IPSEC and DNSSEC

Josh Wilcox (jw14g24)

March 21, 2025

Table of Contents

IPSEC

- IPSEC Secures communication over IP networks by providing encryption, authentication and data integrity

Why IPSEC is needed

- IP networks are not typically encrypted
  - Therefore it is easy for unauthorised parties to **eavesdrop**
  - Data over a network without security can be easily intercepted
- There is no data integrity service
  - Data could be altered or tampered with without any systems to notice it
- No Authentication control
  - Attackers could impersonate legitimate users or devices

How does IPSEC Work

- How does IPSEC Work
  - Authentication Header
  - Encapsulating Security Payload
  - Internet Key Exchange
  - Modes of Sending

How does IPSEC Work - **Authentication Header**

- How does IPSEC Work
  - Authentication Header
  - Encapsulating Security Payload
  - Internet Key Exchange
  - Modes of Sending

- Attaches a cryptographic hash **built from a shared key** to the packet
- Provides data integrity by ensuring that the data has not been altered during transmission
- Provides authentication by verifying the identity of the sender
- Protects the **immutable fields** of the IP header (e.g., source and destination addresses) along with the payload
- **DOES NOT ENCRYPT DATA** - no confidentiality
  - Just a method of authentication and integrity verification
- Operates directly on the IP layer, ensuring end-to-end security
- Can be used in both **transport mode** (protects only the payload) and **tunnel mode** (protects the entire packet)
- Uses algorithms such as HMAC-SHA1 or HMAC-SHA256 for generating the cryptographic hash

How does IPSEC Work - **Encapsulating Security Payload**

- Provides **confidentiality** by encrypting the payload of the packet
- Ensures **data integrity** and **authentication** using cryptographic techniques
- Encrypts the entire payload, ensuring that the data remains private during transmission
- Can operate in both **transport mode** (encrypts only the payload) and **tunnel mode** (encrypts the entire packet, including the IP header)
- Uses encryption algorithms such as AES or 3DES for securing the data
- Can be used with or without encryption, depending on the security requirements
- Operates directly on the IP layer, ensuring end-to-end security

How does IPSEC Work - **Internet Key Exchange**

- Facilitates secure **authentication** and **key exchange** between two devices
- Establishes **Security Associations (SAs)** to define the parameters for encrypted communication
  - SAs include details such as encryption algorithms, keys, and protocols
- Ensures the **confidentiality**, **integrity**, and **authenticity** of the connection
- Operates in two phases:
  - **Phase 1:** Establishes a secure channel using either Main Mode or Aggressive Mode
  - **Phase 2:** Negotiates the SAs for data transfer using Quick Mode
- Supports multiple authentication methods, such as pre-shared keys, digital certificates, or public key encryption
- Commonly used with protocols like **IKEv1** and **IKEv2** for enhanced security and efficiency

How does IPSEC Work - **Modes of Sending**

- **Transport Mode**
  - Only the **payload** (data) of the IP packet is encrypted and/or authenticated.
  - The original IP header remains intact and visible, allowing intermediate devices (e.g., routers) to process the packet.
  - Commonly used for end-to-end communication between two devices.
  - Provides lower overhead compared to Tunnel Mode, as only the payload is secured.
  - Suitable for scenarios where the source and destination devices are directly communicating and trust each other.
- **Tunnel Mode**
  - The entire IP packet, including the original IP header, is encapsulated within a new IP packet.
  - The new outer IP header is added, and the entire inner packet is encrypted and/or authenticated.
  - Commonly used for communication between gateways (e.g., VPNs) or between a gateway and a device.
  - Ensures that the original packet is completely hidden, providing an additional layer of security.
  - Suitable for scenarios where traffic needs to traverse untrusted networks, such as the internet.
  - Adds more overhead compared to Transport Mode due to the encapsulation of the entire packet.

DNSSEC

Why do we need DNSSEC

- DNS provides no authenticity or integrity
  - An attacker could **divert traffic** to its own servers by forging DNS responses.
- An attacker could impersonate a resolver and **return false DNS records**
  - This is known as **DNS spoofing**, where users are directed to malicious websites.
- An attacker could forge responses from an authoritative server and poison a DNS cache
  - **DNS cache poisoning** involves injecting false DNS records into a resolver's cache.
  - This causes the resolver to return incorrect IP addresses for legitimate domain names.
  - Users are unknowingly redirected to malicious websites, enabling phishing, malware distribution, or data theft.
  - Attackers often exploit vulnerabilities in the DNS protocol, such as weak transaction ID generation or lack of source port randomization.

How does DNSSEC WORK

- Uses public-key cryptography to digitally **sign** DNS records
  - Each DNS zone has a pair of cryptographic keys: a private key for signing and a public key for verification.
  - The private key is used to generate a digital signature for DNS records.
  - The public key is published in the DNS and used by resolvers to verify the authenticity of the records.
- Provides **Authenticity** by ensuring that DNS responses come from the legitimate source
  - Resolvers verify the digital signature using the public key.
  - If the signature is valid, the resolver can trust that the data originates from the authoritative source.
- Ensures **Data Integrity** by detecting tampering or corruption of DNS records
  - Any modification to the DNS records invalidates the digital signature.
  - Resolvers reject responses with invalid signatures, preventing the use of altered data.
- Provides **Nonexistence Proof** for domains or records that do not exist
  - Uses a special record type called **NSEC** or **NSEC3**.
  - These records prove the nonexistence of a queried domain or record by listing the next existing domain in the zone.
  - The NSEC/NSEC3 records are also signed, ensuring their authenticity and integrity.
- Relies on a chain of trust starting from the root zone
  - The root zone's public key is distributed to resolvers as a trust anchor.
  - Each zone's public key is signed by its parent zone, creating a hierarchical chain of trust.
  - Resolvers validate signatures by traversing this chain up to the root.