# DNSSEC
# Sichere Namensauflösung im Internet

Marcus Obst*
marcus.obst@etit.tu-chemnitz.de
http://www.tu-chemnitz.de/~maob/nt

Heiko Schlittermann†
hs@schlittermann.de
http://www.schlittermannn.de

Chemnitzer Linux-Tage 2010
http://chemnitzer.linux-tage.de/2010/

## Zusammenfassung

Die Auflösung von Hostnamen in IP-Adressen mit Hilfe des Domain Name System (DNS) ist fundamentaler Bestandteil des Internets, so wie wir es heute kennen. Dass die Entwicklung von DNS keinesfall stillsteht, sondern rasant voranschreitet, soll nachfolgend am Beispiel der Protokollerweiterung *DNS Security Extension* (DNSSEC) gezeigt werden. DNSSEC ist eine umfassende Möglichkeit zur Sicherstellung der Authentizität und Integrität von DNS-Abfragen, die in Zukunft wohl immer mehr an Bedeutung gewinnen wird.

## 1 Einleitung

Das *Domain Name System* (DNS), ein hierarchisches dezentralisiertes System, ist der zentrale Dreh- und Angelpunkt im Internet wenn es um die Umwandlung von Hostnamen in IP-Adressen geht. Neben der einfachen Vorwärtsauflösung spielt DNS eine wichtige Rolle im Zusammenhang mit der Zustellung von E-Mail. Aufgrund der fundamentalen Bedeutung des DNS steht es u.a. immer wieder im Fokus der verschiedenen Angriffsszenarien. Neben illegalen Manipulationen wie z.B. DNS-Spoofing oder Cache Poisoning (siehe [2]), sind aber auch „behördlich legitimierte“ Eingriffe wie z.B. das „Stoppschild“ oder transparente DNS-Proxies technisch möglich. Dass an dieser Stelle Handlungsbedarf besteht, hat die *Internet Engineering Task Force* (IETF) erkannt und einen neuen Standard, die *Domain Name System Security Extension* (DNSSEC), erarbeitet. Im Jahr 2005 wurde erstmals eine Top Level Domain (`.se`), heute sind es auch einige andere (`.cz`, `.bg`, ...) mit einer digitalen Signatur auf Basis von DNSSEC versehen.

Dieser Beitrag möchte einen Einblick in die Grundlagen und Funktionsweise von DNSSEC geben. Nach einer kurzen Vorstellung des aktuellen Stands der Technik und den damit verbundenen Unzulänglichkeiten des DNS folgt eine knappe Einführung in die Public Key Kryptographie. Anschließend werden kurz die notwendigen Komponenten zum Aufsetzen einer signierten Zone erläutert. Das Verhalten des Resolvers bzw. eines DNS-Servers, der die DNS Security Extension benutzt, soll exemplarisch mit einer

*Professur für Nachrichtentechnik, Technische Universität Chemnitz
†schlittermann – internet & unix support, Dresden

„echten“ signierten Zone im Internet demonstriert werden. Serverseitig werden die DNS-Implementierung des ISC Bind9 und die damit verbundenen Werkzeuge vorgestellt.

Da DNSSEC einen ganzheitlichen Ansatz zur Sicherung der Authentizität und Integrität von DNS-Transaktionen im Internet darstellt, ist eine schrittweise Einführung nicht ohne weiteres möglich. Wie man dennoch DNS-Antworten validieren kann, soll am Beispiel von dig und dem Bind9-Resolver gezeigt werden.

## 2 Stand der Technik

Beim Entwurf des DNS spielte die Authentizität der übermittelten Daten ebensowenig eine primäre Rolle wie eine robuste Authentifizierung des Absenders.

DNS-Server erhalten Anfragen und beantworten diese. Um die Quelle der Anfrage machen sie sich in der Regel keine Gedanken, eine sehr schwache Prüfung der Authentizität der Daten und der Identität des Absenders auf Basis der Quell-IP ist möglich.

DNS-Clients (Resolver, Forwarder, ...) stellen Anfragen und vertrauen den Antworten uneingeschränkt. Eine sehr schwache Überprüfung der Authentizität und Quelle der Antworten mit Hilfe von Query-IDs und der Portzuordnung ist möglich.

Transaction Signatures (TSIG) können verwendet werden, um sowohl die Unversehrtheit der Daten als auch die Authentizität des Absenders zu prüfen. Dazu wird ein symmetrisches Verschlüsselungsverfahren verwendet. Alle Beteiligten müssen den verwendeten Schlüssel kennen. Der Sender ergänzt die Antwort um die verschlüsselte Prüfsumme des Datensatzes. Der Empfänger kann die entschlüsselte Prüfsumme mit der selbst errechneten Prüfsumme vergleichen. Praktikabel ist TSIG nur bei einer überschaubaren Anzahl von Kommunikationspartnern. Typischerweise wird TSIG für Zonentransfers und dynamische Updates verwendet. Eine Verschlüsselung der Daten finden nicht statt. Angriffe durch die Wiederholung von „eingefangenen“ Daten werden durch Zeitstempel verhindert.

TSIG steht in der Regel nicht für Stub-Resolver zur Verfügung.[1]

## 3 Grundlagen

Wenngleich TSIG fast alle Forderungen erfüllen würde, die Verwendung eines gemeinsamen Schlüssels verhindert eine Skalierung ebenso wie die sichere Kommunikation von bis dato gegenseitig unbekannten Kommunikationspartnern.

DNSSEC stellt nun einen möglichen Ausweg aus diesem Dilemma dar. Statt auf einem symmetrischen Verfahren aufzusetzen, sieht DNSSEC die Verwendung von asymmetrischer Kryptographie vor. Damit wird es dem Client möglich, auch von bislang unbekannten Zonen die erhaltene Daten auf Authentizität zu prüfen. Dazu nutzt er die (kryptographische) Signatur der erhaltenen Antwort.

[1] Jedenfalls haben wir keine Hinweise darauf gefunden, daß der Stub-Resolver der libc6 das könnte.

```
$ORIGIN xxx.schlittermann.de.
$TTL 1d
@   IN SOA pu.schlittermann.de. hostmaster.schlittermann.de.
    (
                                    ...
    )

    IN NS         pu.schlittermann.de.
    IN TXT        "example zone for DNSSEC"
    IN MX         10 ssl.schlittermann.de.
a   IN A          194.39.236.1
b   IN A          194.39.236.2
```

Listing 1: Beispielhafte DNS-Zone ohne DNSSEC-Absicherung

```
    IN DNSKEY 257 3 5 AwEAAbHQS/v8ACLTAP2Un1QboGbhVftYZC...
```

Listing 2: Einfügen des öffentlichen DNSSEC-Schlüssels zum Absichern der Zone (Der Base64-kodierte Schlüssel ist aus Platzgründen nur unvollständig angegeben).

Beim Signieren wird eine Prüfsumme (Hash-Wert) über die zu schützenden DNS-Einträge berechnet und anschließend mit einem privaten Schlüssel verschlüsselt. Der Client kann mit dem zugehörigen und bekannten öffentlichen Schlüssel diese Signatur entschlüsseln und den erhaltenen Hash-Wert mit dem selbst ermittelten Hash-Wert aus der Antwort vergleichen.

Privater und öffentlicher Schlüssel sind je abzusichernder Zone einzigartig. Die Vertrauenswürdigkeit der öffentlichen Schlüssel wird über Vertrauensketten (Chain of Trust) oder auf alternativen Wegen sichergestellt.

Ähnliche asymmetrische Verfahren finden man bei (kryptographischen) Signaturen von E-Mails mit GnuPG oder PGP oder bei der schlüsselbasierten Authentifizierung der SSH.

DNSSEC hat nicht die Verschlüsselung der Daten zum Ziel, sondern einzig und allein die Prüfbarkeit und damit die Vertrauenswürdigkeit der Daten. Es bleibt dem Client (Resolver) überlassen, ob er diese Möglichkeiten nutzt.

## 4 DNSSEC auf Serverseite

Bisher wurden in einer Zonen-Datei ausschließlich die reinen Nutzdaten (z.B. IP-Adressen, Hostnamen, Mail-Exchanger) einer Domain gespeichert (siehe Listing 1).

Damit ein DNSSEC-fähiger Client (das kann wiederum ein anderer DNS-Server (als Forwarder oder Resolver) oder aber ein entsprechender Stub-Resolver sein) in der Lage ist, die erhaltenen Daten zu prüfen, benötigt er zusätzlich die Signaturen der Einträge, sowie den öffentlichen Teil des Schlüssels, mit dem die Signaturen erzeugt wurden.

Zur Verbreitung des öffentlichen Schlüssel einer Zone existiert ein neuer DNS-Record, der *DNSKEY*. Ein DNSKEY-Record enthält die Zone, für die er gelten soll, sowie den

```
a    86400 IN A   194.39.236.1
     86400 RRSIG A 5 4 86400 20100225192903 (
                  20100126192903 50433 xxx.schlittermann.de.
                  aVZnKDppT+ane7zrZmv0Z2HpyaD3Q4LKvi5E
                  pqfA6ZmiVP+tpfxfdWEazZ8w5RFPX4LpfmUD
                  ... )
```

Listing 4: Signatur im RRSIG-Record für die IP-Adresse von Host *a*

öffentlichen Schlüssel in Base64-Kodierung.[2] Listing 2 zeigt den prinzipiellen Aufbau eines DNSKEY-Eintrags für die Zone *xxx.schlittermann.de*.

Neben dem DNSKEY-Record muss außerdem die eigentliche Signatur eines DNS-Eintrages, die dann später durch den Client geprüfte werden kann, zugänglich gemacht werden. Dies geschieht durch den ebenfalls neu hinzugekommen *RRSIG*-Record. Ein RRSIG-Eintrag enthält aus Effizienzgründen immer die Signatur zu einer Gruppe (RRset) von gleichen Einträgen. Man kann sich also zum Beispiel vorstellen, dass eine DNS-Abfrage, nach der IP-Adresse von einem bestimmten Host zwei Antworten liefert. Da die Antworten beide zum gleichen Host gehören und vom gleichen Typ sind, muss nur ein RRSIG-Record erzeugt und ausgeliefert werden. In Listing 4 ist die Signatur für den Address-Record von Host *a.xxx.schlittermann.de* zu sehen. Zusätzlich zur digitalen Signatur sind noch die Gültigkeitszeit, sowie Informationen zum zugehörigen DNSKEY-Eintrag hinterlegt. Die Signierung von negativen Antworten (beispielsweise, wenn der DNS-Server keine IP-Adresse zu einem Hostname liefern kann) ist durch einen weiteren DNS-Eintrag, den NSEC-Record, realisiert (siehe hierzu [1]).

Der Vollständigkeit halber sei noch erwähnt, dass die DNS-Server-Implementierung des Bind die neu eingeführten DNSSEC-Records DNSKEY und RRSIG erst kennt, nachdem die DNSSEC-Unterstützung in der Nameserver-Konfiguration eingeschaltet ist:

```
options {
                  ...
                  dnssec-enable yes;
};
```

Listing 3: Einschalten von DNSSEC im Bind Nameserver

## 5 Resolver-Fähigkeiten

Mit den neu hinzugekommen DNSSEC-Einträgen DNSKEY und RRSIG für eine Zone bzw. einen einzelnen DNS-Record ist ein Client nun prinzipiell in der Lage, die Antwort auf seine Anfrage zu verifizieren. Nachfolgend soll gezeigt werden, welche Schritte notwendig sind, um die IP-Adresse für den Host *a.xxx.schlittermann.de* herauszufinden und die Antwort gleichzeitig mit Hilfe von DNSSEC zu verifizieren:

1. Anfrage nach dem Address-Record von a.xxx.schlittermann.de an den zuständigen Nameserver. Dabei muss das *DNSSEC OK* Flag (DO) gesetzt sein, um ebenfalls DNSSEC-Einträge zu erhalten.

[2] Es sind noch weitere Informationen wie z.B. der Typ des Verschlüsselungsalgorithmus enthalten.

2. Der DNSSEC-fähige DNS-Server liefert nun wie gewünscht den ursprünglich angefragten A-Record, sowie die dazugehörige Signatur in Form eines RRSIG-Eintrages.
3. Bevor die Validierung des Address-Records durchgeführt werden kann, muss der Resolver noch den im RRSIG-Record angegebenen öffentlichen Schlüssel, mit dem die Signatur erstellt wurde, beim DNS-Server anfragen.
4. Da jetzt alle Informationen verfügbar sind, kann die Signatur des Address-Records von *a.xxx.schlittermann.de* überprüft werden.

Mit Hilfe des Kommandos *dig* lassen sich die einzelnen Schritte nachvollziehen. Zuerst wird der Address-Record sowie die dazugehörige Signatur abgefragt:

```
# dig +dnssec +norec a.xxx.schlittermann.de. @pu.schlittermann.de
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57711
;; flags: qr aa ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;a.xxx.schlittermann.de.                IN      A

;; ANSWER SECTION:
a.xxx.schlittermann.de. 86400   IN      A       194.39.236.1
a.xxx.schlittermann.de. 86400   IN      RRSIG   A 5 4 86400
        20100225192903 20100126192903 50433 xxx.schlittermann.de.
        aVZnKDppT+ane7zrZmv0Z2HpyaD3Q4LKvi5Ehk/6Z1krMaDbhMt...
...
```

Listing 5: DNS-Query mit dig (und +dnssec flag)

Die Angabe des Parameters *+dnssec* weist dig an, zusätzliche DNSSEC-Einträge abzufragen. Die Option *+norec* in Verbindung mit der Angabe des DNS-Servers veranlasst, dass eine nicht-rekursive Anfrage gestellt wird.

Im RRSIG-Eintrag ist hinterlegt[3], mit welchem öffentlichen Schlüssel der Address-Record signiert wurde. Als nächstes muss also der passende DNSKEY-Eintrag angefragt werden:

```
# dig @pu.schlittermann.de xxx.schlittermann.de DNSKEY
;; Truncated, retrying in TCP mode.

; <<>> DiG 9.3.4-P1.2 <<>> @pu.schlittermann.de xxx.schlittermann.de DNSKEY
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18169
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;xxx.schlittermann.de.          IN      DNSKEY

;; ANSWER SECTION:
xxx.schlittermann.de.   86400   IN      DNSKEY  257 3 5
  AwEAAbHQS/v8ACLTAP2Un1QboGbhVftYZChUIJt5O2bzyGdbHShfNILT
  qbZRXLMz4f67TA2y4pGIJ8fFyTzi13GXn+V67mmjauZnXTkSwpe90w2N
  ...
```

Listing 6: DNSKEY-Query mit dig

[3] Durch Angabe der Domain sowie des Fingerabdrucks, ist der DNSKEY-Eintrag eindeutig bestimmt.

Ab diesem Zeitpunkt wäre der Client bzw. Resolver in der Lage, die Antwort des DNS-Servers zu verifizieren. Eine fundamentale Frage, die sich an dieser Stelle allerdings stellt, ist: Woher weiß der Client eigentlich, dass der zurückgelieferte DNSKEY-Eintrag nicht modifiziert wurde?[4]

DNSSEC sieht an dieser Stelle die sogenannte *Chain of Trust* vor, d.h. normalerweise müsste der Client nun zusätzlich noch die darüber liegende Instanz (also den zuständigen Nameserver für die Domain *schlittermann.de*) zur Authentizität des gerade ermittelten DNSKEY-Eintrages befragen.[5]

## 6 Insellösung

Verlässt man sich auf die Chain of Trust, ist eine komplette DNSSEC-Infrastruktur von der Toplevel-Domain bis hin zum Root-Nameserver notwendig. Aktuell ist diese Voraussetzung nicht gegeben, also kann man dem Resolver alternativ eine Liste mit vertrauenswürdigen DNSKEY-Einträge konfigurieren. Im Falle von dig würde man den geprüften(!) DNSKEY-Eintrag von *xxx.schlittermann.de* in der Datei *trusted-key.key* hinterlegen. Anschließend wäre dig in der Lage, die Antwort vom DNS-Server selbständig zu verifizieren:

```
# dig +dnssec +sigchase a.xxx.schlittermann.de
...
;; WE HAVE MATERIAL, WE NOW DO VALIDATION
;; VERIFYING A RRset for a.xxx.schlittermann.de. with DNSKEY:50433: success
;; OK We found DNSKEY (or more) to validate the RRset
;; Ok, find a Trusted Key in the DNSKEY RRset: 30922
;; VERIFYING DNSKEY RRset for xxx.schlittermann.de. with DNSKEY:30922: success

;; Ok this DNSKEY is a Trusted Key, DNSSEC validation is ok: SUCCESS
```

Listing 7: Verifikation mit dig

Zuerst versucht dig, die Chain of Trust durch Anfragen bei der darüberliegenden Domain aufzubauen, wird dort aber nicht fündig. Da der öffentliche Schlüssel von *xxx.schlittermann.de* in der lokalen Datenbank hinterlegt ist, verifiziert dig letztendlich gegen diesen und liefert eine positive Antwort.

Auch der Bind-Nameserver lässt sich mit einer statisch gepflegten Liste von vertrauenswürdigen DNSKEY-Einträgen konfigurieren und kann damit validieren.

In der Bind Version 9.3.x ist es ausreichend, DNSSEC mit der Option aus Listing 3 einzuschalten. Ab Bind Version 9.4 ist für die DNSSEC-Validierung die Zeile `dnssec-validation yes;` im `options`-Abschnitt hinzuzufügen. Natürlich muss dem DNS-Server auch noch die Liste mit vertrauenswürdigen Schlüsseln mitgeteilt werden. Für *xxx.schlittermann.de* sähe das z.B. wie folgt aus:

[4] Möglich wäre ein scheinbar transparenter DNS-Proxy, der die Daten mit seinem privaten Schlüssel signiert.

[5] Dieser Vorgang setzt sich theoretisch über die Toplevel-Domain bis zum Root-Nameserver fort. Die DNSKEY-Einträge der Root-Server sind dann fest in die Client-Resolver eingebaut.

```
trusted-keys {
 "xxx.schlittermann.de." 257 3 5 "AwEAAbHQS/v8AC...";
};
```

Listing 8: Konfiguration des Bind-Nameservers mit einer Liste von vertrauenswürdigen Schlüsseln für die DNSSEC-Validierung

Erfolgt nun eine Anfrage an diesen Server, so markiert er die Antwort mit dem *AUTHENTICATED DATA* Flag (AD). Damit wird dem Client (Stub-Resolver), der nicht selbst prüfen kann oder will, mitgeteilt, daß auf dem angefragten Resolver die Prüfung erfolgreich war. Natürlich muß der Stub-Resolver das jetzt einfach glauben:

```
# dig +dnssec a.xxx.schlittermann.de
...
; <<>> DiG 9.5.1-P3 <<>> +dnssec @localhost a.xxx.schlittermann.de
; (2 servers found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10318
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;a.xxx.schlittermann.de.                IN      A

;; ANSWER SECTION:
a.xxx.schlittermann.de. 86400   IN      A       194.39.236.1
a.xxx.schlittermann.de. 86400   IN      RRSIG   A 5 4 86400 201...
```

Listing 9: Verifikation durch den Resolver

## 7 Fazit

DNSSEC stellt eine Möglichkeit zur Absicherung des Namensauflösung im Internet bereit. Der anfängliche Aufwand zum Aufsetzen einer eigenen mit DNSSEC abgesicherten Zone ist nicht zu vernachlässigen. Kann oder will man aus verschiedenen Gründen nicht auf eine Chain of Trust-Infrastruktur zurückgreifen bietet sich die Möglichkeit dem Resolver eine statische Liste mit vertrauenswürdigen DNSKEY-Einträgen einzustellen. Eine weitere Methode ist die Nutzung der sogenannten *Domain Lookaside Validation*, die aber hier nicht weiter betrachtet wurde (aber im Workshop als auch im Vortrag).

## Literatur


[1] P. Albitz and C. Liu. *DNS and Bind*. O'Reilly, 2006.

[2] A. Klein. BIND 9 DNS cache poisoning, 2007.