKE PRF	HMAC-SHA-256	HMAC-SHA-384	HMAC-SHA-256	HMAC-SHA-384
IKE Integrity	HMAC-SHA- 256-128	HMAC-SHA- 384-192	HMAC-SHA- 256-128	HMAC-SHA- 384-192
IKE DH group	256-bit random ECP	384-bit random ECP	256-bit random ECP	384-bit random ECP

(b) NSA Suite B (RFC 6379)

VPN-A relies on 3DES and HMAC, while VPN-B relies exclusively on AES. Three types of secret-key algorithms are used:

- **Encryption:** For encryption, the cipher block chaining (CBC) mode is used.
- **Message authentication:** For message authentication, VPN-A relies on HMAC with SHA-1 with the output truncated to 96 bits. VPN-B relies on a variant of CMAC with the output truncated to 96 bits.
- **Pseudorandom function:** IKEv2 generates pseudorandom bits by repeated use of the MAC used for message authentication.

RFC 6379 defines four optional cryptographic suites that are compatible with the United States National Security Agency's Suite B specifications. In 2005, the NSA issued Suite B, which defined the algorithms and strengths needed to protect both sensitive but unclassified (SBU) and classified information for use in its Cryptographic Modernization program [LATT09]. The four suites defined in RFC 6379 provide choices for ESP and IKE. The four suites are differentiated by the choice of cryptographic algorithm strengths and a choice of whether ESP is to provide both confidentiality and integrity or integrity only. All of the suites offer greater protection than the two VPN suites defined in RFC 4308.

Table 9.4b lists the algorithms and parameters for the two suites. As with RFC 4308, three categories of secret key algorithms are listed:

- **Encryption:** For ESP, authenticated encryption is provided using the GCM mode with either 128-bit or 256-bit AES keys. For IKE encryption, CBC is used, as it was for the VPN suites.
- **Message authentication:** For ESP, if only authentication is required, then a message authentication algorithm known as GMAC is used. For IKE, message authentication is provided using HMAC with one of the SHA-3 hash functions.
- **Pseudorandom function:** As with the VPN suites, IKEv2 in these suites generates pseudorandom bits by repeated use of the MAC used for message authentication.

For the Diffie–Hellman algorithm, the use of elliptic curve groups modulo a prime is specified. For authentication, elliptic curve digital signatures are listed. The original IKEv2 documents used RSA-based digital signatures. Equivalent or greater strength can be achieved using ECC with fewer key bits.

## 9.7 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

### Key Terms

anti-replay service Authentication Header (AH) Encapsulating Security Payload (ESP) Internet Key Exchange (IKE)	Internet Security Association and Key Management Protocol (ISAKMP) IP Security (IPsec) IPv4 IPv6	Oakley key determination protocol replay attack security association (SA) transport mode tunnel mode
--	---	---

### Review Questions

- 9.1 List and briefly describe some benefits of IPsec.
- 9.2 List and briefly define different categories of IPsec documents.
- 9.3 What parameters identify an SA and what parameters characterize the nature of a particular SA?
- 9.4 What is the difference between transport mode and tunnel mode?
- 9.5 What are the types of secret key algorithm used in IPsec?
- 9.6 Why does ESP include a padding field?
- 9.7 What are the basic approaches to bundling SAs?
- 9.8 What are the roles of the Oakley key determination protocol and ISAKMP in IPsec?

## Problems

- 9.1 Describe and explain each of the entries in Table 9.2.
- 9.2 Draw a figure similar to Figure 9.8 for AH.
- 9.3 List the major security services provided by AH and ESP, respectively.
- 9.4 In discussing AH processing, it was mentioned that not all of the fields in an IP header are included in MAC calculation.
  - a. For each of the fields in the IPv4 header, indicate whether the field is immutable, mutable but predictable, or mutable (zeroed prior to ICV calculation).
  - b. Do the same for the IPv6 header.
  - c. Do the same for the IPv6 extension headers.

In each case, justify your decision for each field.
- 9.5 Suppose that the current replay window spans from 120 to 530.
  - a. If the next incoming authenticated packet has sequence number 340, what will the receiver do with the packet, and what will be the parameters of the window after that?
  - b. If instead the next incoming authenticated packet has sequence number 598, what will the receiver do with the packet, and what will be the parameters of the window after that?
  - c. If instead the next incoming authenticated packet has sequence number 110, what will the receiver do with the packet, and what will be the parameters of the window after that?
- 9.6 When tunnel mode is used, a new outer IP header is constructed. For both IPv4 and IPv6, indicate the relationship of each outer IP header field and each extension header in the outer packet to the corresponding field or extension header of the inner IP packet. That is, indicate which outer values are derived from inner values and which are constructed independently of the inner values.
- 9.7 End-to-end authentication and encryption are desired between two hosts. Draw figures similar to Figure 9.8 that show each of the following.
  - a. Transport adjacency with encryption applied before authentication.
  - b. A transport SA bundled inside a tunnel SA with encryption applied before authentication.
  - c. A transport SA bundled inside a tunnel SA with authentication applied before encryption.
- 9.8 The IPsec architecture document states that when two transport mode SAs are bundled to allow both AH and ESP protocols on the same end-to-end flow, only one ordering of security protocols seems appropriate: performing the ESP protocol before performing the AH protocol. Why is this approach recommended rather than authentication before encryption?
- 9.9 For the IKE key exchange, indicate which parameters in each message go in which ISAKMP payload types.
- 9.10 Where does IPsec reside in a protocol stack?

### MALICIOUS SOFTWARE

- 10.1 Types of Malicious Software (Malware)**
- 10.2 Advanced Persistent Threats**
- 10.3 Propagation—Infected Content—Viruses**
- 10.4 Propagation—Vulnerability Exploit—Worms**
- 10.5 Propagation—Social Engineering—Spam E-mail, Trojans**
- 10.6 Payload—System Corruption**
- 10.7 Payload—Attack Agent—Zombie, Bots**
- 10.8 Payload—Information Theft—Keyloggers, Phishing, Spyware**
- 10.9 Payload—Stealth—Backdoors, Rootkits**
- 10.10 Countermeasures**
- 10.11 Distributed Denial of Service Attacks**
- 10.12 Key Terms, Review Questions, and Problems**



## LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Describe three broad mechanisms malware uses to propagate.
- Understand the basic operation of viruses, worms, and trojans.
- Describe four broad categories of malware payloads.
- Understand the different threats posed by bots, spyware, and rootkits.
- Describe some malware countermeasure elements.
- Describe three locations for malware detection mechanisms.

**Malicious software**, or **malware**, arguably constitutes one of the most significant categories of threats to computer systems. SP 800-83 (*Guide to Malware Incident Prevention and Handling for Desktops and Laptops*, July 2013) defines malware as “a program that is covertly inserted into another program with the intent to destroy data, run destructive or intrusive programs, or otherwise compromise the confidentiality, integrity, or availability of the victim’s data, applications, or operating system.” Hence, we are concerned with the threat malware poses to application programs, to utility programs, such as editors and compilers, and to kernel-level programs. We are also concerned with its use on compromised or malicious Web sites and servers, or in especially crafted spam e-mails or other messages, which aim to trick users into revealing sensitive personal information.

This chapter<sup>1</sup> examines the wide spectrum of malware threats and countermeasures. We begin with a survey of various types of malware and offer a broad classification based first on the means malware uses to spread or **propagate**, and then on the variety of actions or **payloads** used once the malware has reached a target. Propagation mechanisms include those used by viruses, worms, and trojans. Payloads include system corruption, bots, phishing, spyware, and rootkits. The discussion then includes a review of countermeasure approaches. Finally, distributed denial-of-service (DDoS) attacks are reviewed.

### 10.1 TYPES OF MALICIOUS SOFTWARE (MALWARE)

The terminology in this area presents problems because of a lack of universal agreement on all of the terms and because some of the categories overlap. Table 10.1 is a useful guide to some of the terms in use.

---

<sup>1</sup>I am indebted to Lawrie Brown of the Australian Defence Force Academy, who contributed substantially to this chapter.

**Table 10.1** Terminology for Malicious Software

Name	Description
Virus	Malware that, when executed, tries to replicate itself into other executable code; when it succeeds the code is said to be infected. When the infected code is executed, the virus also executes.
Worm	A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network.
Logic bomb	A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.
Backdoor (trapdoor)	Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality.
Mobile code	Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.
Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
Kit (virus generator)	Set of tools for generating new viruses automatically.
Spammer programs	Used to send large volumes of unwanted e-mail.
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
Zombie, bot	Program activated on an infected machine that is activated to launch attacks on other machines.
Spyware	Software that collects information from a computer and transmits it to another system.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

## A Broad Classification of Malware

Although a range of schemes can be used, one useful approach classifies malware into two broad categories, based first on how it spreads or propagates to reach the desired targets and then on the actions or payloads it performs once a target is reached.

Propagation mechanisms include infection of existing executable or interpreted content by viruses that is subsequently spread to other systems; exploit of software vulnerabilities either locally or over a network by worms or drive-by-downloads to allow the malware to replicate; and social engineering attacks that convince users to bypass security mechanisms to install trojans or to respond to phishing attacks.