DNSSEC

DNS SECURITY EXTENSIONS

INTRODUCTION TO DNSSEC FOR SECURING DNS QUERIES AND INFORMATION

Peter R. Egli
INDIGOO.COM
DNS Security Extensions (DNSSEC)

Contents
1. Security Problems of DNS
2. Solutions for securing DNS
3. Security with DNSSEC
4. How to find the signer's public DNS key
5. DNS lookup: Recursive name servers
6. DNSSEC deployment
1. Security problems of DNS (1/3)
DNS is vulnerable to a number of attacks. The attacks are usually targeted at misdirecting a client to a malicious server under the control of the attacker.

Some attacks are now difficult to conduct since DNS servers like BIND fixed the software. Still, DNS is too important thus warranting the need for cryptographic protection against poisoning and spoofing.
1. Security problems of DNS (2/3)

DNS Cache Poisoning:
Inject false information into a DNS server's or a client's DNS cache.

DNS spoofing:
In a DNS spoofing attack, an attacker pretends to be a real DNS server and sends a spoofed DNS answer back to the requestor (either a DNS server or a DNS client). The spoofed answer contains a false mapping of the requested name to IP address.

DNS ID hacking:
Spoofing requires to match the DNS answer ID with the DNS request ID (DNS transaction ID). Thus the attacker must either intercept the DNS request message or "estimate" the DNS transaction ID in the spoofed response (e.g. in Windows XP the DNS ID is monotonically increasing!).
DNS Security Extensions (DNSSEC)

1. Security problems of DNS (3/3)

Unrelated data attack:
In an unrelated data attack, the attacker sends back an answer to the query question, but adds additional and unrelated IP→name mappings to misdirect the client to a malicious site.

Related data attack:
A related data attack is similar to an unrelated attack with the difference that the attacker sends extra information related to the query question such as spoofed MX, NS or CNAME records.

Example question:
www.indigoo.com

Spoofed response with unrelated data:
www.indigoo.com. 1D IN NS www.attacker.com
www.attacker.com. 1D IN A 1.2.3.4
## 2. Solutions for securing DNS

TSIG and SIG(0) are older solutions for securing DNS. DNSSEC is a comprehensive approach for securing DNS (secure DNS database as well as DNS queries).

It is expected that DNSSEC will become the standard solution for securing DNS.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>RFC</th>
<th>Protection of transfer of DNS record</th>
<th>Protection of DNS database</th>
<th>Built-in PKI (automated key distribution)</th>
<th>Hash algorithms</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSIG</td>
<td>RFC2845</td>
<td>Yes</td>
<td>No</td>
<td>No, but TKEY (RFC2930) adds Diffie-Helman to TSIG</td>
<td>RSAMD5 RSASHA1 RSASHA256</td>
<td>Based on MAC (Message Authentication Code). Used for protecting zone transfers (widely used). No levels of authority (every host with the secret key may update a DNS record).</td>
</tr>
<tr>
<td>SIG(0)</td>
<td>RFC2931</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>RSAMD5 RSASHA1 RSASHA256</td>
<td>Not widely used. Uses digital signature instead of MAC.</td>
</tr>
<tr>
<td>DNSSEC</td>
<td>RFC4033 RFC4034 RFC4035</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>RSAMD5 DH ECC RSASHA1</td>
<td>Used to secure DNS transactions as well as the DNS database.</td>
</tr>
</tbody>
</table>
DNS Security Extensions (DNSSEC)

3. Security with DNSSEC
DNSSEC adds a digital signature to the answer records.

Answer records

RRSIG

Answer records

(A, AAAA, NS, MX etc.)

RRSIG (Resource Record Signature) that verifies the answer records

DNS answer packet

DNS server signing process:

Answer records

RRSIG

Hash function

Hash 101011...

Encryption function

Signature 110010...

Signer's private key

Additional info (expiry date etc.)

DNS resolver verification:

Answer records

RRSIG

Hash function

Hash 101011...

Decryption function

Hash 101011...

Signer's public key

? =
4. How to find the signer's public DNS key

An authentication chain leads from root to leaf-domain. Each level contains DS records that reference / point to DNSKEY records in a subdomain.

N.B.: The parent domain is authoritative for its subdomains.

The starting point of the chain of trust is an anchor point (known good public key = trust anchor).

- Resolver's "built-in" public key of domain root
- Verifies
- Domain root
- Designated Signer (or Delegation Signer) Record
- References and verifies a DNSKEY record in the delegated zone (verification through hashing function).
- DNSKEY record
- DS record
  - Verifies
  - Subdomain, e.g. `.ch`
  - Designated Signer Record
- DNSKEY record
  - Verifies
  - Subdomain, e.g. `hsz-t.ch`
  - Designated Signer Record
DNS Security Extensions (DNSSEC)

5. DNS lookup: Recursive name servers
Standard setup where only name servers (e.g. ISP name servers) are security-aware. DNS clients are not security aware (simpler deployment, no trust anchors required in client).

DNS Client → ISP NS → Security aware resolver → hsz-t.ch NS → .ch NS → Root NS

- DNS query for www.hsz-t.com
- Security aware resolver
- Request .ch-DS record, DO=1
  - .ch-DS record + .ch-NS record
  - Verify .ch-DS record with "built-in" public root key
    - Request hsz-t.ch-DS + hsz-t.ch-DNSKEY
      - hsz-t.ch-DS + hsz-t.ch-DNSKEY + hsz-t.ch-NS
      - Verify hsz-t.ch-DNSKEY with .ch-DS, verify hsz-t.ch-DS with hsz-t.ch-DNSKEY
        - Request www.hsz-t.ch
          - www.hsz-t.ch answer with RRSIG record
            - Verify RRSIG record with hsz-t.ch-DNSKEY, verify answer with RRSIG record

DNS answer: www.hsz-t.ch = 193.5.54.123
6. DNSSEC deployment
As of 2013 DNSSEC is not (yet) widely deployed, but since the root zone has been signed in 2010, DNSSEC deployment is gradually picking up speed. Deployment see https://labs.ripe.net/Members/wnagele/dnssec-deployment-today

Problems that prevent fast deployment:
1. "Bootstrap" problem:
DNSSEC requires a certain level of deployment to deliver an increase in security. But as long this level is not reached, people will not see a benefit and thus not deploy DNSSEC ("chicken and egg problem").

2. Complexity:
DNSSEC is (as many security protocols) rather complex (defined in multiple RFCs).

3. DNS server load:
DNSSEC puts additional load on DNS servers (hashing, encryption / decryption).