Wie DNSSEC den Name Service im Internet absichert

Vertraut

Seit einigen Monaten unterstützen alle Root-Server im Domain Name System die Signierung per DNSSEC. Dieser Artikel verrät, was es bringt und wie man es Client-seitig einrichtet. Kurt Seifried

In der Vergangenheit war der Domain Name Service (DNS) immer wieder das Ziel von Angriffen. Der Grund dafür ist, dass das DNS-Proktoll einige inhärente und nicht leicht behebbare Schwächen aufweist, zum Beispiel kurze Transaktions-IDs. Mit Hilfe dieser DNS-Probleme können Angreifer gefälschte Daten in den DNS einschleusen (Cache Poisoning). Mit WLAN verschärft sich das Problem noch einmal: Ein Hacker kann einfach eine DNS-Anfrage abfangen und sie mit gefälschten Daten beantworten, bevor der echte Server es tut.

Hier kommt DNSSEC ins Spiel. Es soll sicherstellen, dass die DNS-Dateien echt sind und niemand sie manipuliert hat. Dafür verwendet DNSSEC klassische Public/Private-Key-Kryptographie, ähnlich wie GPG/PGP: Wer im Besitz des öffentlichen Schlüssels ist, kann die Echtheit von Daten verifizieren, die mit dem privaten Schlüssel signiert wurden. Besonders gut funktioniert das mit DNS, denn es besitzt von Haus aus eine Struktur wie eine Chain of Trust, an deren oberem Ende die Root-Server sitzen (Abbildung 1).

Dieser Artikel beschränkt sich auf eine nähere Betrachtung der Client-Seite, denn für mehr Sicherheit ist es wichtiger, die von anderen gelieferten DNS-Informationen zu überprüfen, als eigene DNS-Einträge zu signieren. Wie viele andere Dinge ist auch DNSSEC in der Theorie einfach, aber in der Praxis ein bisschen komplizierter.

Baumstruktur

Bei einem DNS-Request für »beispiel.org« sieht der befragte Server zuerst in der Zone-Datei für die Wurzel ».« nach, welche Adressen die Root-Server haben. Dann wird er eine Anfrage an einen Root-Server richten, um den Nameserver für die Top Level Domain ».org« zu erfahren. Ist nun das entsprechende Bit »DNSSEC OK« (D0) im »OPT«-Teil der Anfrage gesetzt, sollte der Server mit signierten Daten antworten. Nun muss Ihr Server die Echtheit der empfangenen Antwort verifizieren. Stellt sich also die Frage, wie Sie an den Public Key für die Root-Domain gelangen.

Ähnlich wie bei der Root-Zone werden die meisten DNS-Server bei der Installation mit den Root-Schlüsseln ausgestattet, zum Beispiel bei der Installation des Betriebssystems. So wird das Henne-Ei-Problem vermieden, auf einem sicheren Weg an die Schlüssel zu gelangen.

Mit dem Root-Public-Key prüft Ihr Server, ob die empfangenen Daten authentisch sind. Nun kann er den Domain-Server für »beispiel.org« kontaktieren und sich dabei sicher sein, die richtige Adresse zu verwenden. Dieser liefert etwa eine signierte Antwort für die Adresse von »www.beispiel.org« aus. Das bedeutet, Ihr Server braucht nun auch dessen Public Key, um diese Antwort wieder verifizieren zu können. Auch den bekommen Sie vom DNS-Server für ».org«. Die Vertrauenskette endet also letztlich immer wieder bei den Root-Servern, deren öffentliche Schlüssel Sie auf eine sichere Weise erhalten müssen.

Wegen seines Namens glauben viele Anwender, DNSSEC würde den DNS-Server völlig sicher machen. Leider ist das aber nicht der Fall. Es stellt nur die Authen-

tizität der Antworten sicher, nicht mehr und nicht weniger. Zum Beispiel können Angreifer über ansonsten ungesicherte Leitungen immer noch DNS-Verkehr mitschneiden, denn er ist schließlich nur signiert, aber nicht verschlüsselt. Natürlich kann DNSSEC auch nicht verhindern, dass ein Angreifer für Phishing-Attacken Domains registriert, die jenen seiner Angriffsziele gleichen.

Anwendungen

DNSSEC macht es möglich, andere Services mit Authentifizierung und Verschlüsselung zu versorgen. So lassen sich prinzipiell schon lange PKIX-Zertifikate, PGP-Schlüssel und andere in Cert-Records speichern und übertragen. Ebenso können IPseckey-Records IPsec-Public-Keys speichern. Ohne DNSSEC war das Risiko allerdings groß, dass ein Angreifer diese Daten unterwegs manipuliert.

Jetzt ist zum Beispiel das Management von SSH-Server-Keys möglich, indem Sie die öffentlichen Schlüssel im DNS in SSHFP-Records verpacken und in der Datei »ssh_config« die Option »VerifyHostKeyDNS« aktivieren. Zudem verhindert DNSSEC die Unart mancher Provider, bei fehlgeschlagenen DNS-Anfragen dennoch eine Antwort zu schicken und den Anwender so auf das eigene Webportal umzuleiten. Stattdessen kann eine Domain eine authentifizierte Antwort mit dem Inhalt schicken, dass es keinen passenden Eintrag gibt.

Damit ein Provider DNSSEC unterstützen kann, muss er entsprechend große Antwortpakete erlauben: Weil DNSSEC sowohl signierte Daten als auch Schlüssel transportieren muss, kann eine DNS-Antwort leicht mehrere KByte umfassen. Viele Server oder schlecht konfigurierte Firewalls erlauben nur kleinere Pakete. Wenn die DNSSEC-Daten abgeschnitten werden, sind sie aber nutzlos. Das Ripe [2] hatte einen Dienst eingerichtet, mit dem sich die maximale Paketgröße überprüfen ließ, doch inzwischen wieder eingestellt. OARC betreibt ihn aber weiter. Verwenden Sie zum Test das Dig-Tool:

```
dig +short rs.dns-oarc.net ➥
@DNS-Server txt
```

Wenn die Antwort etwa so ähnlich wie die folgende aussieht, werden Sie Probleme bei der Verwendung von DNSSEC bekommen:

```
rst.x476.rs.dns-oarc.net.
rst.x905.x476.rs.dns-oarc.net.
rst.x1086.x905.x476.rs.dns-oarc.net.
"62.245.157.194 sent EDNS buffer size 4096"
"62.245.157.194 DNS reply size limit is at ➥
least 1086"
```

Mit einer Antwort wie der folgenden können Sie DNSSEC in Betrieb nehmen:

```
...
"213.235.205.2 sent EDNS buffer size 4096"
"213.235.205.2 DNS reply size limit is at ➥
least 3843"
...
```

Wer seinen eigenen DNS-Resolver unter Linux betreiben möchte, muss zuerst ein Paket wie »caching-nameserver« oder »bind« installieren. Neuere Distributionen wie Fedora 13 unterstützen DNSSEC von Haus aus. In der Konfigurationsdatei »named.conf« müssen die beiden folgenden Optionen aktiviert sein:

```
dnssec-enable yes;
dnssec-validation yes;
```

Zusätzlich brauchen Sie auch den Root-Key, der in etwa so aussieht:

```
trusted-keys {
"." 257 3 5 "AwEDHFA234...";
};
```

Die Konfigurationsoption »dnssec-validation yes« sorgt dafür, dass der neu installierte Nameserver DNS-Signaturen überprüft. Wenn die Verifizierung fehlschlägt, antwortet der Server mit »SERVFAIL«. Das verhindert, dass ein Client sich mit einer potenziell gefährlichen Website verbindet.

Wenn DNSSEC eingeschaltet ist, können Sie es wiederum mit Dig testen. Dazu bietet das Tool den Schalter »+dnssec«:

```
dig +dnssec beispiel.org
```

Wenn Sie sich die daraufhin folgende Ausgabe ansehen, achten Sie auf das Flag »ad«. Ist es vorhanden, haben sie authentifizierte DNS-Daten erhalten.

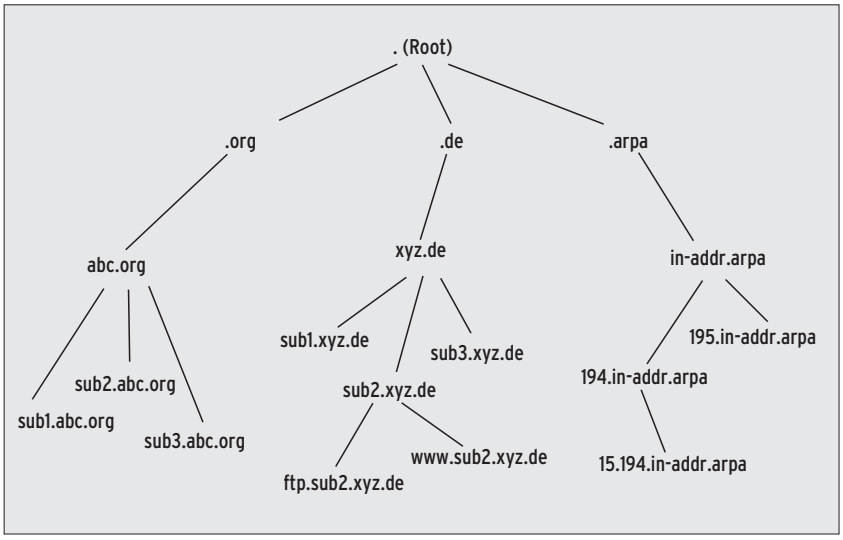

Abbildung 1: Der hierarchische Aufbau des Domain Name System erleichtert die Implementierung von DNSSEC.

Die Zukunft

Nachdem nun alle Root-Server DNSSEC anbieten, können Sie das »DNSSEC OK«-Bit in Clients und Servern einschalten. Für spezifische Domains, die Sie auf jeden Fall verifizieren wollen, können Sie auch »dnssec-must-be-secure« einschalten. Derzeit ist das die beste Lösung, denn es gibt noch sehr viele Domains, die kein DNSSEC unterstützen. *(ofr)* ■

Infos

[1] Angriff auf das DNS-Protokoll: [http://www.net-security.org/dl/articles/Attacking_the_DNS_Protocol.pdf]
[2] Ripe: [http://www.ripe.net]
[3] OARC-Test: [https://www.dns-oarc.net/oarc/services/replysizetest]

Der Autor

Kurt Seifried hat sich als Security-Consultat auf Linux und Netzwerke spezialisiert. Er fragt sich oft, wieso Technologie im Großen funktioniert, aber so häufig im Kleinen versagt.