reports are similar in format to aggregation reports, with these changes: ■ Receivers include as much of the message and message header as is reasonable to allow the domain to investigate the failure. Add an Identity-Alignment field, with DKIM and SPF DMARC-method fields as appropriate. ■ Optionally add a Delivery-Result field. ■ Add DKIM Domain, DKIM Identity, and DKIM selector fields, if the message was DKIM signed. Optionally also add DKIM Canonical header and body fields. ■ Add an additional DMARC authentication failure type, for use when some authentication mechanisms fail to produce aligned identifiers. 300 chapter 8 / Electronic Mail Security 8.11 Key Terms, Review Questions, and Problems Key Terms administrative management domain (ADMD) base64 Cryptographic Message Syntax (CMS) detached signature DNS-based Authentication of Named Entities (DANE) DNS Security Extensions (DNSSEC) Domain-based Message Authentication, Reporting, and Conformance (DMARC) Domain Name System (DNS) DomainKeys Identified Mail (DKIM) electronic mail Internet Mail Access Protocol (IMAP) Mail Delivery Agent (MDA) Mail Submission Agent (MSA) Message Handling Service (MHS) Message Store Message Transfer Agents (MTA) Message User Agent (MUA) Multipurpose Internet Mail Extensions (MIME) Post Office Protocol (POP3) Pretty Good Privacy (PGP) Sender Policy Framework (SPF) session key Simple Mail Transfer Protocol (SMTP) STARTTLS SUBMISSION S/MIME trust Review Questions 8.1 What types of interoperability issues are involved in internet mail architecture and how are they handled? 8.2 What are the SMTP and MIME standards? 8.3 What is the difference between a MIME content type and a MIME transfer encoding? 8.4 Briefly explain base64 encoding. 8.5 Why is base64 conversion useful for an e-mail application? 8.6 What is S/MIME? 8.7 What are the four principal services provided by S/MIME? 8.8 What is the utility of a detached signature? 8.9 What is DKIM? Problems 8.1 The character sequence "." indicates the end of mail data to a SMTP-server. What happens if the mail data itself contains that character sequence? 8.2 What are POP3 and IMAP? 8.3 If a lossless compression algorithm, such as ZIP, is used with S/MIME, why is it preferable to generate a signature before applying compression? 8.4 Before the deployment of the Domain Name System, a simple text file (HOSTS. TXT) centrally maintained at the SRI Network Information Center was used to enable mapping between host names and addresses. Each host connected to the Internet had to have an updated local copy of it to be able to use host names instead of having to cope directly with their IP addresses. Discuss the main advantages of the DNS over the old centralized HOSTS.TXT system. 8.5 For this problem and the next few, consult Appendix H. In Figure H.2, each entry in the public-key ring contains an Owner Trust field that indicates the degree of trust associated with this public-key owner. Why is that not enough? That is, if this owner is trusted and this is supposed to be the owner's public key, why is that trust not enough to permit PGP to use this public key? 8.11 / Key Terms, Review Questions, and Problems 301 8.6 What is the basic difference between X.509 and PGP in terms of key hierarchies and key trust? 8.7 In PGP, what is the expected number of session keys generated before a previously created key is produced? 8.8 A PGP user may have multiple public keys. So that a recipient knows which public key is being used by a sender, a key ID, consisting of the least significant 64 bits of the public key, is sent with the message. What is the probability that a user with N public keys will have at least one duplicate key ID? 8.9 The first 16 bits of the message digest in a PGP signature are translated in the clear. This enables the recipient to determine if the correct public key was used to decrypt the message digest by comparing this plaintext copy of the first two octets with the first two octets of the decrypted digest. a. To what extent does this compromise the security of the hash algorithm? b. To what extent does it in fact perform its intended function, namely, to

help determine if the correct RSA key was used to decrypt the digest? 8.10 Consider base64 conversion as a form of encryption. In this case, there is no key. But suppose that an opponent knew only that some form of substitution algorithm was being used to encrypt English text and did not guess that it was base64. How effective would this algorithm be against cryptanalysis? 8.11 Encode the text "ciphertext" using the following techniques. Assume characters are stored in 8-bit ASCII with zero parity. a. base64 b. Quoted-printable 8.12 Use a 2 * 2 matrix to categorize the properties of the four certificate usage models in DANE. 302 IP Security Chapter 9.1 IP Security Overview Applications of IPsec Benefits of IPsec Routing Applications IPsec Documents IPsec Services Transport and Tunnel Modes 9.2 IP Security Policy Security Associations Security Association Database Security Policy Database IP Traffic Processing 9.3 Encapsulating Security Payload ESP Format Encryption and Authentication Algorithms Padding Anti-Replay Service Transport and Tunnel Modes 9.4 Combining Security Associations Authentication Plus Confidentiality Basic Combinations of Security Associations 9.5 Internet Key Exchange Key Determination Protocol Header and Payload Formats 9.6 Cryptographic Suites 9.7 Key Terms, Review Questions, and Problems 9.1 / Ip Security Overview 303 There are application-specific security mechanisms for a number of application areas, including electronic mail (S/MIME, PGP), client/server (Kerberos), Web access (Secure Sockets Layer), and others. However, users have security concerns that cut across protocol layers. For example, an enterprise can run a secure, private IP network by disallowing links to untrusted sites, encrypting packets that leave the premises, and authenticating packets that enter the premises. By implementing security at the IP level, an organization can ensure secure networking not only for applications that have security mechanisms but also for the many security-ignorant applications. IP-level security encompasses three functional areas: authentication, confidentiality, and key management. The authentication mechanism assures that a received packet was, in fact, transmitted by the party identified as the source in the packet header. In addition, this mechanism assures that the packet has not been altered in transit. The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties. The key management facility is concerned with the secure exchange of keys. We begin this chapter with an overview of IP security (IPsec) and an introduction to the IPsec architecture. We then look at each of the three functional areas in detail. Appendix D reviews Internet protocols. 9.1 Ip Security Overview In 1994, the Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture" (RFC 1636). The report identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from Learning Objectives After studying this chapter, you should be able to: ◆◆ Present an overview of IP security (IPsec). ◆◆ Explain the difference between transport mode and tunnel mode. ◆◆ Understand the concept of security association. ◆◆ Explain the difference between the security association database and the security policy database. ◆◆ Summarize the traffic processing functions performed by IPsec for outbound packets and for inbound packets. ◆◆ Present an overview of Encapsulating Security Payload. ◆◆ Discuss the alternatives for combining security associations. ◆◆ Present an overview of Internet Key Exchange. ◆◆ Summarize the alternative cryptographic suites approved for use with IPsec. 304 chapter 9 / IP Security unauthorized monitoring and control of network traffic and the need to secure enduser-to-end-user traffic using authentication and encryption mechanisms. To provide security, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security

capabilities were designed to be usable both with the current IPv4 and the future IPv6. This means that vendors can begin offering these features now, and many vendors now do have some IPsec capability in their products. The IPsec specification now exists as a set of Internet standards. Applications of IPsec IPsec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include: ■ Secure branch office connectivity over the Internet: A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead. ■ Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters. ■ Establishing extranet and intranet connectivity with partners: IPsec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism. ■ Enhancing electronic commerce security: Even though some Web and electronic commerce applications have built-in security protocols, the use of IPsec enhances that security. IPsec guarantees that all traffic designated by the network administrator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer. The principal feature of IPsec that enables it to support these varied applications is that it can encrypt and/or authenticate all traffic at the IP level. Thus, all distributed applications (including remote logon, client/server, e-mail, file transfer, Web access, and so on) can be secured. Figure 9.1a shows a simplified packet format for an IPsec option known as tunnel mode, described subsequently. Tunnel mode makes use of an IPsec function, a combined authentication/encryption function called Encapsulating Security Payload (ESP), and a key exchange function. For VPNs, both authentication and encryption are generally desired, because it is important both to (1) assure that unauthorized users do not penetrate the VPN, and (2) assure that eavesdroppers on the Internet cannot read messages sent over the VPN. Figure 9.1b is a typical scenario of IPsec usage. An organization maintains LANs at dispersed locations. Nonsecure IP traffic is conducted on each LAN. For traffic offsite, through some sort of private or public WAN, IPsec protocols are used. These protocols operate in networking devices, such as a router or firewall, that connect each LAN to the outside world. The IPsec networking device will typically encrypt all traffic going into the WAN and decrypt traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN. Secure 9.1 / Ip Security Overview 305 transmission is also possible with individual users who dial into the WAN. Such user workstations must implement the IPsec protocols to provide security. Benefits of IPsec Some of the benefits of IPsec: ■ When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing. Figure 9.1 An IPSec VPN Scenario Networking device with IPSec Ethernet switch Unprotected IP tra c Legend: User system with IPSec (a) Tunnel-mode format (b) Example con guration Public (Internet) or private network Authenticated Encrypted ESP auth orig IP hdr IP payload ESP trlr ESP hdr IP tra c protected by IPSec Virtual tunnel: protected by IPSec New IP hdr 306 chapter 9 / IP Security ■ IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization. ■ IPsec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPsec is

implemented in the firewall or router. Even if IPsec is implemented in end systems, upper-layer software, including applications, is not affected. ■ IPsec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization. ■ IPsec can provide security for individual users if needed. This is useful for offsite workers and for setting up a secure virtual subnetwork within an organization for sensitive applications. Routing Applications In addition to supporting end users and protecting premises systems and networks, IPsec can play a vital role in the routing architecture required for internetworking. [HUIT98] lists the following examples of the use of IPsec. IPsec can assure that ■ A router advertisement (a new router advertises its presence) comes from an authorized router. ■ A neighbor advertisement (a router seeks to establish or maintain a neighbor relationship with a router in another routing domain) comes from an authorized router. ■ A redirect message comes from the router to which the initial IP packet was sent. ■ A routing update is not forged. Without such security measures, an opponent can disrupt communications or divert some traffic. Routing protocols such as Open Shortest Path First (OSPF) should be run on top of security associations between routers that are defined by IPsec. IPsec Documents IPsec encompasses three functional areas: authentication, confidentiality, and key management. The totality of the IPsec specification is scattered across dozens of RFCs and draft IETF documents, making this the most complex and difficult to grasp of all IETF specifications. The best way to grasp the scope of IPsec is to consult the latest version of the IPsec document roadmap, which as of this writing is RFC 6071 (IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap, February 2011). The documents can be categorized into the following groups. ■ Architecture: Covers the general concepts, security requirements, definitions, and mechanisms defining IPsec technology. The current specification is RFC 4301, Security Architecture for the Internet Protocol. 9.1 / Ip Security Overview 307 ■ Authentication Header (AH): AH is an extension header to provide message authentication. The current specification is RFC 4302, IP Authentication Header. Because message authentication is provided by ESP, the use of AH is deprecated. It is included in IPsecv3 for backward compatibility but should not be used in new applications. We do not discuss AH in this chapter. ■ Encapsulating Security Payload (ESP): ESP consists of an encapsulating header and trailer used to provide encryption or combined encryption/ authentication. The current specification is RFC 4303, IP Encapsulating Security Payload (ESP). ■ Internet Key Exchange (IKE): This is a collection of documents describing the key management schemes for use with IPsec. The main specification is RFC 7296, Internet Key Exchange (IKEv2) Protocol, but there are a number of related RFCs. ■ Cryptographic algorithms: This category encompasses a large set of documents that define and describe cryptographic algorithms for encryption, message authentication, pseudorandom functions (PRFs), and cryptographic key exchange. ■ Other: There are a variety of other IPsec-related RFCs, including those dealing with security policy and management information base (MIB) content. IPsec Services IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services. Two protocols are used to provide security: an authentication protocol designated by the header of the protocol, Authentication Header (AH); and a combined encryption/authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP). RFC 4301 lists the following services: ■ Access control ■ Connectionless integrity ■ Data origin authentication ■ Rejection of replayed packets (a form of partial sequence

integrity) ■ Confidentiality (encryption) ■ Limited traffic flow confidentiality Transport and Tunnel Modes Both AH and ESP support two modes of use: transport and tunnel mode. The operation of these two modes is best understood in the context of a description of ESP, which is covered in Section 9.3. Here we provide a brief overview. 308 chapter 9 / IP Security Transport Mode Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet.1 Examples include a TCP or UDP segment or an ICMP packet, all of which operate directly above IP in a host protocol stack. Typically, transport mode is used for end-to-end communication between two hosts (e.g., a client and a server, or two workstations). When a host runs AH or ESP over IPv4, the payload is the data that normally follow the IP header. For IPv6, the payload is the data that normally follow both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection. ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. AH in transport mode authenticates the IP payload and selected portions of the IP header. Tunnel Mode Tunnel mode provides protection to the entire IP packet. To achieve this, after the AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new outer IP packet with a new outer IP header. The entire original, inner, packet travels through a tunnel from one point of an IP network to another; no routers along the way are able to examine the inner IP header. Because the original packet is encapsulated, the new, larger packet may have totally different source and destination addresses, adding to the security. Tunnel mode is used when one or both ends of a security association (SA) are a security gateway, such as a firewall or router that implements IPsec. With tunnel mode, a number of hosts on networks behind firewalls may engage in secure communications without implementing IPsec. The unprotected packets generated by such hosts are tunneled through external networks by tunnel mode SAs set up by the IPsec software in the firewall or secure router at the boundary of the local network. Here is an example of how tunnel mode IPsec operates. Host A on a network generates an IP packet with the destination address of host B on another network. This packet is routed from the originating host to a firewall or secure router at the boundary of A’s network. The firewall filters all outgoing packets to determine the need for IPsec processing. If this packet from A to B requires IPsec, the firewall performs IPsec processing and encapsulates the packet with an outer IP header. The source IP address of this outer IP packet is this firewall, and the destination address may be a firewall that forms the boundary to B’s local network. This packet is now routed to B’s firewall, with intermediate routers examining only the outer IP header. At B’s firewall, the outer IP header is stripped off, and the inner packet is delivered to B. ESP in tunnel mode encrypts and optionally authenticates the entire inner IP packet, including the inner IP header. AH in tunnel mode authenticates the entire inner IP packet and selected portions of the outer IP header. Table 9.1 summarizes transport and tunnel mode functionality. 1 In this chapter, the term IP packet refers to either an IPv4 datagram or an IPv6 packet. 9.2 / Ip Security Policy 309 9.2 Ip Security Policy Fundamental to the operation of IPsec is the concept of a security policy applied to each IP packet that transits from a source to a destination. IPsec policy is determined primarily by the interaction of two databases, the security association database (SAD) and the security policy database (SPD). This section provides an overview of these two databases and then summarizes their use during IPsec operation. Figure 9.2 illustrates the relevant relationships. Security Associations A key concept that appears in both the authentication and confidentiality mechanisms for IP is the security association (SA). An association is a

one-way logical connection between a sender and a receiver that affords security services to the traffic carried on it. If a peer relationship is needed for two-way secure exchange, then two security associations are required. A security association is uniquely identified by three parameters. ■ Security Parameters Index (SPI): A 32-bit unsigned integer assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed. ■ IP Destination Address: This is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router. ■ Security Protocol Identifier: This field from the outer IP header indicates whether the association is an AH or ESP security association. Hence, in any IP packet, the security association is uniquely identified by the Destination Address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP). Transport Mode SA Tunnel Mode SA AH Authenticates IP payload and selected portions of IP header and IPv6 extension headers. Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers. ESP Encrypts IP payload and any IPv6 extension headers following the ESP header. Encrypts entire inner IP packet. ESP with Authentication Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header. Encrypts entire inner IP packet. Authenticates inner IP packet. Table 9.1 Tunnel Mode and Transport Mode Functionality 310 chapter 9 / IP Security Security Association Database In each IPsec implementation, there is a nominal2 Security Association Database that defines the parameters associated with each SA. A security association is normally defined by the following parameters in an SAD entry. ■ Security Parameter Index: A 32-bit value selected by the receiving end of an SA to uniquely identify the SA. In an SAD entry for an outbound SA, the SPI is used to construct the packet's AH or ESP header. In an SAD entry for an inbound SA, the SPI is used to map traffic to the appropriate SA. ■ Sequence Number Counter: A 32-bit value used to generate the Sequence Number field in AH or ESP headers, described in Section 9.3 (required for all implementations). ■ Sequence Counter Overflow: A flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA (required for all implementations). ■ Anti-Replay Window: Used to determine whether an inbound AH or ESP packet is a replay, described in Section 9.3 (required for all implementations). ■ AH Information: Authentication algorithm, keys, key lifetimes, and related parameters being used with AH (required for AH implementations). ■ ESP Information: Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP (required for ESP implementations). ■ Lifetime of this Security Association: A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur (required for all implementations). 2 Nominal in the sense that the functionality provided by a Security Association Database must be present in any IPsec implementation, but the way in which that functionality is provided is up to the implementer. Figure 9.2 IPsec Architecture SPD SPD SAD IKEv2 IKEv2 IPsecv3 IPsecv3 Security association database Key exchange IKE SA IPsec SA Pair ESP protects data Security association database Security policy database Security policy database SAD 9.2 / Ip Security Policy 311 ■ IPsec Protocol Mode: Tunnel, transport, or wildcard. ■ Path MTU: Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations). The key management mechanism that is used to distribute keys is coupled to the authentication and privacy mechanisms only by way of

the Security Parameters Index (SPI). Hence, authentication and privacy have been specified independent of any specific key management mechanism. IPsec provides the user with considerable flexibility in the way in which IPsec services are applied to IP traffic. As we will see later, SAs can be combined in a number of ways to yield the desired user configuration. Furthermore, IPsec provides a high degree of granularity in discriminating between traffic that is afforded IPsec protection and traffic that is allowed to bypass IPsec, as in the former case relating IP traffic to specific SAs. Security Policy Database The means by which IP traffic is related to specific SAs (or no SA in the case of traffic allowed to bypass IPsec) is the nominal Security Policy Database (SPD). In its simplest form, an SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic. In more complex environments, there may be multiple entries that potentially relate to a single SA or multiple SAs associated with a single SPD entry. The reader is referred to the relevant IPsec documents for a full discussion. Each SPD entry is defined by a set of IP and upper-layer protocol field values, called selectors. In effect, these selectors are used to filter outgoing traffic in order to map it into a particular SA. Outbound processing obeys the following general sequence for each IP packet. 1. Compare the values of the appropriate fields in the packet (the selector fields) against the SPD to find a matching SPD entry, which will point to zero or more SAs. 2. Determine the SA if any for this packet and its associated SPI. 3. Do the required IPsec processing (i.e., AH or ESP processing). The following selectors determine an SPD entry: ■ Remote IP Address: This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one destination system sharing the same SA (e.g., behind a firewall). ■ Local IP Address: This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one source system sharing the same SA (e.g., behind a firewall). ■ Next Layer Protocol: The IP protocol header (IPv4, IPv6, or IPv6 Extension) includes a field (Protocol for IPv4, Next Header for IPv6 or IPv6 Extension) that designates the protocol operating over IP. This is an individual protocol number, ANY, or for IPv6 only, OPAQUE. If AH or ESP is used, then this IP protocol header immediately precedes the AH or ESP header in the packet. 312 chapter 9 / IP Security ■ Name: A user identifier from the operating system. This is not a field in the IP or upper-layer headers but is available if IPsec is running on the same operating system as the user. ■ Local and Remote Ports: These may be individual TCP or UDP port values, an enumerated list of ports, or a wildcard port. Table 9.2 provides an example of an SPD on a host system (as opposed to a network system such as a firewall or router). This table reflects the following configuration: A local network configuration consists of two networks. The basic corporate network configuration has the IP network number 1.2.3.0/24. The local configuration also includes a secure LAN, often known as a DMZ, that is identified as 1.2.4.0/24. The DMZ is protected from both the outside world and the rest of the corporate LAN by firewalls. The host in this example has the IP address 1.2.3.10, and it is authorized to connect to the server 1.2.4.10 in the DMZ. The entries in the SPD should be self-explanatory. For example, UDP port 500 is the designated port for IKE. Any traffic from the local host to a remote host for purposes of an IKE exchange bypasses the IPsec processing. IP Traffic Processing IPsec is executed on a packet-by-packet basis. When IPsec is implemented, each outbound IP packet is processed by the IPsec logic before transmission, and each inbound packet is processed by the IPsec logic after reception and before passing the packet contents on to the next higher layer (e.g., TCP or UDP). We look at the logic of these two situations in turn. Outbound Packets Figure 9.3 highlights the

main elements of IPsec processing for outbound traffic. A block of data from a higher layer, such as TCP, is passed down to the IP layer and an IP packet is formed, consisting of an IP header and an IP body. Then the following steps occur: 1. IPsec searches the SPD for a match to this packet. 2. If no match is found, then the packet is discarded and an error message is generated. Protocol Local IP Port Remote IP Port Action Comment UDP 1.2.3.101 500 * 500 BYPASS IKE ICMP 1.2.3.101 * * * BYPASS Error messages * 1.2.3.101 * 1.2.3.0/24 * PROTECT: ESP intransport-mode Encrypt intranet traffic TCP 1.2.3.101 * 1.2.4.10 80 PROTECT: ESP intransport-mode Encrypt to server TCP 1.2.3.101 * 1.2.4.10 443 BYPASS TLS: avoid double encryption * 1.2.3.101 * 1.2.4.0/24 * DISCARD Others in DMZ * 1.2.3.101 * * * BYPASS Internet Table 9.2 Host SPD Example 9.2 / Ip Security Policy 313 3. If a match is found, further processing is determined by the first matching entry in the SPD. If the policy for this packet is DISCARD, then the packet is discarded. If the policy is BYPASS, then there is no further IPsec processing; the packet is forwarded to the network for transmission. 4. If the policy is PROTECT, then a search is made of the SAD for a matching entry. If no entry is found, then IKE is invoked to create an SA with the appropriate keys and an entry is made in the SA. 5. The matching entry in the SAD determines the processing for this packet. Either encryption, authentication, or both can be performed, and either transport or tunnel mode can be used. The packet is then forwarded to the network for transmission. Inbound Packets Figure 9.4 highlights the main elements of IPsec processing for inbound traffic. An incoming IP packet triggers the IPsec processing. The following steps occur: 1. IPsec determines whether this is an unsecured IP packet or one that has ESP or AH headers/trailers, by examining the IP Protocol field (IPv4) or Next Header field (IPv6). Figure 9.3 Processing Model for Outbound Packets Search security policy database Search security association database Determine policy Outbound IP packet (e.g., from TCP or UDP) Discard packet No match found No match found Match found Match found DISCARD PROTECT BYPASS Forward packet via IP Internet key exchange Process (AH/ESP) 314 chapter 9 / IP Security 2. If the packet is unsecured, IPsec searches the SPD for a match to this packet. If the first matching entry has a policy of BYPASS, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP. If the first matching entry has a policy of PROTECT or DISCARD, or if there is no matching entry, the packet is discarded. 3. For a secured packet, IPsec searches the SAD. If no match is found, the packet is discarded. Otherwise, IPsec applies the appropriate ESP or AH processing. Then, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP. 9.3 Encapsulating Security Payload ESP can be used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and (limited) traffic flow confidentiality. The set of services provided depends on options selected at the time of Security Association (SA) establishment and on the location of the implementation in a network topology. ESP can work with a variety of encryption and authentication algorithms, including authenticated encryption algorithms such as GCM. ESP Format Figure 9.5a shows the top-level format of an ESP packet. It contains the following fields. Figure 9.4 Processing Model for Inbound Packets Search security policy database Search security association database Packet type Inbound IP packet (from Internet) Discard packet No match found PI cesPI Not BYPASS Match BYPASS found Deliver packet to higher layer (e.g., TCP, UDP) Process (AH/ESP) 9.3 / Encapsulating Security Payload 315 ■ Security Parameters Index (32 bits): Identifies a security association. ■ Sequence Number (32 bits): A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH. ■ Payload Data (variable): This is a

transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption. ■ Padding (0–255 bytes): The purpose of this field is discussed later. ■ Pad Length (8 bits): Indicates the number of pad bytes immediately preceding this field. ■ Next Header (8 bits): Identifies the type of data contained in the payload data field by identifying the first header in that payload (e.g., an extension header in IPv6, or an upper-layer protocol such as TCP). ■ Integrity Check Value (variable): A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field. Figure 9.5 ESP Packet Format Security parameters index (SPI) 32 bits Sequence number Padding (0–255 bytes) Pad length Next header Payload data (variable) Integrity check value - ICV (variable) ICV coverage Encrypted Encrypted (a) Top-level format of an ESP Packet (b) Substructure of payload data Security parameters index (SPI) Sequence number Initialization value - IV (optional) Padding (0–255 bytes) TFC padding (optional, variable) Pad length Next header Rest of payload data (variable) Integrity check value - ICV (variable) ICV coverage Payload 316 chapter 9 / IP Security When any combined mode algorithm is employed, the algorithm itself is expected to return both decrypted plaintext and a pass/fail indication for the integrity check. For combined mode algorithms, the ICV that would normally appear at the end of the ESP packet (when integrity is selected) may be omitted. When the ICV is omitted and integrity is selected, it is the responsibility of the combined mode algorithm to encode within the Payload Data an ICV-equivalent means of verifying the integrity of the packet. Two additional fields may be present in the payload (Figure 9.5b). An initialization value (IV), or nonce, is present if this is required by the encryption or authenticated encryption algorithm used for ESP. If tunnel mode is being used, then the IPsec implementation may add traffic flow confidentiality (TFC) padding after the Payload Data and before the Padding field, as explained subsequently. Encryption and Authentication Algorithms The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service. If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an initialization vector (IV), then these data may be carried explicitly at the beginning of the Payload Data field. If included, an IV is usually not encrypted, although it is often referred to as being part of the ciphertext. The ICV field is optional. It is present only if the integrity service is selected and is provided by either a separate integrity algorithm or a combined mode algorithm that uses an ICV. The ICV is computed after the encryption is performed. This order of processing facilitates rapid detection and rejection of replayed or bogus packets by the receiver prior to decrypting the packet, hence potentially reducing the impact of denial of service (DoS) attacks. It also allows for the possibility of parallel processing of packets at the receiver that is decryption can take place in parallel with integrity checking. Note that because the ICV is not protected by encryption, a keyed integrity algorithm must be employed to compute the ICV. Padding The Padding field serves several purposes: ■ If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (e.g., the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length. ■ The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the ciphertext must be an integer multiple of 32 bits. The Padding field is used to assure this alignment. ■ Additional padding may be added to provide partial traffic-flow confidentiality by concealing the actual length of the payload. Anti-Replay Service A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of

duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence. The Sequence Number field is designed to thwart such 9.3 / Encapsulating Security Payload 317 attacks. First, we discuss sequence number generation by the sender, and then we look at how it is processed by the recipient. When a new SA is established, the sender initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1. If anti-replay is enabled (the default), the sender must not allow the sequence number to cycle past 232 - 1 back to zero. Otherwise, there would be multiple valid packets with the same sequence number. If the limit of 232 - 1 is reached, the sender should terminate this SA and negotiate a new SA with a new key. Because IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order and does not guarantee that all packets will be delivered. Therefore, the IPsec authentication document dictates that the receiver should implement a window of size W, with a default of W = 64. The right edge of the window represents the highest sequence number, N, so far received for a valid packet. For any packet with a sequence number in the range from N - W + 1 to N that has been correctly received (i.e., properly authenticated), the corresponding slot in the window is marked (Figure 9.6). Inbound processing proceeds as follows when a packet is received: 1. If the received packet falls within the window and is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked. 2. If the received packet is to the right of the window and is new, the MAC is checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked. 3. If the received packet is to the left of the window or if authentication fails, the packet is discarded; this is an auditable event. Transport and Tunnel Modes Figure 9.7 shows two ways in which the IPsec ESP service can be used. In the upper part of the figure, encryption (and optionally authentication) is provided directly between two hosts. Figure 9.7b shows how tunnel mode operation can be used to set up a virtual private network. In this example, an organization has four private networks Figure 9.6 Anti-replay Mechanism Fixed window size W N N – W N + 1 Marked if valid packet received Unmarked if valid packet not yet received • • • Advance window if valid packet to the right is received 318 chapter 9 / IP Security interconnected across the Internet. Hosts on the internal networks use the Internet for transport of data but do not interact with other Internet-based hosts. By terminating the tunnels at the security gateway to each internal network, the configuration allows the hosts to avoid implementing the security capability. The former technique is supported by a transport mode SA, while the latter technique uses a tunnel mode SA. In this section, we look at the scope of ESP for the two modes. The considerations are somewhat different for IPv4 and IPv6. We use the packet formats of Figure 9.8a as a starting point. Transport Mode ESP Transport mode ESP is used to encrypt and optionally authenticate the data carried by IP (e.g., a TCP segment), as shown in Figure 9.8b. For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (e.g., TCP, UDP, ICMP), and an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet. If authentication is selected, the ESP Authentication Data field is added after the ESP trailer. The entire transport-level segment plus the ESP trailer are encrypted. Authentication covers all of the ciphertext plus the ESP header. In the context of IPv6, ESP is viewed as an end-to-end payload; that is, it is not examined or processed by intermediate routers. Therefore, the ESP header appears after the IPv6 base header and the hop-by-hop, routing, and fragment extension headers. The

destination options extension header could appear before or after the ESP header, depending on the semantics desired. For IPv6, encryption covers Figure 9.7 Transport-Mode versus Tunnel-Mode Encryptionx Internal Network External Network Encrypted TCP Session (a) Transport-level security Internet Corporate network Corporate network Corporate network Corporate network (b) A virtual private network via tunnel mode Encrypted tunnels carrying IP trac 9.3 / Encapsulating Security Payload 319 the entire transport-level segment plus the ESP trailer plus the destination options extension header if it occurs after the ESP header. Again, authentication covers the ciphertext plus the ESP header. Transport mode operation may be summarized as follows. 1. At the source, the block of data consisting of the ESP trailer plus the entire transport-layer segment is encrypted and the plaintext of this block is replaced Figure 9.8 Scope of ESP Encryption and Authentication Orig IP hdr Hop-by-hop, dest, routing, fragment IPv6 Orig IP hdr IPv4 New IP hdr IPv4 (b) Transport Mode New IP hdr Ext headers IPv6 Authenticated Encrypted Authenticated Encrypted Authenticated Encrypted Authenticated Encrypted (c) Tunnel Mode Orig IP hdr Ext headers TCP Data ESP trlr ESP auth ESP hdr ESP auth Orig IP hdr TCP Data ESP trlr ESP auth ESP hdr Dest TCP Data TCP Data ESP trlr ESP auth ESP trlr ESP hdr ESP hdr Orig IP hdr Extension headers (if present) IPv6 TCP Data Orig IP hdr IPv4 TCP Data (a) Before Applying ESP 320 chapter 9 / IP Security with its ciphertext to form the IP packet for transmission. Authentication is added if this option is selected. 2. The packet is then routed to the destination. Each intermediate router needs to examine and process the IP header plus any plaintext IP extension headers but does not need to examine the ciphertext. 3. The destination node examines and processes the IP header plus any plaintext IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext transport-layer segment. Transport mode operation provides confidentiality for any application that uses it, thus avoiding the need to implement confidentiality in every individual application. One drawback to this mode is that it is possible to do traffic analysis on the transmitted packets. Tunnel Mode ESP Tunnel mode ESP is used to encrypt an entire IP packet (Figure 9.8c). For this mode, the ESP header is prefixed to the packet and then the packet plus the ESP trailer is encrypted. This method can be used to counter traffic analysis. Because the IP header contains the destination address and possibly source routing directives and hop-by-hop option information, it is not possible simply to transmit the encrypted IP packet prefixed by the ESP header. Intermediate routers would be unable to process such a packet. Therefore, it is necessary to encapsulate the entire block (ESP header plus ciphertext plus Authentication Data, if present) with a new IP header that will contain sufficient information for routing but not for traffic analysis. Whereas the transport mode is suitable for protecting connections between hosts that support the ESP feature, the tunnel mode is useful in a configuration that includes a firewall or other sort of security gateway that protects a trusted network from external networks. In this latter case, encryption occurs only between an external host and the security gateway or between two security gateways. This relieves hosts on the internal network of the processing burden of encryption and simplifies the key distribution task by reducing the number of needed keys. Further, it thwarts traffic analysis based on ultimate destination. Consider a case in which an external host wishes to communicate with a host on an internal network protected by a firewall, and in which ESP is implemented in the external host and the firewalls. The following steps occur for transfer of a transport-layer segment from the external host to the internal host. 1. The source prepares an inner IP packet with a destination address of the target internal host. This packet is prefixed by an ESP header;

then the packet and ESP trailer are encrypted and Authentication Data may be added. The resulting block is encapsulated with a new IP header (base header plus optional extensions such as routing and hop-by-hop options for IPv6) whose destination address is the firewall; this forms the outer IP packet. 2. The outer packet is routed to the destination firewall. Each intermediate router needs to examine and process the outer IP header plus any outer IP extension headers but does not need to examine the ciphertext. 9.3 / Encapsulating Security Payload 321 3. The destination firewall examines and processes the outer IP header plus any outer IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext inner IP packet. This packet is then transmitted in the internal network. 4. The inner packet is routed through zero or more routers in the internal network to the destination host. Figure 9.9 shows the protocol architecture for the two modes. Figure 9.9 Protocol Operation for ESP Data Data TCP hdr Data TCP hdr Data Orig IP hdr TCP hdr Data ESP trlr ESP hdr Orig IP hdr ESP auth New IP hdr TCP hdr Data ESP trlr ESP hdr Orig IP hdr ESP auth TCP hdr Data Orig IP hdr TCP hdr Data Orig IP hdr TCP hdr Data (a) Transport mode (b) Tunnel mode ESP trlr ESP hdr ESP auth Application TCP IP IPsec Application TCP IP IPsec IP 322 chapter 9 / IP Security 9.4 Combining Security Associations An individual SA can implement either the AH or ESP protocol but not both. Sometimes a particular traffic flow will call for the services provided by both AH and ESP. Further, a particular traffic flow may require IPsec services between hosts and, for that same flow, separate services between security gateways, such as firewalls. In all of these cases, multiple SAs must be employed for the same traffic flow to achieve the desired IPsec services. The term security association bundle refers to a sequence of SAs through which traffic must be processed to provide a desired set of IPsec services. The SAs in a bundle may terminate at different endpoints or at the same endpoints. Security associations may be combined into bundles in two ways: ■ Transport adjacency: Refers to applying more than one security protocol to the same IP packet without invoking tunneling. This approach to combining AH and ESP allows for only one level of combination; further nesting yields no added benefit since the processing is performed at one IPsec instance: the (ultimate) destination. ■ Iterated tunneling: Refers to the application of multiple layers of security protocols effected through IP tunneling. This approach allows for multiple levels of nesting, since each tunnel can originate or terminate at a different IPsec site along the path. The two approaches can be combined, for example, by having a transport SA between hosts travel part of the way through a tunnel SA between security gateways. One interesting issue that arises when considering SA bundles is the order in which authentication and encryption may be applied between a given pair of endpoints and the ways of doing so. We examine that issue next. Then we look at combinations of SAs that involve at least one tunnel. Authentication Plus Confidentiality Encryption and authentication can be combined in order to transmit an IP packet that has both confidentiality and authentication between hosts. We look at several approaches. ESP with Authentication Option This approach is illustrated in Figure 9.8. In this approach, the user first applies ESP to the data to be protected and then appends the authentication data field. There are actually two subcases: ■ Transport mode ESP: Authentication and encryption apply to the IP payload delivered to the host, but the IP header is not protected. ■ Tunnel mode ESP: Authentication applies to the entire IP packet delivered to the outer IP destination address (e.g., a firewall), and authentication is performed at that destination. The entire inner IP packet is protected by the privacy mechanism for delivery to the inner IP destination. For both cases, authentication applies to the ciphertext rather than the plaintext. 9.4 / Combining Security Associations 323

Transport Adjacency Another way to apply authentication after encryption is to use two bundled transport SAs, with the inner being an ESP SA and the outer being an AH SA. In this case, ESP is used without its authentication option. Because the inner SA is a transport SA, encryption is applied to the IP payload. The resulting packet consists of an IP header (and possibly IPv6 header extensions) followed by an ESP. AH is then applied in transport mode, so that authentication covers the ESP plus the original IP header (and extensions) except for mutable fields. The advantage of this approach over simply using a single ESP SA with the ESP authentication option is that the authentication covers more fields, including the source and destination IP addresses. The disadvantage is the overhead of two SAs versus one SA. Transport-Tunnel Bundle The use of authentication prior to encryption might be preferable for several reasons. First, because the authentication data are protected by encryption, it is impossible for anyone to intercept the message and alter the authentication data without detection. Second, it may be desirable to store the authentication information with the message at the destination for later reference. It is more convenient to do this if the authentication information applies to the unencrypted message; otherwise the message would have to be reencrypted to verify the authentication information. One approach to applying authentication before encryption between two hosts is to use a bundle consisting of an inner AH transport SA and an outer ESP tunnel SA. In this case, authentication is applied to the IP payload plus the IP header (and extensions) except for mutable fields. The resulting IP packet is then processed in tunnel mode by ESP; the result is that the entire, authenticated inner packet is encrypted and a new outer IP header (and extensions) is added. Basic Combinations of Security Associations The IPsec Architecture document lists four examples of combinations of SAs that must be supported by compliant IPsec hosts (e.g., workstation, server) or security gateways (e.g., firewall, router). These are illustrated in Figure 9.10. The lower part of each case in the figure represents the physical connectivity of the elements; the upper part represents logical connectivity via one or more nested SAs. Each SA can be either AH or ESP. For host-to-host SAs, the mode may be either transport or tunnel; otherwise it must be tunnel mode. Case 1. All security is provided between end systems that implement IPsec. For any two end systems to communicate via an SA, they must share the appropriate secret keys. Among the possible combinations are a. AH in transport mode b. ESP in transport mode c. ESP followed by AH in transport mode (an ESP SA inside an AH SA) d. Any one of a, b, or c inside an AH or ESP in tunnel mode We have already discussed how these various combinations can be used to support authentication, encryption, authentication before encryption, and authentication after encryption. 324 cha pter 9 / IP Secur ity Figure 9.10 Basic Combinations of Security Associations Internet Tunnel SA One or Two SAs Local Intranet Local Intranet Host* Host* Security Gateway* Security Gateway* (c) Case 3 Internet Tunnel SA Local Intranet Local Intranet Host Host Security Gateway* Security Gateway* (b) Case 2 * = implements IPsec Internet One or More SAs Local Intranet Local Intranet Host* Host* Router Router (a) Case 1 Internet Local Intranet Host* Host* Security Gateway* (d) Case 4 Tunnel SA One or Two SAs 9.5 / Internet Key Exchange 325 Case 2. Security is provided only between gateways (routers, firewalls, etc.) and no hosts implement IPsec. This case illustrates simple virtual private network support. The security architecture document specifies that only a single tunnel SA is needed for this case. The tunnel could support AH, ESP, or ESP with the authentication option. Nested tunnels are not required, because the IPsec services apply to the entire inner packet. Case 3. This builds on case 2 by adding end-to-end security. The same combinations discussed for cases 1 and 2 are allowed here. The gateway-to-gateway tunnel provides either

authentication, confidentiality, or both for all traffic between end systems. When the gateway-to-gateway tunnel is ESP, it also provides a limited form of traffic confidentiality. Individual hosts can implement any additional IPsec services required for given applications or given users by means of end-to-end SAs. Case 4. This provides support for a remote host that uses the Internet to reach an organization's firewall and then to gain access to some server or workstation behind the firewall. Only tunnel mode is required between the remote host and the firewall. As in case 1, one or two SAs may be used between the remote host and the local host. 9.5 Internet Key Exchange The key management portion of IPsec involves the determination and distribution of secret keys. A typical requirement is four keys for communication between two applications: transmit and receive pairs for both integrity and confidentiality. The IPsec Architecture document mandates support for two types of key management: ■ Manual: A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments. ■ Automated: An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration. The default automated key management protocol for IPsec is referred to as ISAKMP/Oakley and consists of the following elements: ■ Oakley Key Determination Protocol: Oakley is a key exchange protocol based on the Diffie–Hellman algorithm but providing added security. Oakley is generic in that it does not dictate specific formats. ■ Internet Security Association and Key Management Protocol (ISAKMP): ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes. ISAKMP by itself does not dictate a specific key exchange algorithm; rather, ISAKMP consists of a set of message types that enable the use of a variety of key exchange algorithms. Oakley is the specific key exchange algorithm mandated for use with the initial version of ISAKMP. 326 chapter 9 / IP Security In IKEv2, the terms Oakley and ISAKMP are no longer used, and there are significant differences from the use of Oakley and ISAKMP in IKEv1. Nevertheless, the basic functionality is the same. In this section, we describe the IKEv2 specification. Key Determination Protocol IKE key determination is a refinement of the Diffie–Hellman key exchange algorithm. Recall that Diffie–Hellman involves the following interaction between users A and B. There is prior agreement on two global parameters: q, a large prime number; and a, a primitive root of q. A selects a random integer XA as its private key and transmits to B its public key YA = aXA mod q. Similarly, B selects a random integer XB as its private key and transmits to A its public key YB = aXB mod q. Each side can now compute the secret session key: K = (YB) XA mod q = (YA) XB mod q = aXAXB mod q The Diffie–Hellman algorithm has two attractive features: ■ Secret keys are created only when needed. There is no need to store secret keys for a long period of time, exposing them to increased vulnerability. ■ The exchange requires no pre-existing infrastructure other than an agreement on the global parameters. However, there are a number of weaknesses to Diffie–Hellman, as pointed out in [HUIT98]. ■ It does not provide any information about the identities of the parties. ■ It is subject to a man-in-the-middle attack, in which a third party C impersonates B while communicating with A and impersonates A while communicating with B. Both A and B end up negotiating a key with C, which can then listen to and pass on traffic. The man-in-the-middle attack proceeds as 1. B sends his public key YB in a message addressed to A (see Figure 3.14). 2. The enemy (E) intercepts this message. E saves B's public key and sends a message to A that has B's User ID but E's public key YE. This message is sent in such a way that it appears as though it was sent from B's host system. A receives E's message and stores E's

public key with B's User ID. Similarly, E sends a message to B with E's public key, purporting to come from A. 3. B computes a secret key K1 based on B's private key and YE. A computes a secret key K2 based on A's private key and YE. E computes K1 using E's secret key XE and YB and computers K2 using XE and YA. 4. From now on, E is able to relay messages from A to B and from B to A, appropriately changing their encipherment en route in such a way that neither A nor B will know that they share their communication with E. ■ It is computationally intensive. As a result, it is vulnerable to a clogging attack, in which an opponent requests a high number of keys. The victim spends considerable computing resources doing useless modular exponentiation rather than real work. 9.5 / Internet Key Exchange 327 IKE key determination is designed to retain the advantages of Diffie–Hellman, while countering its weaknesses. Features of IKE key determination The IKE key determination algorithm is characterized by five important features: 1. It employs a mechanism known as cookies to thwart clogging attacks. 2. It enables the two parties to negotiate a group; this, in essence, specifies the global parameters of the Diffie–Hellman key exchange. 3. It uses nonces to ensure against replay attacks. 4. It enables the exchange of Diffie–Hellman public key values. 5. It authenticates the Diffie–Hellman exchange to thwart man-in-the-middle attacks. We have already discussed Diffie–Hellman. Let us look at the remainder of these elements in turn. First, consider the problem of clogging attacks. In this attack, an opponent forges the source address of a legitimate user and sends a public Diffie–Hellman key to the victim. The victim then performs a modular exponentiation to compute the secret key. Repeated messages of this type can clog the victim's system with useless work. The cookie exchange requires that each side send a pseudorandom number, the cookie, in the initial message, which the other side acknowledges. This acknowledgment must be repeated in the first message of the Diffie–Hellman key exchange. If the source address was forged, the opponent gets no answer. Thus, an opponent can only force a user to generate acknowledgments and not to perform the Diffie–Hellman calculation. IKE mandates that cookie generation satisfy three basic requirements: 1. The cookie must depend on the specific parties. This prevents an attacker from obtaining a cookie using a real IP address and UDP port and then using it to swamp the victim with requests from randomly chosen IP addresses or ports. 2. It must not be possible for anyone other than the issuing entity to generate cookies that will be accepted by that entity. This implies that the issuing entity will use local secret information in the generation and subsequent verification of a cookie. It must not be possible to deduce this secret information from any particular cookie. The point of this requirement is that the issuing entity need not save copies of its cookies, which are then more vulnerable to discovery, but can verify an incoming cookie acknowledgment when it needs to. 3. The cookie generation and verification methods must be fast to thwart attacks intended to sabotage processor resources. The recommended method for creating the cookie is to perform a fast hash (e.g., MD5) over the IP Source and Destination addresses, the UDP Source and Destination ports, and a locally generated secret value. IKE key determination supports the use of different groups for the Diffie– Hellman key exchange. Each group includes the definition of the two global parameters and the identity of the algorithm. The current specification includes the following groups. 328 chapter 9 / IP Security ■ Modular exponentiation with a 768-bit modulus q = 2768 - 2704 - 1 + 264 * (:2638 * p; + 149686) a = 2 ■ Modular exponentiation with a 1024-bit modulus q = 21024 - 2960 - 1 + 264 * (:2894 * p; + 129093) a = 2 ■ Modular exponentiation with a 1536-bit modulus – Parameters to be determined ■ Elliptic curve group over 2155 – Generator (hexadecimal): X = 7B, Y = 1C8 – Elliptic curve parameters (hexadecimal): A = 0, Y = 7338F ■ Elliptic curve

group over 2185 – Generator (hexadecimal): X = 18, Y = D – Elliptic curve parameters (hexadecimal): A = 0, Y = 1EE9 The first three groups are the classic Diffie–Hellman algorithm using modular exponentiation. The last two groups use the elliptic curve analog to Diffie–Hellman. IKE key determination employs nonces to ensure against replay attacks. Each nonce is a locally generated pseudorandom number. Nonces appear in responses and are encrypted during certain portions of the exchange to secure their use. Three different authentication methods can be used with IKE key determination: ■ Digital signatures: The exchange is authenticated by signing a mutually obtainable hash; each party encrypts the hash with its private key. The hash is generated over important parameters, such as user IDs and nonces. ■ Public-key encryption: The exchange is authenticated by encrypting parameters such as IDs and nonces with the sender's private key. ■ Symmetric-key encryption: A key derived by some out-of-band mechanism can be used to authenticate the exchange by symmetric encryption of exchange parameters. IKEv2 Exchanges The IKEv2 protocol involves the exchange of messages in pairs. The first two pairs of exchanges are referred to as the initial exchanges (Figure 9.11a). In the first exchange, the two peers exchange information concerning cryptographic algorithms and other security parameters they are willing to use along with nonces and Diffie–Hellman (DH) values. The result of this exchange is to set up a special SA called the IKE SA (see Figure 9.2). This SA defines parameters for a secure channel between the peers over which subsequent message exchanges take place. Thus, all subsequent IKE message exchanges are protected by encryption and message authentication. In the second exchange, the two parties authenticate one another and set up a first IPsec SA to be placed in the SADB and used for 9.5 / Internet Key Exchange 329 Figure 9.11 IKEv2 Exchanges HDR, SAi1, KEi, Ni Initiator Responder (a) Initial exchanges HDR, SAr1, KEr, Nr, [CERTREQ] HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr} HDR, SK {IDr, [CERT,] AUTH, SAr2, TSi, TSr} HDR, SK {[N], SA, Ni, [KEi], [TSi, TSr]} (b) CREATE_CHILD_SA exchange HDR, SK {SA, Nr, [KEr], [TSi, TSr]} HDR, SK {[N,] [D,] [CP,] ...} (c) Informational exchange HDR, SK {[N,] [D,] [CP], ...} HDR = IKE header SAx1 = o ered and chosen algorithms, DH group KEx = Di e–Hellman public key Nx = nonces CERTREQ = Certi cate request IDx = identity CERT = certi cate SK {...} = MAC and encrypt AUTH = Authentication SAx2 = algorithms, parameters for IPsec SA TSx = tra c selectors for IPsec SA N = Notify D = Delete CP = Con guration protecting ordinary (i.e. non-IKE) communications between the peers. Thus, four messages are needed to establish the first SA for general use. The CREATE_CHILD_SA exchange can be used to establish further SAs for protecting traffic. The informational exchange is used to exchange management information, IKEv2 error messages, and other notifications. Header and Payload Formats IKE defines procedures and packet formats to establish, negotiate, modify, and delete security associations. As part of SA establishment, IKE defines payloads for exchanging key generation and authentication data. These payload formats provide a consistent framework independent of the specific key exchange protocol, encryption algorithm, and authentication mechanism. IKE Header Format An IKE message consists of an IKE header followed by one or more payloads. All of this is carried in a transport protocol. The specification dictates that implementations must support the use of UDP for the transport protocol. 330 chapter 9 / IP Security 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 Figure 9.12 IKE Formats Next Payload MjVer MnVer Exchange Type Flags Message ID Length (a) IKE header (b) Generic Payload header Initiator's Security Parameter Index (SPI) Responder's Security Parameter Index (SPI) 0Bit: 8 16 24 31 Next Payload C RESERVED Payload Length Bit: 0 8 16 31 9.5 / Internet Key Exchange 331 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 Table 9.3 IKE Payload Types 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. 332 chapter 9 / IP Security 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 Initial Contact Payload Set Window Size Invalid IKE SPI Additional TS Possible Invalid Major Version IPCOMP Supported Invalid Syntax NAT Detection Source IP Invalid Payload Type NAT Detection Destination IP Invalid Message ID Cookie Invalid SPI Use Transport Mode 9.6 / Cryptographic Suites 333 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, 334 chapter 9 / IP Security 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. 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-SHA256-128 HMAC-SHA384-192 HMAC-SHA256-128 HMAC-SHA384-192 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) Table 9.4 Cryptographic Suites for IPsec 9.7 / Key Terms, Review Questions, And Problems 335 Key Terms 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 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? 336 chapter 9 / IP Security 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? 337 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—Stealthing—Backdoors, Rootkits 10.10 Countermeasures 10.11 Distributed Denial of Service Attacks 10.12 Key Terms, Review Questions, and Problems Malicious Software Part 3: SysteM Security Chapter 338 chapter 10 / Malicious Software 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 chapter1 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