DNSsec

Josh Wilcox (jw14g24)

April 18, 2025

Table of Contents

Overview of DNSSEC

- DNSSEC (Domain Name System Security Extensions) is a suite of extensions that add **security** to the DNS protocol
- It addresses critical vulnerabilities in the original DNS system:
  - DNS was designed in the 1980s without built-in security measures
  - Originally relied on trust without verification mechanisms
- DNSSEC protects against DNS-specific attacks:
  - **Cache poisoning** - when attackers inject false information into DNS caches
  - **Man-in-the-middle attacks** - intercepting DNS queries and providing fake responses
- DNSSEC does **NOT** provide:
  - Encryption of DNS data (it remains readable)
  - Protection against DDoS attacks
  - Confidentiality of DNS queries

Why DNSSEC is Needed

- The DNS protocol translates human-readable domain names (e.g., example.com) to IP addresses
- Without DNSSEC, this process is vulnerable to several attacks:
  - **DNS spoofing** - an attacker responds faster than legitimate DNS servers
  - **DNS cache poisoning** - falsified records are stored in DNS resolvers
    - This allows attackers to redirect users to malicious websites
    - Can persist for as long as the cache entry's Time-To-Live (TTL)
  - **DNS hijacking** - modifying DNS settings at various points
- Real-world consequences:
  - Users directed to phishing sites instead of legitimate banking websites
  - Email redirection for credential theft or business email compromise
  - Compromising encrypted connections (HTTPS) by directing to attacker-controlled sites

How DNSSEC Works

- DNSSEC uses **digital signatures** to verify the authenticity of DNS data
- It implements a **chain of trust** from the DNS root zone down to individual domain names
- Key components:
  - **Public-key cryptography** - uses asymmetric key pairs
    - Private keys sign DNS records
    - Public keys verify the signatures
  - **Digital signatures** - attached to DNS records
    - Created using the zone's private key
    - Can be verified using the zone's public key
    - Any modification to the record will invalidate the signature
- DNSSEC adds new DNS record types:
  - **DNSKEY** - contains the public key used for verification
  - **RRSIG** - contains the digital signature for a record set
  - **DS** - links a child zone to its parent zone (creates the chain of trust)
  - **NSEC/NSEC3** - proves the non-existence of records

Chain of Trust

- DNSSEC establishes a hierarchical **chain of trust** from the DNS root down
- How the chain works:
  - The DNS root zone is the trusted starting point (trust anchor)
  - The root zone's public key is widely distributed and trusted
  - Each parent zone authenticates its child zones using DS (Delegation Signer) records
  - This creates an unbroken chain from the root to any signed domain
- Verification process:
  - DNS resolver starts with the trusted root key
  - Validates signatures at each level of the domain hierarchy
  - Each successful verification allows trusting the next level
  - Any broken link in the chain causes validation failure
- Practical example:
  - To verify example.com, the resolver:
  - Validates the root (.) zone's signature
  - Uses the root to validate the .com zone
  - Uses the .com zone to validate example.com

Key Types

- DNSSEC uses two types of key pairs for each zone:
- **Key Signing Key (KSK)**:
  - The more secure, rarely changed key
  - Used only to sign the Zone Signing Key
  - Published in the parent zone (as DS records)
  - Functions as the ”anchor of trust” for the zone
  - Typically uses stronger cryptography and longer key length
  - Changing the KSK requires coordination with the parent zone
- **Zone Signing Key (ZSK)**:
  - Used to sign all the actual records in the zone
  - Changed more frequently (key rotation)
  - Only referenced within the zone itself
  - Typically uses shorter key length for better performance
  - Can be rolled over without involving the parent zone
- This separation provides:
  - Better security (compromise of ZSK doesn’t compromise entire chain)
  - Operational flexibility (easier key rotation)
  - Performance benefits (smaller signatures for routine operations)

DNSSEC Record Types

- **DNSKEY Record**:
  - Contains the public key for a zone
  - Used to verify RRSIG records
  - Can be either KSK or ZSK (flagged accordingly)
  - Multiple DNSKEY records can exist for key rotation
- **RRSIG (Resource Record Signature)**:
  - Contains the digital signature for a set of DNS records
  - Includes information on: which key was used, signature validity period, the algorithm used
  - Allows verification that records haven't been tampered with
- **DS (Delegation Signer)**:
  - Published in the parent zone
  - Contains a hash of a child zone's KSK
  - Creates the "chain of trust" between parent and child
  - Example: .com zone contains DS records for example.com
- **NSEC/NSEC3**:
  - Provide authenticated denial of existence
  - Prove that a requested DNS record really doesn't exist
  - NSEC lists the next secure record in the zone
  - NSEC3 uses hashed names to prevent zone enumeration

DNSSEC Validation Process

- When a DNSSEC-aware resolver receives a response:
  1. The resolver checks if the response is signed (contains RRSIG)
  2. It retrieves the DNSKEY record for the zone
  3. It validates the DNSKEY using the DS record from the parent zone
  4. It uses the validated DNSKEY to verify the RRSIG
  5. If signature verification succeeds, the data is considered authentic
- Three possible validation states:
  - **Secure**: Full DNSSEC validation succeeded
  - **Insecure**: Domain not signed with DNSSEC (intentionally unsigned)
  - **Bogus**: Validation failed (potential attack or misconfiguration)
- When validation fails:
  - DNSSEC-validating resolvers will refuse to return the answer
  - They return a SERVFAIL error instead
  - This prevents users from receiving potentially malicious data

Challenges and Limitations

- **Complexity**:
  - DNSSEC adds significant complexity to DNS administration
  - Requires careful key management and regular key rotation
  - Misconfiguration can cause domains to become unreachable
- **Increased DNS response size**:
  - DNSSEC responses are significantly larger than regular DNS
  - Can cause issues with UDP packet fragmentation
  - May require fallback to TCP, increasing latency
  - Can be exploited for DNS amplification attacks
- **Key management challenges**:
  - Key compromise requires emergency key rollover
  - Key rollovers must be carefully scheduled and executed
  - Failure to update keys can break the chain of trust
- **Limitations of DNSSEC**:
  - Does NOT encrypt DNS queries or responses
  - Does NOT prevent eavesdropping on DNS traffic
  - Does NOT protect against DDoS attacks
  - Requires both authoritative servers AND resolvers to support it

DNSSEC vs. DNS over HTTPS/TLS

- DNSSEC and DoH/DoT solve different problems:
- **DNSSEC**:
  - Provides **data integrity** and **origin authentication**
  - Ensures DNS data hasn't been tampered with
  - Works throughout the DNS hierarchy
  - Does NOT provide privacy or encryption
- **DNS over HTTPS (DoH) / DNS over TLS (DoT)**:
  - Provide **transport security** and **privacy**
  - Encrypt DNS traffic between client and resolver
  - Prevent eavesdropping and some man-in-the-middle attacks
  - Do NOT validate the authenticity of DNS data itself
- **Combined approach**:
  - Both technologies can and should be used together
  - DNSSEC ensures authentic data
  - DoH/DoT ensures private, encrypted transport
  - Together they address most DNS security issues

Practical Implementation

- **Enabling DNSSEC validation** (as a user):
  - Most modern DNS resolvers support DNSSEC validation
  - Public resolvers like Cloudflare (1.1.1.1) and Google (8.8.8.8) validate DNSSEC
  - ISP resolvers may or may not validate DNSSEC
  - Local resolvers can be configured to validate DNSSEC
- **Implementing DNSSEC** (for domain owners):
  - Generate KSK and ZSK key pairs
  - Sign all records in the zone
  - Publish DNSKEY records in the zone
  - Coordinate with parent zone to publish DS record
  - Implement regular key rotation procedures
  - Set up monitoring for signature expiration
- **Testing DNSSEC**:
  - Online tools like DNSViz and DNSSEC Analyzer
  - Command-line tools: dig +dnssec, delv
  - Browser extensions to verify DNSSEC status