Sicherheit im Internet

SS 2004

Sicherheiterweiterungen im DNS nach RFC 2535

- Referatausarbeitung -

Dozent:	Prof. Dr. P. Trommler
Studentin:	Ursula Loch
Matrikelnummer:	690214

INHALTSVERZEICHNIS

1) Einführung

Datenintegrität und Authentifizierung spielen für sicherheitsbewusste Resolver[1] eine große Rolle. Um dies gewährleisten zu können, wurden für das Domain Name System (DNS) Sicherheitserweiterungen entwickelt, welche sich auf die Benutzung von verschlüsselten digitalen Signaturen stützen. Ähnlich wie die Resource Records[2] werden auch diese digitalen Signaturen in gesicherten Zonen[3] gehalten. Diese Erweiterungen gewährleisten auch die Speicherung der authentifizierten Public Keys im DSN, was sowohl die Sicherheit des DNS, als auch die allgemeinen Dienste zur Verbreitung der Public Keys unterstützt. Die zu Beginn mit einigen Schlüsseln konfigurierten sicherheitsbewussten Resolver können nun, mit Hilfe der gespeicherten Schlüssel, zusätzlich authentifizierende Schlüssel anderer Zonen lernen.
Für die Verwendung von anderen Protokollen können zu deren Unterstützung mit DNS Namen verknüpfte Schlüssel abgefragt werden. Diesbezüglich wurden schon für eine Vielzahl von Schlüsseltypen und Algorithmen Vorkehrungen getroffen.
Neben den aufgeführten Punkten ermöglichen die der DNS Sicherheitserweiterungen (DNSSec Erweiterungen) noch die optionale Authentifizierung von DNS Protokolltransaktionen und Anforderungen.

2) Einordnung der DNSSec in das OSI-Schichtenmodell

Bevor die einzelnen Aspekte der Sicherheitserweiterungen im Domain Name System genauer dargestellt werden, soll zuerst anhand einer Grafik deren Einordnung in das OSI-Schichtenmodell erfolgen:

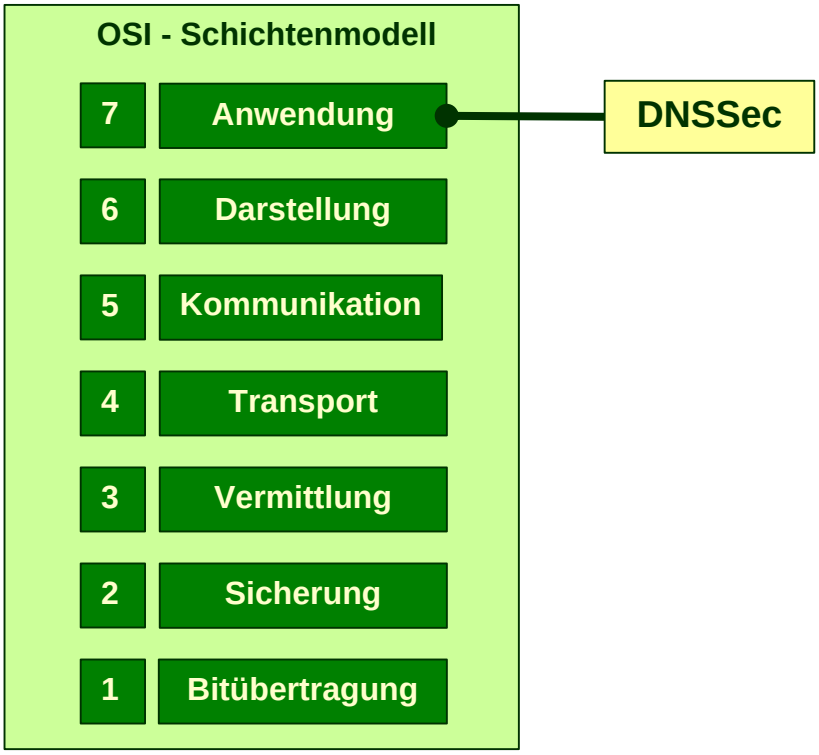

Abb. 1: Einordnung der DNSSec in das OSI-Schichtenmodell

[1] Kurzbeschreibung „DNS-Resolver“ im Anhang A
[2] Kurzbeschreibung „Resource Records“ im Anhang B
[3] Kurzbeschreibung „Zonen und DNS Sicherheit“ im Anhang C

3) Überblick Erweiterungen

Im Allgemeinen unterstützen die DNSSec Erweiterungen drei Dienste:
Der erste dient der Verteilung der für die DNSSec notwendigen Schlüssel. Anfänglich war sogar eine Nutzung der DNSSec als Public Key Infrastructure (PKI) geplant, jedoch wurde dieses Vorhaben wieder verworfen. Da die DNS Schlüssel von DNS Administratoren verwaltet werden müssen, andere Administratoren jedoch die weiteren Schlüssel anlegen und verwalten, wäre der Anstieg des Administrationsaufwands immens.
Der zweite Dienst unterstützt die Authentifizierung der Herkunft der Daten. Durch die Verwendung einer digitalen Signatur kann zum einen sichergestellt werden, dass die Antworten des DNS-Servers auch tatsächlich aus dem Datensatz der zuständigen DNS-Server stammen, und zum anderen kann mit der digitalen Signatur auch die Datenintegrität gewährleistet werden.
Der dritte Dienst bezieht sich auf die Authentifizierung von Transaktionen und Anfragen. Um dies zu ermöglichen werden Transaktionssignaturen verwendet, welche an das Ende einer Serverantwort angefügt werden. Somit kann der Resolver sicher sein, dass er mit demjenigen Server kommuniziert, mit dem er auch tatsächlich kommunizieren will. Darüber hinaus wird dem Resolver versichert, dass die empfangenen Nachrichten bei der Übertragung nicht verfälscht wurden.
Innerhalb der DNSSec Erweiterungen erfahren TTL (Time To Live), CNAMES[4] (Canonical Names) und die Delegationspunkte besondere Berücksichtigung. Dies wird jedoch in Kapitel 8 noch näher erläutert werden.

4) SIG RR – Signature Resource Record

Die bereits erwähnte digitale Signatur wird in so genannten Signature Resource Records (SIG RRs) gespeichert. Diese SIG RRs dienen durch die nicht fälschbare enthaltene Signatur zur Sicherung der Datenintegrität. Zudem authentisiert ein SIG RR ein ganzes RR-Set eines bestimmten Typs, Klasse und Namens und verknüpft es mit dem Domänennamen des Signierers.
Die beinhaltete digitale Signatur enthält neben dem kryptographischen Hashwert auch Daten über den Ersteller der Signatur und das angewendetes Verfahren, sowie das Gültigkeitsintervall der Signatur. Letzteres ist notwendig, da der Schlüssel, welcher zur Erstellung der Signatur verwendet wird, mindestens alle fünf Jahre geändert werden sollte, damit die Signaturen sicher bleiben.
Die Signatur wird an eine Antwort auf eine DNS-Anfrage als zusätzliche Information angehängt.

Die Struktur der Record-Daten (RDATA) des SIG RR wird in der nachstehenden Abbildung verdeutlicht:

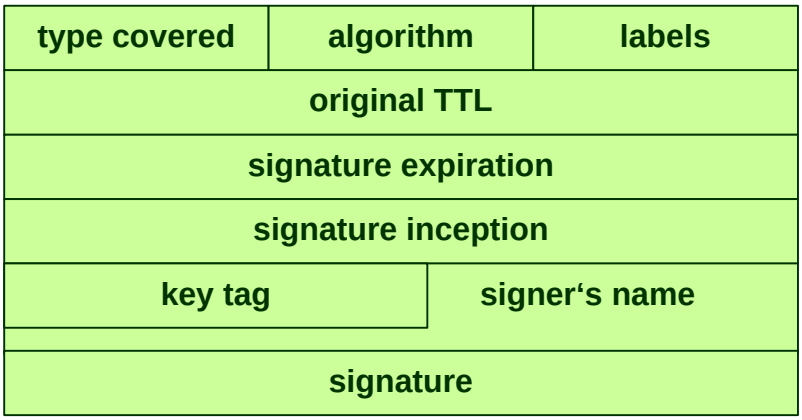

Abb.2: RDATA – Struktur eines SIG RR

Die Bedeutung der einzelnen Felder wird im Folgenden erläutert:
Das **type covered** Feld gibt den Typ des RR Sets an, welcher von der Signatur abgedeckt wird.
Im **algorithm** Feld wird vermerkt, in Verbindung mit welchem Algorithmus der Schlüssel benutzt wird.

[4] Ressource Record der den Domänen-Namen enthält

Das Feld **labels** enthält die Anzahl der Labels im ursprünglichen Besitzernamen des SIG RRs ohne „root“ und Wildcards. Der Besitzername www.gmx.de würde somit zu einer Labelanzahl von drei führen. Ist bei einer Abfrage der RR Name länger als „labels“ es angibt, kann der Resolver es als Folge einer Wildcard-Ersetzung erkennen. Ist bei einer Abfrage der Besitzernamen dagegen kürzer als es das „labels“-Feld angibt, muss der SIG RR als beschädigt gekennzeichnet und ignoriert werden.
Das **original TTL** Feld muss bei Verifizierung der Signatur wieder hergestellt werden. Dies impliziert im Allgemeinen, dass alle RRs eines bestimmten Namens, Typs und Klasse, also alle RRs eines RR Sets, den selben TTL-Startwert haben müssen.
Das Feld **signature expiration** enthält die Ablaufzeit der Signatur, während das **signature inception** Feld den Beginn der Gültigkeit der Signatur angibt.
Das Feld **key tag** dient dazu, möglichst effizient aus mehreren anwendbaren Schlüsseln auszuwählen.
Das **signer‘s name** Feld enthält den Domänennamen des Signierers der zugleich auch der Besitzername des Public Keys ist. Letzterer wird benutzt, um die Signatur zu verifizieren.
Das letzte Feld **signature** enthält schließlich die Signatur. Diese verknüpft die RDATA-Felder mit dem RR Set der in ‚type covered‘ enthaltenen RRs. Durch die Verknüpfung derjenigen RRs, welche denselben Besitzernamen und dieselbe Klasse wie der SIG RR
haben, ist dieser RR Set authentifiziert.

Um eine Transaktion zu authentifizieren, wird ein spezieller SIG RR ans Ende der Antwort eingefügt. Anders als der im vorhergehenden Text beschriebene SIG RR wird der Transaktions SIG RR nicht mittels Zone Key, sondern mit einem Server Host Key signiert.
Anhand der Verifikation des Transaktions SIG RR durch den Resolver kann sichergestellt werden, dass Anfrage und Antwort bei der Übermittlung nicht verändert wurden, die Antwort der Anfrage entspricht und die Antwort wirklich von dem Server stammt, an den die Anfrage gesendet wurde.

Um Anfragen mittels SIG RR zu signieren, werden spezielle SIG RR am Ende der Anfragen eingebunden. Dies ist jedoch nur für Update-Anfragen sinnvoll und kann bei älteren DNS-Servern zu Fehlern führen. Jedoch kann das Signieren von Anfragen für zukünftig mögliche Anfragen notwendig werden und somit öfters zum Einsatz kommen.

Des Weiteren soll - soweit dies möglich ist - für jeden authentifizierten RR Set, den die Anfrage zurückliefert, ebenfalls ein authentifizierender SIG RR gesendet werden. Hierzu existiert eine Vielzahl von Richtlinien, wann und wie SIG RR in Antworten eingebaut werden können oder sollen bzw. wie solche Antworten mit SIG RRs verarbeitet werden müssen. Diese Regeln können in dem, dieser Ausarbeitung zugrunde liegenden RFC 2535 in Kapitel 4.2 bzw. 4.3 nachgelesen werden.

Um die in dem SIG RR enthaltene Signatur zu erstellen, muss zu Beginn mittels einer Hashfunktion der Hashwert des RR Sets berechnet werden. Hierzu wird im Allgemeinen die Verwendung des DSA-Verfahrens empfohlen. Um nun die digitale Signatur zu erhalten, muss der Hashwert noch mit dem Private Key der Domain verschlüsselt werden.
Soll die Integrität einer Nachricht überprüft werden, muss zuerst der Hashwert der Nachricht (ohne die Signatur) berechnet werden. Danach wird die digitale Signatur mit dem Public Key entschlüsselt. Stimmen die beiden Werte überein, kann davon ausgegangen werden, dass die Nachricht nicht verändert wurde.
Der für die Integritätsprüfung benötigte Public Key kann durch eine DNS Abfrage ermittelt werden. Die Übermittlung erfolgt mit Hilfe eines Key Resource Records.

5) KEY RR – KEY Resource Record

Der KEY Resource Record (KEY RR) beinhaltet einen mit einem DNS-Namen verknüpften Public Key. Dies kann ein Public Key einer Zone, eines Hosts, eines Benutzers oder einer anderen abschließenden Entität sein. Der Schlüssel befindet sich entweder in der Zone, deren Signatur mit dem zugehörigen Private Key erstellt wurde, oder in der ihr übergeordneten Zone (Superzone).
Liegt der Private Key in der Superzone vor, so muss dieser ein Null Key sein, d.h. er dient nicht zur Verschlüsselung, sondern zeigt an, dass die untergeordnete Zone (Subzone) noch keine DNSSec eingeführt hat. Liegt also in der Subzone kein Schlüssel vor, muss zunächst überprüft werden, ob in

der Superzone ein Null Key vorliegt. Nur wenn dieser dort vorhanden ist, kann sicher gesagt werden, dass die Subzone unsicher ist.
Die Sicherheitserweiterungen im DNS schreiben mindestens einen Schlüssel pro Zone vor, jedoch sind auch mehrere erlaubt. Aufgrund dessen müssen sicherheitsbewusste DNS Implementierungen mindestens zwei gleichzeitig gültige Schlüssel des gleichen Typs handhaben können, welche mit dem gleichen Namen verbunden sind.
Wie schon die SIG RR werden auch die KEY RR – falls vorhanden – von sicherheits-bewussten DNS Servern an das Ende von Antworten angefügt.
Die RDATA eines KEY RR haben folgende Struktur:

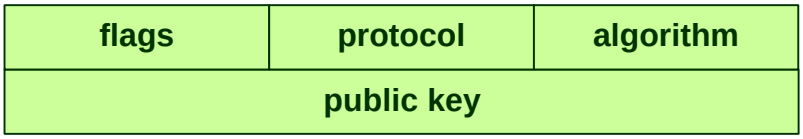

Abb.2: RDATA – Struktur eines KEY- RR

Das Feld **flags** legt fest, mit welcher Funktion der Besitzernamen und der Publik Key verknüpft ist. Da DNS Namen auf drei verschiedene Kategorien verweisen können, kann diese Funktion der Verweis auf eine Zone, der Verweis auf einen Host (oder eine andere Endentität) oder der Verweis auf die Zuordnung eines Benutzers oder eines Accounts zu einem DNS Namen sein.
Das **protocol** Feld gibt an, in Verbindung mit welchem Internet-Protokoll der Schlüssel benutzt wird.
Analog dazu gibt das **algorithm** Feld an, in Verbindung mit welchem Algorithmus der Schlüssel verwendet wird.
Das abschließende Feld **public key** beinhaltet schließlich den Public Key.

6) Zonenstatus

Für jeden verwendeten Algorithmus kann sich eine Zone in einem von drei verschiedenen Stati befinden:
Die Zone hat den Status **sicher**, wenn jeder abgerufenen RR durch einen SIG RR authentifiziert wird.
Den Status **experimentell unsicher** hat eine Zone dann, wenn SIG RRs optional vorhanden sein können. Liegen in der Zone SIG RRs vor, müssen diese überprüft werden.
Der Status der Zone ist **unsicher**, wenn keine SIG RRs benötigt werden, um RRs von einer Zone abzufragen.

Um den Zonenstatus zu bestimmen, müssen die KEY RRs der Zone überprüft werden:
Behauptet jeder glaubwürdige KEY RR der Zone, das es für diese Zone keinen Schlüssel gibt, dann ist die Zone für den überprüften Algorithmus unsicher.
Existiert für die betrachtete Zone sowohl ein KEY RR mit Schlüssel, als auch einer ohne Schlüssel, so ist die Zone als experimentell unsicher zu bewerten.
Wenn jeder vertrauenswürdige KEY RR der Zone einen Schlüssel spezifiziert, ist die Zone für diesen Algorithmus sicher. Von dieser Zone werden nur authentifizierte RRs akzeptiert.

Zur Verdeutlichung der Statusbestimmung folgt ein kurzes Beispiel:
Es wird von einem Resolver ausgegangen, der der Superzone von Z und einer dritten Partei X vertraut. Die Daten der Zone Z können von keinem, von einem (Superzone von Z oder X) oder von beiden (Superzone von Z und Partei X) signiert werden. Abhängig von den signierten KEY RR der Zone Z wird nun der Status anhand folgenden Schemas bestimmt:

X \ Superzone	k.A.	NoKey	gemischt	Schlüssel
k.A.	illegal	unsicher	experim.	sicher
NoKey	unsicher	unsicher	experim.	sicher
gemischt	experim.	experim.	experim.	sicher
Schlüssel	sicher	sicher	sicher	sicher

Abb. 3: Schema zur Bestimmung des Zonenstatus

7) NXT RR – Next Resource Record

Next Resource Records (NXT RRs) ermöglichen sowohl eine authentifizierbare Antwort auf Anfragen nach nicht existierenden Rechnernamen als auch auf Anfragen nach nicht existierenden DNS Einträgen. Somit kann zum Beispiel versichert werden, dass es in einer Zone keinen RR mit einem bestimmten Besitzernamen geben kann.
Darüber hinaus zeigen NXT RR an, welche RR Typen für einen existierenden Namen vorliegen.
Die RDATA eines NXT RR sind wie folgt strukturiert:

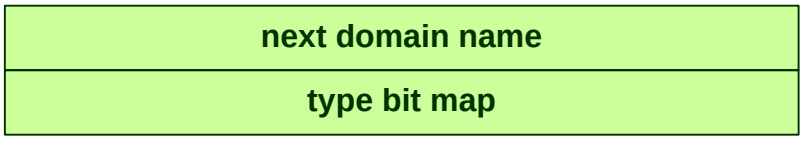

Abb. 4: RDATA – Struktur eines NXT RRs

Das Feld **next domain name** enthält den nächsten Domänennamen nach lexikographischer Ordnung.
Das **type bit map** Feld beinhaltet alle RR Typen, die für einen existierenden Namen vorliegen.

NXT RRs erstellen eine Kette aus allen Besitzernamen, d.h. ein NXT RR verweist jeweils auf den nächsten in lexikographischer Ordnung vorkommenden Eintrag. Vorraussetzung hierfür ist die Speicherung aller DNS Daten in lexikographischer Ordnung.
Die Existenz eines NXT RR deutet darauf hin, dass kein weiterer Name zwischen dem Besitzernamen des NXT RR und dem Namen in seinen RDATA existiert. Außerdem wird sichergestellt, dass kein anderer Typ unter diesem Besitzernamen existiert.
Das Problem bei einer solchen Kette aus NXT RRs ist jedoch, dass kein Name für die Record Daten des letzten NXT RRs vorhanden ist.
Dies wurde jedoch durch eine ringförmige Anordnung der NXT RRs gelöst, bei der der letzte NXT RR wieder auf den ersten verweist. Er enthält somit in den RDATA den Zonennamen.

Antworten auf Anfragen bzgl. der Nicht-Existenz eines Namens benötigen u. U. mehrere NXT RR, da auch bewiesen werden muss, dass keine Wildcard existiert, deren Erweiterung zurückgegeben werden müsste bzw. dass nicht mehr Namen (oder Wildcards) existieren, die bei der Antwort hätten berücksichtigt werden müssen.

8) TTL, CNAMEs und Delegationspunkte

Wie bereits in Kapitel 3 angesprochen, werden TTL, CNAMEs und Delegationspunkte besonders berücksichtigt.
Das TTL-Feld enthält eine Anzahl in Sekunden, die angibt, wie lange ein anderer Nameserver das abgefragte Ergebnis zwischenspeichert. Gewöhnlich beträgt diese Zeitspanne zwei Tage. Da der TTL-Wert beim Zwischenspeichern der Daten verringert werden soll, andererseits jedoch keine Änderung

der Daten zwischen Signierung und Verifizierung der Signatur erlaubt ist, ergibt sich an dieser Stelle ein gravierender Widerspruch.
Eine erste Idee diesen Widerspruch zu lösen bestand darin, das TTL-Feld außerhalb der Signatur zu halten. Jedoch wirft dieser Lösungsansatz ein anderes Problem auf, da somit Server unbemerkt willkürlich große TTL-Werte setzten könnten.
Letztendlich gelöst wurde das Problem durch die Einbindung des ursprünglichen TTL-Werts in die Signatur und die Übertragung der Daten mit einem aktuellen TTL-Wert. So können Server zwar den aktuellen TTL-Wert verändern, der sicherheitsbewusste Resolver richtet sich jedoch nach dem signierten TTL-Wert. Da Signaturen noch zusätzlich Anfangs- und Ablaufzeiten besitzen, kann ein Resolver der die Zeit kennt, auch die Gültigkeit einer Signatur erkennen.

Ein weiteres Problem ergibt sich, wenn gesicherte RRs mit dem gleichen Besitzernamen wie der des CNAME RRs durch einen ungesicherten Server abgefragt werden. Besonders eine anfängliche Abfrage eines CNAMEs oder eines anderen Typs kann keine zugeordneten SIG RRs, KEY RRs oder NXT RRs abfragen. Wenn andere Typen als CNAME abgefragt werden, werden der CNAME und der Typ des Namens (oder der Reihe von Namen), auf den der CNAME verweist, zurückgegeben. Wird eine spezielle Anfrage nach einem SIG RR gestellt, wird nicht der SIG RR des CNAME Domänennamens zurückgegeben, sondern der SIG RR des Namens, auf den der CNAME verweist.
Für sichere CNAMEs im DNS müssen sicherheitsbewusste Server verwendet werden, die folgende Anforderungen erfüllen:
Zum einen müssen sie KEY RRs, SIG RRs und NXT RRs zusammen mit CNAMEs erlauben. Weiterhin müssen sie die Verarbeitung von CNAMEs bei Abfragen von KEY RR, SIG RR, NXT RR und CNAMEs unterdrücken. Außerdem müssen sie automatisch die SIG RRs zurückgeben, die die CNAMEs authentifizieren, auf die bei der Bearbeitung der Anfrage gestossen wurde. Dies stellt eine Änderung zu RFC 1034/1035 da, da dort in Knoten in denen ein CNAME RR vorkommt, andere RR-Typen verboten waren.

Wie nebenstehende Abbildung 5 zeigt, gehören Delegationspunkte sowohl zu einer Superzone, als auch zu der zugehörigen Subzone. Sie können von beiden Zonen signiert sein und gestellte Anfrage können RRs und SIG RRs von beiden Zonen erhalten.
Gilt die Superzone als sicher, muss jede ihrer Subzonen einen von der Superzone signierten KEY RR enthalten. Dieser KEY RR kann auch in der Superzone enthalten sein.
Ist eine der Subzonen unsicher, so muss diese durch einen Schlüssel als unsicher deklariert sein. Dieser Schlüssel wird von der Superzone signiert und ist auch in dieser enthalten.
Für alle außer einem RR Typ sind die Subzonendaten maßgebend, deshalb sollte nur der KEY RR in der Superzone signiert werden (falls er dort auftritt). NS RRs (enthalten einen Name Server Eintrag) und andere glue address RRs (dienen zur Kontaktaufnahme zu Nameservern in Subzonen) sollten nur in der Superzone signiert sein. SOA (Start Of Authority Eintrag: enthält Attribute der Domäne) und andere RRs mit der Zone als Eigentümer sollten nur in der Subzone erscheinen und werden auch nur dort signiert. Die NXT RRs bilden die oben angesprochene Ausnahme. Sie erscheinen sowohl in der Super- als auch in der Subzone, wenn beide Zonen sicher sind. In diesem Fall können sowohl die Daten der Superzone als auch die Daten der Subzone maßgebend sein.

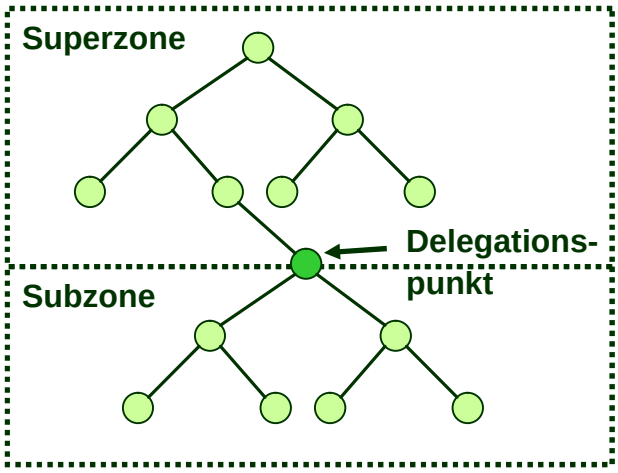

Abb. 5: Delegatinspunkt

9) Sichere Namensauflösung

Das Abfragen oder Auflösen von sicheren Daten beginnt mit einem oder mehreren statisch im Resolver konfigurierten Public Keys.
Die Daten auf den sicherheitsbewussten Servern sind in vier Kategorien aufgeteilt:

Die erste Kategorie bilden die authentifizierten Daten. Bei diesen Daten liegt jeweils eine gültige Signatur aufgrund eines Schlüssels vor, welcher mittels einer Kette von SIG RRs und KEY RRs zu einem statisch konfigurierten Schlüssel rückverfolgbar ist. Die SIG RR und KEY RR sind durch Resolverrichtlinien zugelassen.
Die zweite Kategorie beinhaltet die unbearbeiteten Daten. Diese Daten besitzen keine authentifizierte Signatur, jedoch mindestens eine zusätzliche, die der Resolver noch zu authentifizieren versucht.
Die dritte Kategorie wird von den unsicheren Daten gebildet. Diese Daten können entweder niemals authentifiziert werden, oder sie wurden in der Zone aus der sie stammen als „bad“ kategorisiert. Dies begründet sich entweder auf der Tatsache, dass sie sich die Daten in einer unsicheren Zone befinden, durch eine unsichere Zone angekommen sind, eine unsignierte glue address haben oder Name Service Daten eines Delegationspunkts sind.
Die letzte Kategorie beinhaltet die Bad Daten. Da bei diesen Daten alle Signatur-Tests fehlgeschlagen sind, werden sie vom Server gelöscht.

Die beiden vorher ungenutzten AD und CD Header Bits werden außerhalb des DNS Anfrage/Antwort Headers zugewiesen. Das AD (authentic data) Bit zeigt bei einer Antwort an, dass alle Daten in der Antwort und im Authentifizierungsbereich der Antwort vom Server nach dessen Richtlinien authentifiziert sind. Das CD (checking disabled) Bit zeigt bei einer Anfrage an, das unbearbeitete, (noch) nicht authentifizierten Daten für den Resolver, der die Abfrage sendet, akzeptabel sind.
Sicherheitsbewusste Resolver dürfen dem AD Bit nur vertrauen, wenn sie dem Server vertrauen, mit dem sie kommunizieren und entweder einen sicheren Kanal zu diesem haben oder sichere DNS Transaktionen benutzen.
Sicherheitsbewusste Resolver, die Verschlüsselung nutzen wollen, sollten auf das CD Bit in der Abfrage bestehen, um zum einen der Abfrage eigene Richtlinien aufzwingen zu können und zum um die Latenzzeit zu verringern, indem sie dem Server erlauben, mit unbearbeiteten Daten zu antworten.

Die Verkettung durch Schlüssel geschieht nach folgendem Schema:
Die RR Sets sind im Allgemeinen von einem oder mehreren SIG RR signiert. Jeder dieser SIG RR hat einen Signierer, unter dessen Name der Public Key gespeichert ist, der bei der Authentifizierung des SIG RR verwendet wird. Die einzelnen Public Keys werden von einem SIG RR signiert, welche wiederum Signierernamen besitzen, die auf einen Schlüssel verweisen u. s. w.
Eine Authentifizierung führt somit zu einer Kette mit abwechselnden SIG RRs und KEY RRs.

Die Validierung jedes SIG RRs mit Bezug zu einem Schlüssel muss einen objektiven Verschlüsselungstest bestehen. Dieser Verschlüsselungstest wird von einem Verschlüsselungsalgorithmus beinhaltet. Letztlich entscheiden Resolverrichtlinien, ob ein bestimmter SIG RR bestimmte Daten authentifizieren kann oder nicht.
Es gibt drei empfohlene Richtlinien:

- A < B (Domänenname A ist kürzer als Domänenname B; erhalten durch Weglassen von einem oder mehreren Labels auf der linken Seite von B): Dies bedeutet, dass A indirekt oder direkt eine Superdomäne von B ist.
- A = B: Dies bedeutet, das A und B die selben Domänennamen sind.
- A > B (Domänenname A ist länger als Domänenname B; erhalten durch Hinzufügen von einem oder mehreren Labels auf der linken Seite von B): Dies bedeutet das A indirekte oder direkte Subdomäne von B ist.

Sei STATIC der Besitzername eines Satzes von statisch konfigurierten, vertrauenswürdigen Schlüsseln auf einem Resolver. OWNER sei ein RR Set mit Besitzername OWNER. SIGNER ist dann ein gültiger Name eines SIG RRs, der OWNER authentifiziert, wenn folgende drei Regeln gelten:
1) OWNER > oder = SIGNER: OWNER ist in derselben Domäne oder in einer Subdomäne von B, außer wenn der Signierer gleich root ist, denn dann muss OWNER gleich root oder gleich einem Domänennamen höchster Ebene sein.
Diese Regel ist eine Regel zum Absteigen innerhalb eines DNS Baums. Sie beinhaltet ein spezielles Verbot für die root Zone, aufgrund der Beschränkung, dass die root Zone nur eine Ebene tief ist. Diese ist die wichtigste der drei Regeln.
2) (OWNER < SIGNER) und (SIGNER > oder = static): OWNER ist Superdomäne von SIGNER und SIGNER ist statisch konfiguriert oder eine Subdomäne eines statisch konfigurierten Schlüssels.
Regel 2 dient dem Aufsteigen von einem oder mehreren statisch konfigurierten Schlüsseln innerhalb des DNS Baums. Sie bewirkt nichts, wenn nur die root Zone statisch konfigurierte Schlüssel besitzt.
3) SIGNER = static: SIGNER ist genau ein statisch konfigurierter Schlüssel.

Diese Regel erlaubt unmittelbare Quer-Zertifizierung. Auch sie bewirkt nichts, wenn nur die root Zone statisch konfigurierte Schlüssel besitzt.

10) Server und Resolver Konformität

Um DNS Sicherheit verwirklichen zu können, muss natürlich auch die Server und Resolver Konformität gewährleistet werden. Es sind zwei Stufen der Serveranpassung definiert:
Innerhalb der Anpassungsstufe BASIC speichert der Server SIG RRs, KEY RRs und NXT RRs und fragt diese ab. Jedoch können u. a. sichere CNAMEs nicht unterstützt werden. Diese Stufe stellt die Mindestanforderung für untergeordnete Server und Caching-Server dar.
In der Anpassungsstufe FULL sind alle Fähigkeiten von BASIC enthalten. Darüber hinaus liest der Server SIG RRs, KEY RRs und NXT RRs in Zonendaten ein, fügt, bei gegebener Zonendatei und gegebenem Private Key, geeignete SIG und NXT RR hinzu und bindet ordnungsgemäß und automatisch SIG RRs, KEY RRs und NXT RRs in Antworten ein. Weiterhin unterdrückt er bei der Abfrage der RRs des Sicherheitstyps die Zurücklieferung von CNAMEs, erkennt CD Abfrage Header Bit und setzt AD Abfrage Header Bits angemessen. Außerdem werden die beiden NXT RRs an den Delegationspunkten richtig behandelt. Die Anpassungsstufe FULL stellt die Mindestanforderung für übergeordnete Server für sichere Zonen dar. Für vollkommen sicheren Betrieb sollten auch alle untergeordneten, zwischenspeichernden und alle anderen Server, die mit der Zone zu tun haben, der Anpassungsstufe FULL entsprechen.
Analog zur Serveranpassung, sind auch bei der Resolveranpassung zwei Stufen definiert:
Innerhalb der Stufe BASIC bearbeitet der Resolver SIG RRs, KEY RRs und NXT RRs, wenn sie explizit abgefragt werden.
In der Anpassungsstufe FULL versteht der Resolver KEY RRs, SIG RRs und NXT RRs einschließlich der Verifizierung deren Signaturen für den vorgeschriebenen Algorithmus. Der Resolver speichert die Informationen darüber im Cache und in der Datenbank, welche RRs für welche Erweiterung authentifiziert wurden. Außerdem führt er zusätzliche Abfragen durch, falls diese gebraucht werden um SIG RRs, KEY RRs oder NXT RRs zu erhalten. Darüber hinaus setzten Resolver der Stufe FULL gewöhnlich das CD Abfrage Header Bit bei ihren Abfragen.

11) Probleme von DNSSec

Ein Problem der DNS Sicherheit liegt darin, dass es die Effizienz des DNS stark negativ beeinflusst. Ein Aspekt ist der erhebliche Anstieg der notwendigen Datenbankgröße. Ein anderer Aspekt ist die erhebliche Verlangsamung der Namensauflösung, welche durch das Signieren der Zonen und das Entschlüsseln der Signaturen begründet ist.
Ein weiters Problem der DNS Sicherheit ist darin zu sehen, dass ein Fehler in der Verteilung der Root-Keys verheerende Folgen nach sich zieht. Davon abgesehen, dass ein solcher Fehler die gesamten Sicherheitserweiterungen des DNS nutzlos machen würde, könnten alle Resolver die DNSSec benutzen keine Namensauflösung mehr durchführen.
Diese schwerwiegenden Probleme konnte bis heute noch nicht gelöst werden.

12) Anhang

Anhang A: DNS-Resolver

Jeder Netzwerk-Client, der auf einen Internet-Host zugreifen möchte, aber nur dessen Domain-Adresse kennt, muss wie bereits erwähnt einen Nameserver konsultieren. Dies geschieht mittels eines so genannten Resolvers. Dabei existiert der Resolver nicht als eigenständiges Programm. Vielmehr besteht er aus einer Bibliothek von Software-Routinen, dem Resolver-Code, die zu jedem Programm gelinkt wird, das Adressen nachschlagen muss. Diese Bibliothek weiß, wie Anfragen über Rechner an den Nameserver formuliert werden.
Der Resolver übernimmt im Wesentlichen folgende drei Aufgaben:

- Er erzeugt die Abfrage und übermittelt sie an den Nameserver. In der Regel ist dies der Nameserver des Internet-Providers, über den man ins Internet geht.
- Im Falle einer erfolgreichen Namensauflösung interpretiert er die Antwort des Nameservers.
- Anschließend übergibt der Resolver die Informationen an das Programm, das die Daten angefordert hat, beispielsweise an den Webbrowser.

(Quelle: http://www.tecchannel.de/internet/205/6.html)

Anhang B: Ressource Records

Das Domain Name System speichert die Informationen über Hosts in so genannten Resource Records (RRs). Es gibt bis zu 20 verschiedene Typen.
Wichtige Resource Records:

A	Definiert eine IP-Adresse
CNAME	Canonical Name: Beinhaltet den Domainnamen
HINFO	Host Information: Zwei Strings spezifizieren die CPU und das Betriebssystem. Diese Funktion wird kaum unterstützt.
NS	Name Server: Definiert den autoritativen Nameserver für eine Domain
PTR	Pointer: Wird fast ausschließlich mit der .in-addr.arpa-Domain verwendet, um Reverse Lookups zu ermöglichen, d.h., man sucht den Domainnamen einer bekannten IP-Adresse.

Die Resource Records werden bei einer Anfrage an den Client übermittelt. Dazu werden diese an den DNS-Header angehängt. Für jeden DNS-Eintrag existiert ein Resource Record, der bei einer Anfrage an einen Nameserver als Antwort übermittelt wird.
Der RR einer DNS-Antwort enthält folgende sechs Felder:

Domain Name	Enthält den Domainnamen, der aufgelöst werden soll
Type	Spezifiziert den RR-Typ
Class	In diesem Feld steht in der Regel 1 für Internet-Daten
Time To Live	Enthält die Anzahl in Sekunden, wie lange ein anderer Nameserver das Ergebnis zwischenspeichert; meistens sind dies zwei Tage
Resource Data Length	Gibt die Länge des Feldes "Resource Data" an
Resource Data	Enthält die IP-Adresse

(Quelle: http://www.tecchannel.de/internet/205/8.html)

Anhang C: Zonen und DNS Sicherheit

Eine Zone ist als eine Dateneinheit zu verstehen, die vollständig unter der Kontrolle des Zoneneigentümers steht. Jeder RR Set ist durch einen speziellen Private Key signiert, dessen Besitzer der Zonenmanager ist. Der Public Key, der dazu dient eine Zone zu authentifizieren, sollte vor dem Laden der Zone in den Hauptserver in lokalen Konfigurationsdateien definiert werden, so dass die Zone authentifiziert werden kann.