Internet Key Exchange

In computing, **Internet Key Exchange** (**IKE**, sometimes **IKEv1** or **IKEv2**, depending on version) is the protocol used to set up a security association (SA) in the IPsec protocol suite. IKE builds upon the Oakley protocol and ISAKMP.[1] IKE uses X.509 certificates for authentication – either pre-shared or distributed using DNS (preferably with DNSSEC) – and a Diffie–Hellman key exchange to set up a shared session secret from which cryptographic keys are derived.[2][3] In addition, a security policy for every peer which will connect must be manually maintained.[2]

Contents

History

The Internet Engineering Task Force (IETF) originally defined IKE in November 1998 in a series of publications (Request for Comments) known as RFC 2407, RFC 2408 and RFC 2409:

- RFC 2407 defined the Internet IP Security Domain of Interpretation for ISAKMP.[4]
- RFC 2408 defined the Internet Security Association and Key Management Protocol (ISAKMP).[5]
- RFC 2409 defined the Internet Key Exchange (IKE).[6]

RFC 4306 updated IKE to version two (IKEv2) in December 2005.[7] RFC 4718 clarified some open details in October 2006.[8] RFC 5996 combined these two documents plus additional clarifications into the updated IKEv2,[9] published in September 2010. A later update upgraded the document from Proposed Standard to Internet Standard, published as RFC 7296 in October 2014.

The parent organization of the IETF, The Internet Society (ISOC), has maintained the copyrights of these standards as freely available to the Internet community.

Architecture

Most IPsec implementations consist of an IKE daemon that runs in user space and an IPsec stack in the kernel that processes the actual IP packets.

User-space daemons have easy access to mass storage containing configuration information, such as the IPsec endpoint addresses, keys and certificates, as required. Kernel modules, on the other hand, can process packets efficiently and with minimum overhead—which is important for performance reasons.

The IKE protocol uses UDP packets, usually on port 500, and generally requires 4–6 packets with 2–3 round trips to create an SA (security association) on both sides. The negotiated key material is then given to the IPsec stack. For instance, this could be an AES key, information identifying the IP endpoints and ports that are to be protected, as well as what type of IPsec tunnel has been created. The IPsec stack, in turn, intercepts the relevant IP packets if and where appropriate and performs encryption/decryption as required. Implementations vary on how the interception of the packets is done—for example, some use virtual devices, others take a slice out of the firewall, etc.

IKEv1 consists of two phases: phase 1 and phase 2.[10]

IKEv1 phases

IKE phase one's purpose is to establish a secure authenticated communication channel by using the Diffie–Hellman key exchange algorithm to generate a shared secret key to encrypt further IKE communications. This negotiation results in one single bi-directional ISAKMP Security Association (SA).[11] The authentication can be performed using either pre-shared key (shared secret), signatures, or public key encryption.[12] Phase 1 operates in either Main Mode or Aggressive Mode. Main Mode protects the identity of the peers and the hash of the shared key by encrypting them; Aggressive Mode does not.[10]

During IKE phase two, the IKE peers use the secure channel established in Phase 1 to negotiate Security Associations on behalf of other services like IPsec. The negotiation results in a minimum of two unidirectional security associations (one inbound and one outbound).[13] Phase 2 operates only in Quick Mode.[10]

Problems with IKE

Originally, IKE had numerous configuration options but lacked a general facility for automatic negotiation of a well-known default case that is universally implemented. Consequently, both sides of an IKE had to exactly agree on the type of security association they wanted to create – option by option – or a connection could not be established. Further complications arose from the fact that in many implementations the debug output was difficult to interpret, if there was any facility to produce diagnostic output at all.

The IKE specifications were open to a significant degree of interpretation, bordering on design faults (Dead-Peer-Detection being a case in point), giving rise to different IKE implementations not being able to create an agreed-upon security association at all for many combinations of options, however correctly configured they might appear at either end.

Improvements with IKEv2

The IKEv2 protocol was described in Appendix A of RFC 4306 in 2005. The following issues were addressed:

- Fewer Request For Comments (RFCs): The specifications for IKE were covered in at least three RFCs, more if one takes into account NAT traversal and other extensions that are in

common use. IKEv2 combines these in one RFC as well as making improvements to support for NAT traversal (Network Address Translation (NAT)) and firewall traversal in general.
- Standard Mobility support: There is a standard extension for IKEv2 named [rfc:4555 Mobility and Multihoming Protocol] (MOBIKE) (see also, IPsec) used to support mobility and multihoming for it and Encapsulating Security Payload (ESP). By use of this extension IKEv2 and IPsec can be used by mobile and multihomed users.
- NAT traversal: The encapsulation of IKE and ESP in User Datagram Protocol (UDP port 4500) enables these protocols to pass through a device or firewall performing NAT.[14]
- Stream Control Transmission Protocol (SCTP) support: IKEv2 allows for the SCTP protocol as used in Internet telephony protocol, Voice over IP (VoIP).
- Simple message exchange: IKEv2 has one four-message initial exchange mechanism where IKE provided eight distinctly different initial exchange mechanisms, each one of which had slight advantages and disadvantages.
- Fewer cryptographic mechanisms: IKEv2 uses cryptographic mechanisms to protect its packets that are very similar to what IPsec ESP uses to protect the IPsec packets. This led to simpler implementations and certifications for Common Criteria and FIPS 140-2 (Federal Information Processing Standard (FIPS), which require each cryptographic implementation to be separately validated.
- Reliability and State management: IKEv2 uses sequence numbers and acknowledgments to provide reliability and mandates some error processing logistics and shared state management. IKE could end up in a dead state due to the lack of such reliability measures, where both parties were expecting the other to initiate an action - which never eventuated. Work arounds (such as Dead-Peer-Detection) were developed but not standardized. This meant that different implementations of work-arounds were not always compatible.
- Denial of Service (DoS) attack resilience: IKEv2 does not perform much processing until it determines if the requester actually exists. This addressed some of the DoS problems suffered by IKE which would perform a lot of expensive cryptographic processing from spoofed locations.

Supposing **HostA** has a Security Parameter Index (SPI) of A and **HostB** has an SPI of B, the scenario would look like this:

```
HostA ------------------------------------------------- HostB
     |HDR(A,0),sai1,kei,Ni------------------------->   |
     |   <----------------------------HDR(A,0),N(cookie)|
     |HDR(A,0),N(cookie),sai1,kei,Ni---------------->   |
     |   <--------------------------HDR(A,B),SAr1,ker,Nr|
```

If **HostB** (the responder) is experiencing large amounts of half-open IKE connections, it will send an unencrypted reply message of `IKE_SA_INIT` to **HostA** (the initiator) with a notify message of type `COOKIE`, and will expect **HostA** to send an `IKE_SA_INIT` request with that cookie value in a notify payload to **HostB**. This is to ensure that the initiator is really capable of handling an IKE response from the responder.

Protocol extensions

The IETF ipsecme working group has standardized a number of extensions, with the goal of modernizing the IKEv2 protocol and adapting it better to high volume, production environments. These extensions include:

- **IKE session resumption**: the ability to resume a failed IKE/IPsec "session" after a failure, without the need to go through the entire IKE setup process (RFC 5723).
- **IKE redirect**: redirection of incoming IKE requests, allowing for simple load-balancing between multiple IKE endpoints (RFC 5685).

- **IPsec traffic visibility**: special tagging of ESP packets that are authenticated but not encrypted, with the goal of making it easier for middleboxes (such as intrusion detection systems) to analyze the flow (RFC 5840).
- **Mutual EAP authentication**: support for EAP-only (i.e., certificate-less) authentication of both of the IKE peers; the goal is to allow for modern password-based authentication methods to be used (RFC 5998).
- **Quick crash detection**: minimizing the time until an IKE peer detects that its opposite peer has crashed (RFC 6290).
- **High availability extensions**: improving IKE/IPsec-level protocol synchronization between a cluster of IPsec endpoints and a peer, to reduce the probability of dropped connections after a failover event (RFC 6311).

Implementations

IKE is supported as part of the IPsec implementation in Windows 2000, Windows XP, Windows Server 2003, Windows Vista and Windows Server 2008.[15] The ISAKMP/IKE implementation was jointly developed by Cisco and Microsoft.[16]

Microsoft Windows 7 and Windows Server 2008 R2 partially support IKEv2 (RFC 7296) as well as MOBIKE (RFC 4555) through the *VPN Reconnect* feature (also known as *Agile VPN*).

There are several open source implementations of IPsec with associated IKE capabilities. On Linux, Libreswan, Openswan and strongSwan implementations provide an IKE daemon which can configure (i.e., establish SAs) to the KLIPS or XFRM/NETKEY kernel-based IPsec stacks. XFRM/NETKEY is the Linux native IPsec implementation available as of version 2.6.

The Berkeley Software Distributions also have an IPsec implementation and IKE daemon, and most importantly a cryptographic framework (OpenBSD Cryptographic Framework, OCF), which makes supporting cryptographic accelerators much easier. OCF has recently been ported to Linux.

A significant number of network equipment vendors have created their own IKE daemons (and IPsec implementations), or license a stack from one another.

There are a number of implementations of IKEv2 and some of the companies dealing in IPsec certification and interoperability testing are starting to hold workshops for testing as well as updated certification requirements to deal with IKEv2 testing.

The following open source implementations of IKEv2 are currently available:

- OpenIKEv2 (https://github.com/OpenIKEv2),
- strongSwan,
- Libreswan,
- Openswan,
- Racoon from the KAME project,
- iked (http://man.openbsd.org/OpenBSD-current/man8/iked.8) from the OpenBSD project.,
- Rockhopper VPN Software (http://rockhoppervpn.sourceforge.net/)

Vulnerabilities

Leaked NSA presentations released by *Der Spiegel* indicate that IKE is being exploited in an unknown manner to decrypt IPSec traffic, as is ISAKMP.[17] The researchers who discovered the Logjam attack state that breaking a 1024-bit Diffie–Hellman group would break 66% of VPN servers, 18% of the top million HTTPS domains, and 26% of SSH servers, which the researchers claim is consistent with the leaks.[18] This claim was refuted by both Eyal Ronen and Adi Shamir in their paper "Critical Review of Imperfect Forward Secrecy" [19] and by Paul Wouters of Libreswan in an article "66% of VPN's are not in fact broken" [20]

IPSec VPN configurations which allow for negotiation of multiple configurations are subject to MITM-based downgrade attacks between the offered configurations, with both IKEv1 and IKEv2.[21] This can be avoided by careful segregation of client systems onto multiple service access points with stricter configurations.

Both versions of the IKE standard are susceptible to an offline dictionary attack when a low entropy password is used. For the IKEv1 this is true for main mode and aggressive mode.[22][23][24]

See also

- IPsec
- Key-agreement protocol
- Group Domain of Interpretation
- Kerberized Internet Negotiation of Keys
- Computer network

References

1. The Internet Key Exchange (IKE), RFC 2409, §1 Abstract
2. *RFC 3129: Requirements for Kerberized Internet Negotiation of Keys* (http://tools.ietf.org/html/rfc3129), Internet Engineering Task Force, June 2001, p. 1
3. *RFC 4322: Opportunistic Encryption using the Internet Key Exchange (IKE)* (http://tools.ietf.org/html/rfc4322), Internet Engineering Task Force, June 2001, p. 5
4. "RFC 2407 The Internet IP Security Domain of Interpretation for ISAKMP" (http://www.ietf.org/rfc/rfc2407.txt). Internet Engineering Task Force (IETF).
5. "RFC 2408 Internet Security Association and Key Management Protocol (ISAKMP)" (http://www.ietf.org/rfc/rfc2408.txt). Internet Engineering Task Force (IETF).
6. D. Harkins. "RFC 2409 The Internet Key Exchange (IKE)" (http://www.ietf.org/rfc/rfc2409.txt). Internet Engineering Task Force (IETF).
7. C. Kaufman (Microsoft) (December 2005). "RFC 4306 Internet Key Exchange (IKEv2) Protocol" (http://www.ietf.org/rfc/rfc4306.txt). Internet Engineering Task Force (IETF).
8. Eronen, P.; Hoffman, P. (October 2006). "RFC 4718 IKEv2 Clarifications and Implementation Guidelines" (http://www.ietf.org/rfc/rfc4718.txt). Internet Engineering Task Force (IETF).
9. Kaufman, C.; Hoffman, P.; Nir, Y.; Eronen, P. (September 2010). "RFC 5996 Internet Key Exchange (IKEv2) Protocol" (http://www.ietf.org/rfc/rfc5996.txt). Internet Engineering Task Force (IETF).
10. "RFC 2409 The Internet Key Exchange (IKE)", Internet Engineering Task Force (IETF), p. 5
11. "RFC 2409 The Internet Key Exchange (IKE)", Internet Engineering Task Force (IETF), p. 6
12. "RFC 2409 The Internet Key Exchange (IKE)", Internet Engineering Task Force (IETF), p. 10-16
13. "RFC 4306 Internet Key Exchange (IKEv2) Protocol", Internet Engineering Task Force (IETF), p. 11,33

14. "RFC 4306: Internet Key Exchange (IKEv2) Protocol", Internet Engineering Task Force (IETF), p 38-40
15. Internet Key Exchange: Internet Protocol Security (IPsec): Technet (https://technet.microsoft.com/en-us/library/cc784994(WS.10).aspx)
16. Using IPSec in Windows 2000 and XP, Part 1 (http://www.securityfocus.com/infocus/1519)
17. *Fielded Capability: End-to-end VPN SPIN9 Design Review* (http://www.spiegel.de/media/media-35529.pdf) (PDF), NSA via 'Der Spiegel', p. 5
18. Adrian, David; Bhargavan, Karthikeyan; Durumeric, Zakir; Gaudry, Pierrick; Green, Matthew; Halderman, J. Alex; Heninger, Nadia; Springall, Drew; Thomé, Emmanuel; Valenta, Luke; VanderSloot, Benjamin; Wustrow, Eric; Zanella-Béguelin, Santiago; Zimmermann, Paul (October 2015). *Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice* (https://weakdh.org/imperfect-forward-secrecy-ccs15.pdf) (PDF). 22nd ACM Conference on Computer and Communications Security (CCS '15). Denver. Retrieved 15 June 2016.
19. Ronen, Eyal; Shamir, Adi (October 2015). "Critical Review of Imperfect Forward Secrecy" (http://www.wisdom.weizmann.ac.il/~eyalro/RonenShamirDhReview.pdf) (PDF).
20. Wouters, Paul (October 2015). "66% of VPN's are not in fact broken" (https://nohats.ca/wordpress/blog/2015/10/17/66-of-vpns-are-not-in-fact-broken/).
21. Bhargavan, Karthikeyan; Brzuska, Christina; Fournet, Cédric; Kohlweiss, Markulf; Zanella-Béguelin, Santiago; Green, Matthew (January 2016). "Downgrade Resilience in Key-Exchange Protocols" (https://eprint.iacr.org/2016/072.pdf) (PDF).
22. Pliam, John (2 October 1999). "Authentication Vulnerabilities in IKE and Xauth with Weak Pre-Shared Secrets" (http://skysrv.pha.jhu.edu/~jpliam/ima/xauth/). *Johns Hopkins University*. Archived (https://web.archive.org/web/20020610050311/http://www.ima.umn.edu/~pliam/xauth/) from the original on 10 June 2002. Retrieved 5 February 2020.
23. McGrew, David (5 July 2011). "Great Cipher, But Where Did You Get That Key" (https://web.archive.org/web/20110709020412/http://blogs.cisco.com/security/great-cipher-but-where-did-you-get-that-key/). *Cisco Blog*. Archived from the original (http://blogs.cisco.com/security/great-cipher-but-where-did-you-get-that-key/) on 9 July 2011. Retrieved 11 February 2020.
24. Felsch, Dennis (August 2018). "The Dangers of Key Reuse: Practical Attacks on IPsec IKE" (https://www.usenix.org/conference/usenixsecurity18/presentation/felsch). *27th USENIX Security Symposium*. Retrieved 11 February 2020.

External links

- RFC 2407 Internet Security Association and Key Management Protocol (ISAKMP) (http://www.ietf.org/rfc/rfc2408.txt), Internet Engineering Task Force (IETF)
- RFC 2409 The Internet Key Exchange (IKE) (http://www.ietf.org/rfc/rfc2409.txt), Internet Engineering Task Force (IETF)
- RFC 7296: Internet Key Exchange Protocol Version 2 (IKEv2) (http://www.ietf.org/rfc/rfc7296.txt), Internet Engineering Task Force (IETF)
- Overview of IKE (from Cisco) (http://www.ciscopress.com/articles/article.asp?p=25474&seqNum=1)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Internet_Key_Exchange&oldid=1017750139"

This page was last edited on 14 April 2021, at 11:48 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.