DNS Security Extensions (DNSSEC)

Seminar Innovative Internettechnologien und Mobilkommunikation WS2008/2009

Ralf Glauberman
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: glauberm@in.tum.de
Betreuer: Andreas Müller

***Kurzfassung*— In dieser Arbeit geht es um die DNS Sicherheitserweiterungen *DNSSEC*. Dabei werden der Grund für die Einführung von DNSSEC, die sich durch DNSSEC ergebenden Änderungen am Aufbau des DNS, neue Möglichkeiten, aber auch Probleme bei der Einführung von DNSSEC aufgezeigt. Ebenfalls wird auf die historische Entwicklung von DNSSEC und den bisherigen Stand der Verbreitung eingegangen.**

***Schlüsselworte*— Domain Name System Security Extensions DNS DNSSEC**

I. Einleitung

Alle Nutzer des Internets sind es heute gewohnt, Namen für Server zu verwenden. Die Übersetzung der Namen in für Computer nutzbare IP-Adressen passiert dabei in der Regel automatisch und ist für die meisten so selbstverständlich, dass sich kaum noch jemand Gedanken macht, wie dies eigentlich funktioniert. Seit langer Zeit dient das *Domain Name System (DNS)* der Auflösung von Namen in IP-Adressen und umgekehrt. Die Zeiten, in denen Host-Dateien für diese Auflösung manuell gepflegt werden mussten, sind längst vergessen. Obwohl eine funktionierende Namensauflösung für die Funktion des gesamten Internets unabdingbar ist, macht sich kaum jemand Gedanken, wie diese eigentlich abläuft. Um so bedrohlicher wirken die in letzter Zeit zunehmenden Berichte über Verwundbarkeiten im DNS-System (vgl. [**?**]). Zwar sind Angriffe auf das DNS nichts Neues (siehe [**?**]), jedoch scheinen die Bedrohungen durch Manipulationen heute, in einer Zeit da Phishing-Angriffe an der Tagesordnung sind und eine funktionierende Namensauflösung für viele Personen und auch große Firmen unabdingbar ist, wesentlich gefährlicher als noch vor einigen Jahren zu sein. Den meisten ist dabei nicht bekannt, dass seit längerer Zeit an einer Lösung gearbeitet wird, die Manipulationen am DNS effektiv ausschließen soll: die *DNS Sicherheitserweiterungen DNSSEC*. Dabei ist es durchaus sinnvoll, sich mit diesem neuen Standard einmal genauer zu beschäftigen und aufzuzeigen, welche Probleme dadurch genau behoben werden können.

II. Aufbau des DNS

Die Informationen des DNS sind hierarchisch in einer verteilten, dezentralen Datenbank gespeichert. Als einer der ältesten Dienste des Internets - die Anfänge des DNS gehen bis 1983 zurück - spielte Sicherheit bei der Konzeption keine Rolle. Alle Informationen sind öffentlich und weder Anfragen noch Antworten sind vor Manipulationen geschützt. Logisch ist das DNS in Zonen aufgeteilt, welche als Container für andere Zonen und Einträge, so genannte *Resource Records (RRs)* dienen. Jede Zone hat einen eindeutigen Besitzer, untergeordnete Zonen können einen anderen Besitzer haben. Resource Records (RRs) haben einen Namen, einen Typ, einen Inhalt (Wert) und Attribute. Verschiedene Typen sind bekannt, z.B. A für IPv4-Adressen, AAAA für IPv6-Adressen, CNAME für Alias-Namen oder PTR für die Rückwärtsauflösung von IP-Adressen in Namen. Das Format des Wertes ist abhängig vom Typ.

III. Sicherheitsprobleme beim bisherigen DNS

Zu der Zeit, als das DNS entwickelt wurde, spielte Sicherheit bei Computersystemen noch keine große Rolle. Daher wurde DNS nicht als sicheres System konzipiert, was eine Reihe von Angriffsmöglichkeiten eröffnet. Vorweg sei gesagt, dass DNSSEC nicht alle diese Angriffe verhindern kann.

A. Angriffe auf das DNS

Eine der wichtigsten Angriffsmöglichkeiten ist die gezielte Manipulation des Inhalts von Anfragen oder Antworten. Dadurch ist es möglich, für einen angefragten RR einen anderen Inhalt zurückzuliefern, die Existenz eines RR zu verleugnen oder die Existenz eines in Wirklichkeit nicht existierenden RRs vorzuspiegeln. Des weiteren besteht eine Gefahr durch *Denial of Service (DoS)* Angriffe auf DNS-Server. Auch können Informationen über den Inhalt einer Zone unbeabsichtigt offengelegt werden, also z.B. eine Auflistung aller RRs und Unterzonen einer Zone ermöglicht werden. Dies ist nach der ursprünglichen Ansicht, nach der alle Informationen im DNS grundsätzlich öffentlich sind, keine Bedrohung, allerdings aus verschiedenen Gründen, die im Kapitel **??** erläutert werden, heute oftmals unerwünscht.

B. Angriffe mit Hilfe des DNS

Es gibt auch Angriffe, bei denen das DNS nicht Ziel des Angriffs ist, sondern lediglich als Hilfsmittel zur Durchführung des Angriffs genutzt wird. Dazu gehören z.B. Phishing-Angriffe, bei denen das Opfer durch manipulierte DNS-Antworten auf einen falschen Server umgeleitet wird. Auch lassen sich durch das DNS DoS-Angriffe auf andere Systeme

ausführen. DNS arbeitet in der Regel mit dem verbindungslosen Protokoll UDP. Anfragepakete an einen Server sind in der Regel deutlich kleiner als von dem Server verschickte Antwortpakete. Wenn ein Angreifer also einem Server eine große Menge von Anfragen mit gefälschter Absenderadresse schickt, wird dieser den Besitzer der Absenderadresse mit einer um ein vielfaches höheren Datenmenge überfluten. DNS-Server können also zur Verstärkung von DoS-Angriffen genutzt werden (*DNS Amplification Attack*)

IV. Geschichte von DNSSEC

Das folgende Kapitel soll einen kurzen Überblick über die geschichtliche Entwicklung von DNS und DNSSEC geben. Diese Übersicht basiert größtenteils auf [**?**] und [?], wo weiterführende Informationen verfügbar sind.

- **1983:** DNS wird erfunden und der erste Server implementiert
- **1988:** DNS beginnt eine Rolle im Internet zu spielen
- **1990:** schwere Sicherheitslücken im DNS werden entdeckt, die Informationen werden zurückgehalten, da quasi alle bisherigen Dienste dadurch unsicher sind
- **1995:** der Artikel von 1990 wird veröffentlicht, in der IETF beginnt man über Sicherheitserweiterungen nachzudenken
- **1999:** DNSSEC scheint bereit zum Deployment zu sein, Implementierung existiert
- **2001:** DNSSEC stellt sich als für den praktischen Einsatz unbrauchbar heraus, da es bei großen Zonen schlecht skaliert. Ein neuer Standard, *DNSSECbis*, wird entworfen aber neue Implementationen sind erforderlich. Da die alte Version von DNSSEC keine praktische Rolle spielt oder gespielt hat, wird im weiteren nur noch von DNSSEC gesprochen, wenn DNSSECbis gemeint ist.
- **2002-2003:** weitere Tests zeigen, dass DNSSECbis jetzt einsatzfähig ist
- **Oktober 2005:** Schweden führt für die TLD .se DNSSEC ein, dies ist damit die erste ccTLD[1] mit DNSSEC
- **2008:** NSEC3 wird spezifiziert um weitere Probleme zu beheben (siehe **??**)

V. Funktionsweise von DNSSEC

Um die Funktionsweise von DNSSEC zu verstehen muss man sich zunächst mit den zugrunde liegenden Ideen, Konzepten und den angestrebten Zielen vertraut machen.

A. Ziele von DNSSEC

Bei DNSSEC soll eine Ende-zu-Ende Sicherheit zwischen dem Besitzer der DNS-Zone und der abfragenden Instanz, dem *DNS-Resolver*, erreicht werden. Dabei soll sowohl vor manipulierten DNS-Anfragen und Antworten, z.B. durch einen Man-in-the-middle oder manipulierte DNS-Caches, als auch vor manipulierten DNS-Servern geschützt werden. Kein Ziel von DNSSEC ist es hingegen, die übertragenen Daten geheim zu halten (also zu verschlüsseln), DoS-Angriffe zu verhindern oder die Offenlegung von DNS-Zonen zu verhindern.

[1]Länder Top Level Domains, Country Code Top Level Domain, ccTLD, im Gegensatz zu generischen TLDs (gTLD) wie .com oder .net)

B. Ideen hinter DNSSEC

DNSSEC arbeitet mit digitalen Signaturen basierend auf asymmetrischer Cryptographie. Dabei erstellt der Zonenbesitzer ein Paar aus öffentlichem und privatem Schlüssel. Der öffentliche Schlüssel wird der DNS-Zone als neuer RR hinzugefügt und ist somit öffentlich abfragbar, der private Schlüssel wird nicht auf dem DNS-Server gespeichert. Für jeden RR der Zone wird nun eine digitale Signatur mit dem privaten Schlüssel erzeugt, die erstellten Signaturen werden ebenfalls der Zone als neue RRs hinzugefügt. Die gesamte Signatur der Zone kann offline erfolgen, anschließend kann die Zonendatei auf die (möglicherweise nicht einmal vertrauenswürdigen) DNS-Server (*Nameserver*) übertragen werden. Der private Schlüssel wird nicht auf den Servern benötigt. Ein DNS-Resolver, der einen Namen auflösen will, kann aus dem DNS nicht nur den entsprechenden Wert ermitteln, sondern auch die zugehörige Signatur und den öffentlichen Schlüssel, womit er die Authentizität des empfangenen Wertes bestätigen kann.

Allerdings muss auch die Authentizität des öffentlichen Schlüssels bestätigt werden, bevor diesem getraut werden kann. Dazu existiert in der übergeordneten Zone neben dem NS-Resource Record, der die eigentliche Delegation der untergeordneten Zone darstellt, ein *Delegation-Signer (DS)* RR. Dieser erhält einen Hash des öffentlichen Schlüssels der untergeordneten Zone. Der DS-Eintrag gehört zur übergeordneten Zone und ist daher mit deren öffentlichem Schlüssel signiert. Kann ein Resolver also dem öffentlichen Schlüssel einer Zone vertrauen, so kann er durch Nutzung der DS-Einträge auch allen öffentlichen Schlüsseln von untergeordneten Zonen vertrauen. Im Idealfall kann ein Resolver also ausgehend vom öffentlichen Schlüssel der Root-Zone (.) allen anderen Zonen vertrauen, da er nach und nach über die DS-RRs die öffentlichen Schlüssel aller untergeordneten Zonen verifizieren kann. Der öffentliche Schlüssel der Root-Zone kann allerdings nicht anderweitig bestätigt werden und muss dem Resolver daher bekannt sein. Selbstverständlich können auch weitere Schlüssel dem Resolver bekannt gemacht werden, um die Auflösung zu vereinfachen.

C. Weitere Abstraktionen

Bei DNSSEC muss ein Kompromiss bezüglich der Schlüssellänge eingegangen werden. Kürzere Schlüssel ermöglichen es, Signaturen schneller zu erstellen und zu überprüfen. Darüber hinaus benötigen die damit erzeugten Signaturen weniger Speicherplatz in den Zonen und verursachen weniger Datenverkehr bei Abfragen. Leider sind kurze Schlüssel weniger sicher gegen Angriffe und müssen daher öfter gewechselt werden. Der Wechsel von Schlüsseln ist aufwendig, da auch der Verwalter der übergeordneten Zone über den neuen Schlüssel informiert werden und den DS-RR aktualisieren muss. Bei DNSSEC wird daher für jede Zone nicht nur ein Paar aus öffentlichem und privatem Schlüssel gebildet, sondern zwei Paar. Ein Paar wird als *Key Signing Key (KSK)* bezeichnet, das andere als *Zone Signing Key (ZSK)*. Der in der übergeordneten Zone existierende DS-RR verweist dabei auf den KSK. Mit diesem wird der

ZSK signiert (beide öffentlichen Schlüssel sowie die mit dem KSK erzeugte Signatur des ZSK befinden sich in der untergeordneten Zone). Mit dem ZSK werden nun wiederum alle anderen Einträge der Zone signiert. Dieses Vorgehen bietet einige entscheidende Vorteile:

- Der KSK kann sehr lang und sicher sein und muss daher nicht oft gewechselt werden.
- Der ZSK kann kürzer sein, dadurch wird weniger Platz benötigt und große Zonen können schneller signiert werden.
- Durch die hohe Lebensdauer des KSK muss der DS-RR nicht oft aktualisiert werden.
- Der ZSK kann oft gewechselt werden, da keine weitere Instanz am Schlüsseltausch beteiligt werden muss. Der KSK bleibt gleich und damit kann der neue ZSK signiert werden. Mit diesem wiederum werden alle Zoneneinträge neu signiert.
- Bei dynamisch aktualisierten Zonen kann der private Schlüssel des ZSK auf dem Server verbleiben um Änderungen automatisch zu signieren, der private Schlüssel des KSK hingegen bleibt offline. Nach einer Kompromittierung des Servers kann die Sicherheit durch Wechsel des ZSK schnell wieder hergestellt werden, der KSK kann dabei beibehalten werden.

D. Ablauf einer Schlüsselüberprüfung durch den Resolver

1) Der Resolver fragt einen Eintrag www.example.com ab (A-Record)
2) von den Nameservern für die Rootzone (.) erhält er einen NS-Eintrag für die com. Zone und einen DS-Eintrag für den dortigen KSK sowie den ZSK und die Signatur für die .-Zone
3) der KSK für die .-Zone ist bekannt, damit wird der .-ZSK bestätigt, damit wiederum der DS-Eintrag
4) von dem com.-*Nameserver (NS)* wird der dortige KSK, ZSK, sowie NS- und DS-RRs für example.com. abgefragt
5) mit dem vorher durch den DS-RR bestätigten .com-KSK wird der dortige ZSK, damit der DS-RR bestätigt
6) von dem example.com-NS wird KSK, ZSK, www.example.com-A-RR und dessen Signatur abgefragt
7) Diese werden der Reihe nach bestätigt

E. Wem vertraut der Resolver

Eine der Kernfragen bei DNSSEC ist die Frage, wem ein Resolver vertrauen muss und wem nicht. Wie aus dem oben erläuterten Ablauf hervorgeht muss der Resolver bei der Auflösung des Eintrags www.example.com genau folgenden Instanzen und Gegebenheiten vertrauen:

- seinem einprogrammierten .-Schlüssel, da der Schlüssel für die .-Zone nicht anderweitig bestätigt werden kann
- den Besitzern der .-Zone, com.-Zone und example.com.-Zone, da der Besitzer einer Zone den dazugehörigen privaten Schlüssel kennt und daher alle darin enthaltenen RRs und Delegierungen auf untergeordnete Zonen ändern kann.
- der Sicherheit der eigenen Ausführung. Wenn der Resolver selber durch Viren oder ähnliches manipuliert wurde ist selbstverständlich keine Sicherheit mehr möglich.

Der Resolver muss jedoch keiner der folgenden Instanzen vertrauen und ist daher vor allen Manipulationen von diesen geschützt:

- einem Serverbetreiber für einen der beteiligten Nameserver
- den Nameservern seines Providers
- anderen zwischengeschalteten DNS-Caches
- der Sicherheit des Kommunikationsweges mit den Servern (z.B. können Antworten, die durch einen Man-in-the-Middle manipuliert wurden erkannt werden)

Einen Sonderfall bilden so genannte Stub-Resolver, die in vielen Betriebssystemen eingebaut sind. Diese sind keine vollwertigen Resolver, die Nameserver rekursiv abfragen und die Antworten per DNSSEC verifizieren können, stattdessen werden alle Anfragen lediglich an einen anderen, vollwertigen Resolver weitergeleitet und von diesem die fertige Antwort erhalten. In diesem Fall, wenn also der Stub-Resolver die Antwort nicht selber verifiziert, muss dieser zusätzlich dem von ihm verwendeten Resolver und der Sicherheit des Kommunikationskanals zu diesem vertrauen. Letzteres kann z.B. mittels IPSec oder TLS sichergestellt werden, dies ist aber nicht Thema dieser Arbeit.

F. Authenticated denial of existence

Eine Besonderheit bei DNSSEC ist die so genannte *authenticated denial of existence*. Wie bisher gezeigt wurde, kann ein Resolver einen Eintrag effektiv auf Authentizität überprüfen. Dadurch können sowohl Manipulationen am Inhalt eines Eintrags als auch das vorspiegeln von nicht existierenden Einträgen verhindert werden. Jedoch bleibt das Problem, dass ein Angreifer die Existenz eines in Wahrheit existierenden RRs verleugnen kann. Der Resolver kann dies erst einmal nicht erkennen, da er ja keine Signatur für das Nicht-Existieren eines RRs erhalten kann. Bei DNSSEC wurde jedoch auch dieses Problem gelöst. Dazu werden zunächst alle Einträge einer Zone alphabetisch sortiert, anschließend wird zwischen je zwei RRs mit unterschiedlichem Namen ein weiterer neuer RR eingefügt. Dieser hat den Namen des direkt davor stehenden Eintrags und als Wert den Namen des folgenden Eintrags und gibt an, dass zwischen diesen beiden Einträgen in alphabetischer Ordnung keine weiteren Einträge liegen. Als Typ wird der neue Typ *NSEC* verwendet. Diese neu erstellten Einträge werden nun wie alle anderen Einträge der Zone signiert. Wenn ein Server nun eine negative Antwort verschickt (also aussagt, dass ein angefragter Name nicht existiert), so schickt er mit dieser Antwort den NSEC-RR, in dessen Bereich der angeforderte Name liegen müsste, würde er existieren und dessen Signatur. Der anfragende Resolver überprüft nun die Signatur und ob der NSEC-Eintrag wirklich den angeforderten Schlüssel enthalten müsste. Ist dies erfolgreich kann der Resolver sicher sein, dass der angefragte Eintrag wirklich nicht existiert, da sonst kein gültiger NSEC-Eintrag für den Bereich existieren könnte.

Wenn beispielsweise die beiden Einträge a.example.com und c.example.com existieren, aber b.example.com angefordert wird, so teilt der Server dem Resolver mit, dass auf den Eintrag a.example.com direkt c.example.com folgt. Daher weiß der Resolver, dass b.example.com nicht existiert. Ebenso ist durch den NSEC-Eintrag bewiesen, dass a1.example.com, bcde.example.com, etc. nicht existieren.

VI. Änderungen am bisherigen DNS

Obwohl DNSSEC weitgehend abwärtskompatibel ist, sind doch einige Änderungen am bisherigen DNS erforderlich. Wie im Abschnitt **??** erwähnt, kennt das DNS verschiedene Typen von Resource Records. Weit verbreitete Typen sind z.B. A für die Abbildung von Namen in IPv4-Adressen, AAAA für die Abbildung von Namen in IPv6 Adressen, CNAME für Aliase, PTR für die Abbildung von IP-Adressen in Namen, TXT für beliebige Texte sowie eine Reihe weiterer, mehr oder weniger oft verwendeter Typen. DNSSEC führt nun eine Reihe neuer Typen ein, im Detail sind das die folgenden:

- **RRSIG** speichert Signaturen für andere RRs
- **DNSKEY** speichert die öffentlichen Schlüssel von KSK und ZSK, mit denen die Signaturen verifiziert werden können
- **DS** speichert einen Zeiger auf den DNSKEY einer untergeordneten Zone (in der übergeordneten Zone)
- **NSEC** speichert, welche Namen in einer Zone nicht existieren (siehe Abschnitt **??**)
- **NSEC3** ist ein sichererer Nachfolger von NSEC (siehe Abschnitt **??**)
- **NSEC3PARAM** speichert Hilfsinformationen für NSEC3

Darüber hinaus sind einige kleinere Änderungen am Protokoll erforderlich, da durch die Übermittlung von Schlüsseln und Signaturen die Pakete deutlich größer werden als bisher. Dies führt dazu, dass alle Komponenten, also Nameserver, Resolver, Caches, etc. aktualisiert werden müssen um DNSSEC zu unterstützen. Allerdings kann diese Aktualisierung schrittweise erfolgen, da neues und altes System zu einander hinreichend kompatibel sind, jedoch entsteht für alte Komponenten kein Sicherheitsgewinn durch DNSSEC.

A. Neue Schlüsseltypen im Detail

1) RRSIG: Ein Beispiel für einen RRSIG-RR könnte wie folgt aussehen:

```
host.example.com. 86400 IN RRSIG A 5 3
     86400 20030322173103 (
     20030220173103 2642 example.com.
     oJB1W6WNGv+ldvQ3WDG0MQkg5IEhjRip8WTr
     PYGv07h108dUKGMeDPKijVCHX3DDKdfb+v6o
     B9wfuh3DTJXUAfI/M0zmO/zz8bW0Rznl8O3t
     GNazPwQKkRN20XPXV6nwwfoXmJQbsLNrLfkG
     J5D6fwFm8nN+6pBzeDQfsS3Ap3o= )
```

host.example.com. ist dabei der Name des RR, 86400 ist die Gültigkeitsdauer (TTL) dieses Eintrags, IN gibt an, dass es um die DNS-Klasse Internet geht. RRSIG ist der Typ des RR, A besagt, dass dies eine Signatur für einen oder mehrere A-Records ist. 5 steht für das verwendete Verschlüsselungsverfahren, 3 ist der Labelcount, welcher für Wildcard-RRs[2] benötigt wird. 86400 gibt noch einmal die TTL an. Dies ist erforderlich, weil der TTL-Wert z.B. durch zwischengeschaltete Proxys verändert werden kann und die Signatur dadurch ungültig werden würde. Daher wird die TTL zum Zeitpunkt des signierens hier noch einmal wiederholt. Anschließend sind die Zeitstempel angegeben, bis wann und ab wann die Signatur gültig ist. example.com. ist der Name dessen, der die Signatur ausgestellt hat, hier also der Besitzer der Zone example.com., wo auch nach dem passenden öffentlichen Schlüssel gesucht werden muss, welche durch den Schlüsselidentifizierer 2642 erkannt werden kann. Anschließend folgt die eigentliche Signatur in Base64-Kodierung.

2) DNSKEY: Als nächstes soll das Format des RR-Typs DNSKEY vorgestellt werden, welches zur Speicherung von öffentlichen Schlüsseln dient.

```
example.com. 86400 IN DNSKEY 256 3 5
     ( AQPSKmynfzW4kyBv015MUG2DeIQ3
       Cbl+BBZH4b/0PY1kxkmvHjcZc8no
       kfzj31GajIQKY+5CptLr3buXA10h
       WqTkF7H6RfoRqXQeogmMHfpftf6z
       Mv1LyBUgia7za6ZEzOJBOztyvhjL
       742iU/TpPSEDhm2SNKLijfUppn1U
       aNvv4w==  )
```

Hierbei lassen sich auf Anhieb gewisse Ähnlichkeiten mit dem obigen Beispiel erkennen. Name, TTL und Klasse sind wieder wie oben, der Typ ist diesmal DNSKEY. die folgenden drei Zahlen geben Informationen zum Verschlüsselungsverfahren, Schlüssellänge, etc. Danach folgt der eigentliche Schlüssel, wieder in Base64-Kodierung.

3) DS: Ein weiterer wichtiger Typ ist der Delegation Signer DS.

```
dskey.example.com. 86400 IN DNSKEY 256 3 5
     ( AQOeiiR0GOMYkDshWoSKz9Xz
       fwJr1AYtsmx3TGkJaNXVbfi/
       2pHm822aJ5iI9BMzNXxeYCmZ
       DRD99WYwYqUSdjMmmAphXdvx
       egXd/M5+X7OrzKBaMbCVdFLU
       Uh6DhweJBjEVv5f2wwjM9Xzc
       nOf+EPbtG9DMBmADjFDc2w/r
       ljwvFw==
     ) ;  key id = 60485

dskey.example.com. 86400 IN DS 60485 5 1
     ( 2BB183AF5F22588179A53B0A
       98631FAD1A292118 )
```

In diesem Beispiel sind zwei RRs aufgeführt, zuerst einmal wieder ein DNSKEY-RR, der in der untergeordneten Zone gespeichert ist. Aus dem Schlüssel lässt sich nach einem

[2]Die genaue Funktionsweise von Wildcard-RRs kann hier nicht erklärt werden, dafür sei auf weiterführende Literatur verwiesen

ebenfalls im DNSSEC-Standard festgelegten Verfahren die Schlüssel-ID 60485 berechnen. In der übergeordneten Zone wird nun der DS-RR eingefügt. Dieser hat natürlich auch wieder einen Namen, TTL, Klasse und den Typ DS. Anschließend folgt die Key-ID und weitere Informationen zum Schlüssel. Zuletzt folgt eine Prüfsumme des Schlüssels in Hexadezimaldarstellung.

4) NSEC: Besonders einfach ist der Aufbau von NSEC-Einträgen.

```
alpha.example.com. 86400 IN NSEC
     host.example.com.
     (A MX RRSIG NSEC TYPE1234 )
```

Auch hier sind wieder die üblichen Felder vorhanden, der Typ ist NSEC. Der Name des RR ist in diesem Fall alpha.example.com., der Wert ist host.example.com., dieser NSEC-Eintrag gibt also an, dass es keine RRs gibt, die in alphabetischer Ordnung zwischen alpha und host liegen würden. Anschließend folgt noch eine Aufzählung der Typen, für die RRs mit Namen alpha.example.com. existieren, in diesem Fall sind das A, MX, RRSIG, NSEC und TYPE1234. Mit Hilfe dieser Informationen lässt sich auch feststellen, ob ein bestimmter RR-Typ für einen Namen nicht existiert.

5) NSEC3 und NSEC3PARAM: Abschließend seinen noch die Formate der Schlüsseltypen NSEC3 und NSEC3PARAM erwähnt, weitere Informationen zur Verwendung dieser Schlüssel finden sich im Abschnitt **??**.

```
example.com. 86400 IN NSEC3PARAM 1 0 12
     aabbccdd

0p9mhaveqvm6t7vbl5lop2u3t2rp3tom.example.com
     86400 IN NSEC3
     1 1 12 aabbccdd (
     2t7b4g4vsa5smi47k61mv5bv1a22bojr
     MX DNSKEY NS SOA NSEC3PARAM RRSIG )
```

Der RR-Typ NSEC3PARAM gibt nur einige Hilfsinformationen zur verwendeten Hashfunktion sowie den verwendeten Salt[3] aabbccdd an und muss nicht weiter beschrieben werden. Der RR-Typ NSEC3 hat große Ähnlichkeit mit dem Typ NSEC, allerdings werden hier nicht direkt die Namen dieses und des nächsten Schlüssels verwendet, sondern Hashwerte von diesen. Als Typ ist NSEC3 angegeben, anschließend folgen Informationen zum verwendeten Hash-Verfahren, der Salt und Informationen zur Behandlung von Wildcard-RRs. Genauere Informationen hierzu finden sich in [**?**]. Auch hier werden wieder die RR-Typen angegeben, die bei dem RR mit dem Hash 0p9mhaveqvm6t7vbl5lop2u3t2rp3tom existieren. 2t7b4g4vsa5smi47k61mv5bv1a22bojr ist der Hash des nächsten existierenden Namens.

VII. Bisherige Verbreitung von DNSSEC

Bisher ist DNSSEC noch nicht sonderlich weit verbreitet, von den Länder Top Level Domains (ccTLDs) verwenden bisher lediglich die folgenden DNSSEC:

- .bg (Bulgarien)
- .br (Brasilien)
- .cz (Tschechien)
- .pr (Puerto Rico)
- .se (Schweden)

Noch schlechter sieht es bei den *generischen Top Level Domains (Generic Top Level Domain, gTLD)* aus, von denen DNSSEC bisher lediglich bei .museum Verwendung findet. Allerdings ist fest geplant, dass die Zone .gov ab Januar 2009 DNSSEC verwendet, bis Ende 2009 sollen des weiteren die Zone .mil sowie alle Unterzonen von .mil und .gov DNSSEC verwenden. Zur Zeit testet die IANA elf IDN-TLDs (internationalisierte Top Level Domains), bei all diesen wird auch DNSSEC getestet.

Darüber hinaus existierten in den Zonen .com, .net, .arpa TLDs Testprojekte für DNSSEC, die aber inzwischen alle wieder eingestellt wurden. Auch in untergeordneten Zonen, also Second- und Third-Level Domains oder noch tieferen Zonen wird DNSSEC verwendet oder wurde bereits damit getestet.

VIII. Probleme von DNSSEC

Leider bringt DNSSEC auch einige Probleme mit sich. Diese Probleme haben verschiedenste Ursachen. Zunächst einmal ist DNSSEC nicht in der Lage alle Probleme des bisherigen DNS zu beheben. Denial of Service (DoS) Angriffe sind weiterhin möglich und werden durch die zusätzlichen Berechnungen für die kryptographischen Überprüfungen und die größeren Datenmengen sogar noch verschärft. Ebenso sind DNS Amplification Attacks weiterhin möglich und durch die bei DNSSEC deutlich größeren Antwortpakete wird auch dieses Problem noch verschlimmert. Darüber hinaus gibt es bei der Einführung von DNSSEC auch diverse politische Probleme. Historisch bedingt haben die USA eine große Kontrolle über das Internet und besonders das DNS. Diese wollen die USA nicht aufgeben, viele andere Nationen hingegen wünschen keine einseitig amerikanische Dominanz des Internets mehr. Diese Problematik trifft DNSSEC, wenn es darum geht, wer die privaten Schlüssel für die Root-(.)-Zone erhält, ebenso ist die Verwaltung der Schlüssel für die generischen Top Level Domains, vor allem .com und und .net noch nicht geklärt. Hierzu gab es bereits viele Vorschläge, diverse US-Behörden, die IANA, die UN und weitere wurden diskutiert. Auch wurde darüber nachgedacht, die Schlüssel auf mehrere Länder aufzuteilen, so dass eine Signierung nur gemeinsam erfolgen kann. Eine Lösung zeichnet sich indes nicht ab. Auch technisch gesehen sind längst nicht alle Probleme aus der Welt. Durch die NSEC-Einträge und die Signaturen steigt die Größe der Zonendateien um 100-500%, was dazu führt, dass die Nameserver gerade für die großen Zonen deutlich mehr Leistung benötigen. Auch wurden bisher im DNS nur relative Zeiten in Form der TTL-Werte verwendet, DNSSEC verwendet bei den Signaturen aber absolute Zeitstempel für die Gültigkeitszeiträume. Dadurch werden erstmals auf allen beteiligten Systemen korrekt funktionierende Rechneruhren vor-

[3]Zur Bedeutung von Salt-Werten für die Sicherheit von kryptographischen Hashes sei hier auf Literatur zum Thema Kryptographie verwiesen

ausgesetzt. Darüber hinaus wurden bei vielen ADSL-Routern schwere Fehler bei der Implementation von DNS gemacht, was dazu führt, dass die DNS-Komponenten dieser Router nicht mehr funktionieren, wenn in den Antworten DNSSEC-Daten enthalten sind. Auch organisatorische Probleme sind zu lösen. Während bisher Zonen, an denen wenig Änderungen durchgeführt werden mussten, teilweise über Jahre ohne Veränderungen und Wartungen auskamen, muss bei DNSSEC jede Zone in regelmäßigen, nicht zu langen Zeitintervallen mit jeweils neuen Schlüsseln neu signiert werden. Für diese regelmäßige Wartung, d.h. den Wechsel des ZSK, muss ein Ablauf zuverlässig etabliert werden. Auch der seltenere Tausch des KSK muss zuverlässig funktionieren, da bei Fehlern hierbei ganze Zweige des DNS unerreichbar werden können, wenn eine einzige Signatur nicht verifiziert werden kann. Auf Seite der Resolver muss eine sichere Verteilung des Root-Schlüssels gewährleistet werden, sowohl erstmalig, als auch bei eventuell notwendigen Änderungen. Auf zwei besondere Probleme von DNSSEC soll im folgenden noch genauer eingegangen werden:

A. Problem des Schlüsselwechsels

Da DNSSEC keinerlei Möglichkeit vorsieht, einmal erstellte Signaturen zurück zu rufen oder vor dem Ablauf ihrer Gültigkeit für ungültig zu erklären, muss die Gültigkeitsperiode für Schlüssel und Signaturen recht gering gewählt werden, damit im Falle einer Kompromittierung eines Schlüssels die zugehörigen Signaturen nicht mehr allzu lang gültig bleiben. Dies führt dazu, dass Schlüssel regelmäßig geändert und neue Signaturen erzeugt werden müssen. Da DNS jedoch keine einfache, zentrale Datenbank ist, sondern Caches verwendet, kann es passieren, dass RRs noch an Resolver ausgeliefert werden, wenn sie in der authoritativen Version der Zone schon nicht mehr vorhanden sind. Dies führt zu Problemen, wenn z.B. ein Schlüssel im Cache vorhanden ist, nicht aber die zugehörige Signatur, da diese evtl. nicht mehr erhältlich ist. Das gleiche Problem tritt auf, wenn eine Signatur im Cache vorhanden, der Schlüssel aber nicht mehr erhältlich ist und führt im Endeffekt dazu, dass die Überprüfung der Signatur fehlschlagen würde. Somit wären alle untergeordneten Einträge und Domänen für DNSSEC beachtende Resolver nicht mehr erreichbar! Für dieses Problem gibt es nur eine Lösung: neue Schlüssel und Signaturen werden in der Zonendatei eine Zeit lang parallel mit den alten vorgehalten. Somit können zu allem, was evtl. noch in Caches vorhanden ist, die zugehörigen Daten weiterhin abgerufen werden und die Cache-Einträge werden nach und nach durch die jeweils neueren Versionen ersetzt. Der Nachteil an dieser Lösung ist natürlich, dass alle Signaturen mehrfach existieren und bei einer Anfrage immer alle Versionen von Signaturen und Schlüsseln ausgeliefert werden müssen. Dadurch steigen Transfervolumen und Zonengröße noch einmal deutlich.

B. Zone enumeration

Ein weiterer sehr wichtiger Fehler von DNSSEC besteht darin, dass der Inhalt von Zonen offengelegt wird. Wie bereits erläutert wurde, ist es möglich, über die NSEC-Einträge nicht nur zu erfahren, dass ein gewisser Eintrag nicht existiert, sondern auch, wie der nächste existierende Eintrag heißt. Ausgehend von diesem Namen kann man nun wiederum den nächsten existierenden Namen mittels der NSEC-Einträge ermitteln. Durch wiederholen dieses Vorgangs ist es möglich, alle Einträge einer Zone mittels der verknüpfenden NSEC-Einträge aufzuzählen und somit den gesamten Inhalt der Zone zu ermitteln. Dies wird als *Zone enumeration* oder *Zone walking* bezeichnet. Während dies zwar dem ursprünglichen Konzept des DNS, nach dem alle Informationen öffentlich sein sollten, nicht widerspricht, ist es heute aus einer Reihe von Gründen oft unerwünscht. Durch die Kenntnis aller Einträge einer Zone lassen sich oftmals Informationen über den internen Aufbau eines Netzwerks, die darin vorhandenen Systeme und deren Rollen und ähnliches ableiten. Dieses Wissen kann für einen späteren Angriff auf dieses Netzwerk sehr hilfreich sein. Auch entstehen in vielen Ländern rechtliche Probleme. So meint zum Beispiel die DENIC, DNSSEC sei unvereinbar mit deutschen Datenschutzgesetzen (siehe [**?**]) und DNSSEC sei daher für die .de-Zone nicht einsetzbar. Viele andere (vor allem europaische) Länder-Registries teilen diese Bedenken. Dieses Thema wird oft kontrovers diskutiert, eben weil eigentlich alle Informationen im DNS öffentlich sind. Jedoch gibt es einen Unterschied, ob nur einzelne Datensätze abgefragt werden können, oder eine Liste aller Datensätze erstellt werden kann. Oft hört man auch das Argument ‘Domain Names können auch anders, z.B. durch Wörterbuchangriffe gefunden werden’, jedoch lässt sich auch das leicht entkräften, wenn man sich die von der DENIC veröffentlichten Zahlen ansieht (siehe [?]). So reichen das deutsche Wörterbuch, das englische Wörterbuch und die Wörterlisten des bekannten Passwortknackers ‘John the Ripper’ gerade einmal aus, um etwa ein Prozent der .de-Zone zu erraten. Selbst ein durchprobieren aller Domainnamen mit einer Länge von acht Zeichen liefert nur dreizehn Prozent der .de-Zone. Selbst wenn man im Besitz der größten Zone des Internets, .com, wäre, könnte man damit nur 42 Prozent der recht kleinen .nl-Zone durch ausprobieren herausfinden. Es scheint also keinen gangbaren Weg zu geben, den Inhalt einer DNS-Zone auch nur annähernd vollständig zu ermitteln. Zwar sind auch heute schon viele Zonen durch falsch konfigurierte Nameserver öffentlich einsehbar, jedoch scheint der Trend eher zu einer Absicherung der Zonendaten zu gehen, wie ich bei meinen Tests der Top-Level-Domains ermitteln konnte. Während im Dezember 2004 noch 141 von 258 TLDs und damit etwa 55% das Herunterladen der Zonendateien von mindestens einem authoritativen Nameserver oder dem im SOA-Eintrag angegebenen Server erlaubten, sind dies vier Jahre später im Dezember 2008 nur noch 78 von 280 TLDs. Nimmt man dazu noch die sechs produktiven Zonen, die DNSSEC verwenden und die elf IDN-Testzonen, welche allesammt die Auflistung der Einträge durch Zone enumeration ermöglichen, sinkt der Prozentsatz der offenliegenden Zonen um 21% auf nun etwa 34%. Auch sind die ungeschützten Zonen kleiner geworden, während 2004 noch große Zonen wie .be mit etwa 42MB und etwa 900.000 RRs und .fi mit etwa 11MB und etwa

235.000 RRs öffentlich waren, sind die größten ungesicherten Zonen heute .sg, .ma und .kz mit 5MB, 4,5MB und 4,2MB.[4]

C. Die Lösung: NSEC3

Zum Glück existiert seit März 2008 eine Lösung für das Problem der Zone enumeration, die es ermöglicht Zonendaten geheim zu halten, dabei aber trotzdem die Nicht-Existenz eines RRs nachzuweisen. Diese Erweiterung von DNSSEC wird als *hashed authenticated denial of existence* oder auch als *NSEC3* bezeichnet. Dabei wird fast genauso wie bei NSEC verfahren, jedoch wird von jedem Namen zuerst ein kryptographischer Hash gebildet, anschließend werden diese Hashes sortiert und die NSEC3-Records erstellt und signiert. Auch mit den Hashes lässt sich nachweisen, dass ein RR nicht existiert, aufgrund der Unumkehrbarkeit der Hash-Funktionen ist es aber nicht möglich, die ursprünglichen Namen wieder zu ermitteln. Dabei finden auch Salts Verwendung, um das Knacken der Hashes zu erschweren, ebenso gibt es Methoden, um Hash-Kollisionen zu vermeiden. NSEC3 ermöglicht es unter anderem der DENIC, DNSSEC einzusetzen, ohne dabei den Datenschutz zu vernachlässigen und bereitet somit den Weg für eine weite Verbreitung von DNSSEC.

IX. Zusammenfassung

Zusammenfassend lässt sich feststellen, dass eine DNS-weite Einführung von DNSSEC technisch möglich ist. Essentiell dabei ist eine Einführung von DNSSEC für die Root-(.)-Zone, da ansonsten für jeden Resolver eine Liste mit einer Vielzahl von vertrauenswürdigen öffentlichen Schlüsseln gepflegt werden muss. Der Aufwand für die Umstellung aller existierenden Systeme (vor allem aller Resolver) ist sehr hoch, dies ist jedoch kein Hinderniss, da die Abwärtskompatibilität vorhanden und eine schrittweise Einführung somit möglich ist. Dabei profitieren zwar ältere Resolver nicht unmittelbar, bis DNSSEC weiträumig eingeführt ist dürften allerdings viele davon ohnehin durch neuere Systeme ersetzt worden sein.

X. Ausblick

Leider steht gerade mit Blick auf die Rootzone zu befürchten, dass politische Probleme die Verbreitung von DNSSEC weiter verzögern dürften. Gerade die jüngere Entwicklung (siehe [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**] und [**?**]) macht auch keine allzu großen Hoffnungen auf eine baldige Einigung. Des Weiteren benötigt die Aktualisierung der Kern-Infrastruktur des DNS, also aller großen Nameserver und Caches, Zeit. Allerdings existieren massive Anstrengungen zur Einführung von DNSSEC, z.B. bei .gov und .mil. Clientseitig hat Microsoft angekündigt, dass Windows 7 und Windows Server 2008 SP2 DNSSEC unterstützen werden. Wie bei allen neuen Technologien, welche weit verbreitete und bewährte Technik ersetzen sollen, besteht bei der Einführung auch hier ein Henne-Ei Problem, solange kaum Zonen DNSSEC verwenden ist das Interesse gering, die Resolver aufzurüsten, so lange kaum Resolver davon profitieren scheuen viele Zonenbesitzer den zusätzlichen Aufwand von DNSSEC. Allerdings eröffnet DNSSEC auch ganz neue Möglichkeiten, das DNS zu nutzen. So ist es z.B. denkbar, E-Mail Zertifikate für Benutzer im DNS zu speichern und somit zuverlässig global zu verbreiten, davon könnte die Verwendung von verschlüsselten und signierten E-Mails stark profitieren und deren Verbreitung zunehmen. Auch sind ganz neue Möglichkeiten der Spam-Bekämpfung denkbar.

[4]Die Größe der DNSSEC geschützten Zonen wurde nicht bestimmt. Insbesondere könnte .se um einige größer sein.

Literatur

[1] B. Claise, "IPFIX protocol specifications," Internet-Draft, draft-ietf-ipfix-protocol-07, December 2004.

[2] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-based IP traceback," in *ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2001.

[3] A. Belenky and N. Ansari, "IP traceback with deterministic packet marking," *IEEE Communications Letters*, vol. 7, no. 4, pp. 162–164, 2003.

[16] M. Ermert and V. Briegleb. (2007, November) Igf: Politische und technische probleme bei dnssec. heise online. [Online]. Available: http://www.heise.de/netze/IGF-Politische-und-technische-Probleme-bei-DNSSEC--/news/meldung/99000

[17] M. Ermert and A. Wilkens. (2007, Juni) Icann soll rasch rootzone mit dnssec signieren. heise online. [Online]. Available: http://www.heise.de/netze/ICANN-soll-rasch-Rootzone-mit-DNSSEC-signieren--/news/meldung/91501

[18] M. Ermert and J. Kuri. (2007, March) Department of homeland security will den masterschlüssel fürs dns. heise online. [Online]. Available: http://www.heise.de/netze/Department-of-Homeland-Security-will-den-Masterschluessel-fuers-DNS--/news/meldung/87620

[19] M. Ermert and V. Briegleb. (2007, May) Icann soll signatur der dns-rootzone übernehmen. heise online. [Online]. Available: http://www.heise.de/netze/ICANN-soll-Signatur-der-DNS-Rootzone-uebernehmen--/news/meldung/89730

[20] M. Ermert and V. Briegleb. (2007, May) Rootzone-sicherung sorgt weiter für debatten [update]. heise online. [Online]. Available: http://www.heise.de/newsticker/Rootzone-Sicherung-sorgt-weiter-fuer-Debatten-Update--/meldung/89997

[21] M. Ermert and A. Wilkens. (2006, March) Igf: Diskussion über den masterschlüssel für die dns-aufsicht. heise online. [Online]. Available: http://www.heise.de/security/IGF-Diskussion-ueber-den-Masterschluessel-fuer-die-DNS-Aufsicht--/news/meldung/80479

[22] B. Müller. (2008, Juli) Improved dns spoofing using node re-delegation. SEC Consult Unternehmensberatung GmbH. [Online]. Available: http://www.sec-consult.com/files/Whitepaper-DNS-node-redelegation.pdf

[23] S. M. Bellovin. (2005) Using the domain name system for system break-ins. AT&T Bell Laboratories. [Online]. Available: http://citeseer.ist.psu.edu/bellovin95using.html

[24] R. Glauberman. (2004, December) [full-disclosure] again: zone transfers, a spammer's dream? Mailing List. [Online]. Available: http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2004-12/0812.html