DNS Privacy

Patrick Werneck
Betreuer: Johannes Naab
Seminar Innovative Internettechnologien und Mobilkommunikation, SS2014
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
werneck@in.tum.de

KURZFASSUNG

Im Domain Name System (DNS) existieren sowohl auf Seiten der Clients als auch auf Seiten der Betreiber erhebliche Sicherheitslücken im Bezug auf die Vertraulichkeit und Privatheit der jeweils eigenen Daten. Der Informationsfluss von Clients, die einen Domain Name auflösen wollen, findet unverschlüsselt statt und wird meist durch mehrere Server geleitet. Serverbetreiber und Angreifer, die die Nachrichten zwischen Client und primären DNS-Server mitschneiden können, haben so Zugriff auf Informationen zu den verwendeten Domain Names, sowie deren Häufigkeit. Auf Seiten der DNS Serverbetreiber öffnet die Verwendung der standardisierten DNS Sicherheitserweiterung DNSSec Angreifern die Möglichkeit die komplette Zone des Betreibers auszulesen. Mit Verschlüsselung der DNS Nachrichten können nur die auf dem Übertragungsweg auftretenden Probleme behoben werden. Weitere Lösungsansätze, die einen Umbau des Domain Name Systems erfordern, greifen auch diese Probleme auf, sind jedoch aufgrund ihrer schlechten Skalierbarkeit nicht realistisch einsetzbar.

Schlüsselworte

dns, privacy, DNSCurve, DNSCrypt, cPIR, PPDNS, Range-Query, QNAME Minimisation

1. EINLEITUNG

Das Domain Name System (DNS) ist eine hierarchische verteilte Datenbank die verschiedene Schlüssel-Wert-Paare speichern und zur Verfügung stellt. Unter anderem werden menschlich lesbare und einprägsame Namen in IP-Adressen umgewandelt, die für das Versenden und Routen von Paketen durch das Internet benötigt werden. Dadurch ist das DNS zu einem integralen Bestandteil des Internets geworden. Nach Betrachtung des Protokollverlaufs, der in Abschnitt 2 beschrieben wird, ist leicht zu erkennen, dass das ursprüngliche DNS keinerlei Sicherheitsmechanismen besitzt. Dadurch kann jede DNS-Nachricht mitgelesen und bei aktiven Angreifern auch beliebig verändert werden. Während die Probleme der Datenintegrität und Authentizität bereits durch die Erweiterung Domain Name System Security Extensions (DNSSEC) behandelt werden, wird der Aspekt der Privatsphäre und Vertraulichkeit der Nachrichten bisher nicht beachtet. Diese Probleme entstehen jedoch nicht nur durch die Möglichkeit des Angreifers Nachrichten mitzulesen, sondern auch durch die Nameserver selbst, da auch diese alle DNS-Nachrichten sehen können. So ist es möglich sowohl die angefragte Domain als auch die Häufigkeit dieser Anfragen zu bestimmen. Ersteres ermöglicht zusätzlich zur reinen Information eine Zensur der erreichbaren Domainnamen durch den Betreiber des Nameservers für seine Nutzerbasis. Letzteres kann für die Überwachung des Netzes auf neue sich schnell verbreitende Seiten verwendet werden. Diese können dann analysiert und dem Ergebnis nach blockiert werden. Es kann außerdem eine Statistik für einzelne Nutzer angelegt werden, durch die Rückschlüsse auf die Lebensweise gezogen werden kann.

In diesem Artikel werden verschiedene Lösungsansätze für die Datenschutzprobleme aufgezeigt. Dafür wird zuerst in Abschnitt 2 das DNS Protokoll analysiert, woraufhin die einzelnen Angriffsmöglichkeiten auf die Privatsphäre aus Sicht der Endnutzer und Zonenbetreiber in Abschnitt 3 vorgestellt werden. Abschnitt 4 führt darauf aufbauend einzelne Lösungsansätze für einzelne Teilprobleme auf, die in Abschnitt 5 verglichen werden und ein Ausblick für vielversprechende Kandidaten gegeben wird.

2. DNS ÜBERBLICK

Der Namensraum des Domain Name System ist in einer Baumstruktur gegliedert. Jeder Knoten in diesem Baum stellt eine Domain dar und besitzt bis auf die Wurzel ein nichtleeres Label. Der zu jedem Knoten zugehörige Fully Qualified Domain Name (FQDN) bildet sich durch Aneinanderreihung aller Label der Knoten vom Blatt bis zur Wurzel mit einem Punkt („.“) als Separatorzeichen. Knoten auf der ersten Ebene unter dem Wurzelknoten werden als Top-Level Domains (TLD) bezeichnet, alle folgenden Ebenen entsprechend ihrer Ebenentiefe (Second-Level Domain, Third-Level Domain, usw.). Der in Abbildung 1 benutzte Domainname *www.example.com* ist also der Knoten *www.example.com* mit dem Label *www* und dem Elternknoten *example.com* (Second-Level Domain), welcher *com* als Elternknoten hat (TLD). Der Knoten *com* stammt wiederum von *root*, der Wurzel des Domain Name Systems, ab.

Innerhalb des Domain Name System Baums bilden einzelne Teilbäume eine Zone. Für jede Zone ist dabei mindestens ein autoritativer Nameserver verantwortlich um eingehende DNS Anfragen zu beantworten und die Zone zu verwalten. Die Informationen zu den einzelnen Knoten der Zone werden dabei in Resource Records (RR) abgespeichert. Jedes RR besteht aus einen Typ, den dazugehörig entsprechend kodierten Daten, ein Namensattribut, welches den zugehörigen Domainnamen des RR bestimmt und eine TTL, die die Gültigkeit des RR bestimmt, enthalten.Mit dem RR Typ A wird ein Domainnamen auf eine IPv4-Adresse abgebil-

doi: 10.2313/NET-2014-08-1_18

det. Alle zu einer gegebenen Anfrage passenden Resource Records werden zusammengefasst RRset genannt. Um ein gewisses Maß an Ausfallsicherheit zu erreichen können mehrere autoritative Nameserver betrieben werden. Die Zoneninformationen werden dabei von einem Master-Server auf Slave-Server mit Hilfe der Zonentransferprotokollen IXFR und AXFR, die hier nicht weiter behandelt werden, verteilt.

Neben den autoritativen Nameserver gibt es noch die rekursiven Nameserver. Im Gegensatz zu den autoritativen Nameservern verwalten diese keine eigene Zone, sondern bieten Clients, die DNS Anfragen einleiten (Stub Resolver), eine rekursive Namensauflösung an. Das heißt der rekursive Nameserver vollzieht vom Wurzelknoten, bzw. den Root-Servern an bis zum vom Client angefragten Blattknoten alle nötigen Anfragen. Um die Last der autoritativen Nameserver zu reduzieren und Zeit zu sparen, werden die erhaltenen Antworten zwischengespeichert, und erst mit Ablauf der in den RR enthaltenen TTL wieder verworfen. Dadurch muss nicht für jede Anfrage erneut der gesamte Baum durchlaufen werden, und der Client erhält ohne erhöhte Latenz die angefragte Antwort. Eine Anfrage besteht dabei jeweils aus einem Domainnamen (QNAME) und einem Anfragetype des angefragten Domainnamens. Der Anfragetype ist bis auf wenige Ausnahmen der Typ des RR welches als Antwort zurückgegeben werden soll.

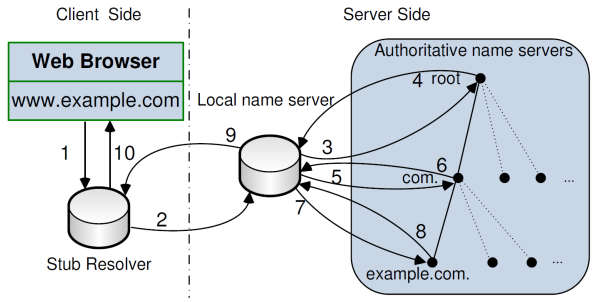

Abbildung 1: Rekursive Namensauflösung der Domain *www.example.com* [12]

Zur Veranschaulichung des rekursiven Namensauflösungsprozesses werden im Folgenden alle benötigten Einzelschritte von der Clientanfrage ausgehend anhand von Abbildung 1 beschrieben. (1) Eine Applikation - in diesem Fall ein Browser - will eine Verbindung zu *www.example.com* aufbauen, und fragt dafür die IP beim Stub Resolver, dem lokalen Dienst des Betriebssystems, ab. Dieser schickt, da Stub Resolver im Allgemeinen über keinen Cache verfügen, in (2) eine DNS Anfrage mit der Frage „Wie lautet die IP von *www.example.com*“ an den konfigurierten rekursiven Nameserver. Sofern dieser die Antwort noch nicht im Cache liegen hat und auch keine anderen Nameserver kennt, die für diese Domain zuständig sind, leitet der rekursive Resolver in (3) die Anfrage des Clients an einen Root-Nameserver weiter. Dieser Antwortet in (4) mit den Domain-Namen und IPs der zuständigen autoritativen Nameserver der *com* Zone. (5) Der rekursive Resolver leitet nun die Anfrage des Clients an den autoritativen Nameserver der *com* Zone weiter, welcher in (6) den rekursiven Nameserver an den autoritativen Server der *example.com* Zone delegiert. Nach der erneuten Weiterleitung der Clientanfrage in (7) antwortet der autoritative Nameserver für die *example.com* Zone in (8) mit der IP Adresse des Hosts *www.example.com*. Da der rekursive Resolver nun die Adresse kennt, schickt er in (9) die Antwort „*www.example.com* hat die Adresse *192.0.2.42*“ an den Stub-Resolver, welcher wiederum in (10) der Applikation die IP-Adresse liefern kann und sich diese mit dem Host *www.example.com* verbindet.

Da das gesamte DNS Protokoll nur Klartextnachrichten versendet, bietet es zunächst keine Mechanismen zur Sicherung von Integrität, Authentizität und Vertraulichkeit. Die Erweiterung DNSSEC setzt dabei die ersten beiden Ziele um, verzichtet aber ausdrücklich auf den Aspekt der Vertraulichkeit, da diese nur öffentlich zugängliche Informationen enthält. DNSSEC soll die gute Skalierbarkeit des DNS Systems beibehalten, indem alle nötigen kryptographischen Operationen vorberechnet werden. So wird mit Hilfe von Public Key Kryptographie für jedes ursprüngliche RRSet in der Zonendatei ein zusätzliches RR vorberechnet, welches die Signatur des originalen RRSet enthält. Es werden außerdem die zusätzlichen RR NSEC beziehungsweise NSEC3 eingeführt, um die nicht Existenz eines Domain Namens beweisen zu können. Bei NSEC werden lexikographisch geordnet Paare von Domain Namen gebildet, so dass ein Ring gebildet wird. Sollte ein Name angefragt werden, der nicht existiert, so wird mit dem NSEC RR geantwortet, welches das Paar enthält zwischen den der angefragte Name liegt, und damit die nicht Existenz des Domain Namens sichergestellt ohne zur Laufzeit Antworten signieren zu müssen.

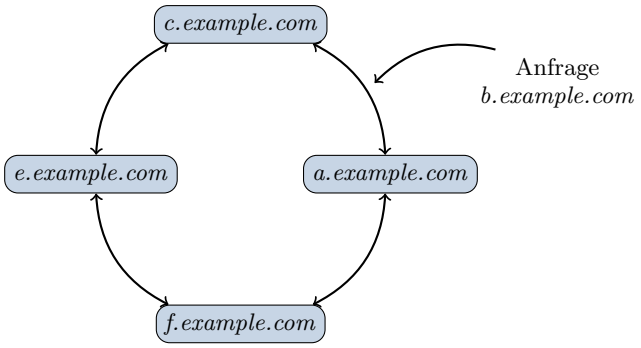

Abbildung 2: Anfrage einer nicht existenten Domain auf eine NSEC Ringstruktur mit vier Domainnamen

In Abbildung 2 ist eine solche NSEC Struktur mit vier Domainnamen gebildet. Bei der Anfrage der nicht existenten Domain *b.example.com* wird das signierte NSEC RR mit dem Paar *a.example.com* und *c.example.com* als Antwort gesendet und damit die Nichtexistenz des angefragten Namens sichergestellt. Bei NSEC3 werden anstatt der Klartext Hostnamen gehashte Domainnamen zur Bildung von sortierten Paaren verwendet, um den in Abschnitt 3.2 beschriebenen Angriff zu erschweren.

3. ANGRIFFSANALYSE

Aufbauend auf der vorhergehenden Analyse des DNS Protokolls und der Erweiterung DNSSec werden im folgenden die verschiedenen Angriffsmöglichkeiten auf die Privatheit der DNS-Daten sowohl der Sicht des Endnutzer als auch aus der Sicht des Zonenbetreibers analysiert.

doi: 10.2313/NET-2014-08-1_18

3.1 Endnutzer

Wird eine DNS-Anfrage an einen Nameserver verschickt, ist dies mit dem Ziel verbunden, sich mit dem entsprechenden Dienst zu verbinden. Aus Sicht der Endnutzer ist nicht der Inhalt der Antwort an sich die schützenswerte Information, sondern die Tatsache, das dieser einen bestimmten Dienst nutzen will. So ist beispielsweise nicht die öffentlich verfügbare Information, dass die Seite *foo.example.com* die die IP 192.0.2.42 hat schützenswert, sondern das der Nutzer diese Seite besuchen will. Auch kann eine DNS Anfrage Informationen über die verwendete Software geben, indem nach einem softwarespezifischen Serverrecord einer Domain wie *_bittorrent-tracker.* gesucht wird.

3.1.1 Auf dem Weg zu Ziel

Wie aus dem bereits in Kapitel 2 erläutert wurde, werden alle Nachrichten unverschlüsselt ausgetauscht. Dadurch können sie sowohl auf dem Weg vom Stub Resolver zum konfigurierten rekursiven Nameserver als auch von diesem zu den jeweiligen autoritativen Servern mitgeschnitten und gelesen werden. Da typischerweise Stub Resolver über keinen Cache verfügen, werden zwischen diesem und dem rekursiven Resolver jegliche DNS Anfragen von Anwendungen auf dem Client an den Server versendet. Ein Angreifer zwischen diesen beiden Stellen kann so ein komplettes Verlaufsbild des Clients, wie häufig und zu welcher Zeit verschiedene Dienste genutzt werden, erstellen. Im Gegensatz dazu sind die Möglichkeiten der DNS-Traffic-Analyse bei der Übertragung von rekursiven zu autoritativen Nameserver eingeschränkt, da das tatsächliche Anfragevolumen durch Caching im rekursiven Resolver und Anfragen anderer Clients nicht genau ermittelt werden kann. Zusätzlich verdeckt der Nameserver die Ursprungsadresse der Anfrage.

3.1.2 Durch Serverbetreiber

Zusätzlich zu der Möglichkeit Daten auf dem Weg zum jeweiligen Server mitzuschneiden können auch die Serverbetreiber selbst alle DNS Anfragen aufzeichnen und verwerten. Dabei kann der Betreiber eines rekursiven Nameserver analog zu Abschnitt 3.1.1 erneut den kompletten DNS Verlauf einzelner Nutzer auswerten. Auch Betreiber autoritativer Nameserver können erhalten den vollständigen Anfragenamen. Dadurch kann das durch Caching beschränkte Anfragevolumen bestimmt werden. Durch die statistische Auswertung der Lebensdauer der gegebenen Antworten und dem Anfrageintervall lässt sich jedoch das vollständige Anfragevolumen des rekursiven Nameservers, nicht jedoch des einzelnen Stub Resolvers, abschätzen [12].

Im Gegensatz zum Mitschneiden der DNS-Pakete auf dem Weg zu den jeweiligen Nameservern kann diese Schwachstelle nicht durch Schützen des Kommunikationskanals erreicht werden, da dieses Problem innerhalb der DNS Anfragebearbeitung auftritt.

3.2 Zonenbetreibers

Die Datenschutzaspekte der Zonenbetreiber beruhen darauf, dass der vollständige Zoneninhalt nicht öffentlich verfügbar sein soll, da dieser die komplette Netzwerkinfrastruktur auflistet. Angreifer, die den Inhalt der Zone kennen, können dadurch die Existenz wichtiger Ziele ausmachen. Ohne die Informationen aus der Zone kann es bedeutend schwieriger sein an diese Informationen zu gelangen, da so erst der korrekte Name des Hosts oder direkt die IP gefunden werden muss. So besteht zwar bei Einsatz von DNS ohne die Erweiterung DNSSEC zwar prinzipiell die Möglichkeit durch alle möglichen Namen zu iterieren, dies ist jedoch aufgrund der beschränkten Übertragungsrate nicht praktikabel. Außerdem erzeugt ein solcher Angriff eine große Datenflut, auf die der Zonenbetreiber entsprechend reagieren kann.

Durch die Einführung der Erweiterung DNSSEC wird, wie in Abschnitt 2 erläutert, mit dem neuen RR NSEC eine Klartext Ringstruktur aller vorhandenen Domainnamen gebildet. Wird nun ein Domainname abgefragt, der nicht existent ist, werden als Antwort zwei existente Domain Namen geliefert. Um die komplette Zone zu erhalten werden solange DNS-Anfragen mit Namen, die zwischen zwei existenten Domainnamen liegen, die noch nicht in der erhaltenen Ringstruktur verbunden sind, versendet, bis der komplette Ring empfangen wurde. Durch diese Methode werden nur so viele DNS-Anfragen benötigt wie in der Zone Einträge vorhanden sind [9].

Auch nachdem durch RFC5155 [11] das NSEC3 RR als Alternative für das NSEC RR eingeführt wurde, und die Domainnamen nicht mehr in Klartext, sondern in gehashter Form eine Ringstruktur bilden, kann auch hier ein analoger Angriff stattfinden. Da in diesem Fall jedoch statt der Klartextnamen nur die Hashwerte der existenten Domainnamen verfügbar sind, müssen diese noch lokal per Bruteforce erraten werden. Im Vergleich zu DNS ohne DNSSec wird so der limitierende Faktor des Bruteforceprozess von der Netzwerkgeschwindigkeit auf die des Prozessors verschoben, und dadurch erheblich beschleunigt. Zusätzlich werden nur noch wenige DNS-Anfragen benötigt und der Angriff dadurch für den Zonenbetreiber schwerer erkennbar gemacht [3].

4. LÖSUNGSMÖGLICHKEITEN

Nach der Analyse der verschiedenen Angriffsmöglichkeiten werden in diesem Abschnitt nun verschiedene Lösungsmöglichkeiten für einzelne Teilprobleme vorgestellt. Diese wurden dabei unabhängig voneinander entwickelt und können zum Teil miteinander kombiniert werden, um mehrere Teilprobleme aufzugreifen.

4.1 Verschlüsselung

In diesem Unterabschnitt werden verschiedene Lösungsansätze die Verschlüsselung der übertragenen Nachrichten oder Resource Records als Lösung vorschlagen. Verschlüsselungslösungen sind dabei für den Endnutzer nur in der Lage die in Abschnitt 3.1.1 beschriebenen Probleme durch Mitschnitt der Daten auf dem Übertragungsweg zu lösen, da die eigentlich DNS-Anfragen weiterhin durch den Nameserver verarbeitet werden müssen.

4.1.1 t-DNS: DNS über TCP mit TLS

Hu et al. [8] schlagen vor DNS-Nachrichten über TCP mit TLS verschlüsselt zu übertragen. Bereits in der Spezifikation von DNS in RFC1034 [14] und RFC1035 [13] wird eine TCP Implementierung vorgeschrieben. Aufbauend auf dieser Voraussetzung wird zum initialisieren der Verbindung eine normale DNS-Anfrage über TCP mit gesetztem TO-flag und einem beliebigen nicht aussagekräftigen Namen als Domain

gesendet. Antwortet der Server mit einem TXT Record und RCODE=0, kann mit dem TLS-Sitzungsaufbau begonnen werden. Nachdem die TLS-Sitzung eingerichtet wurde, können DNS Anfragen in normalem Format gesendet werden.

Mit Einsatz von TCP und TLS werden pro Verbindungsaufbau zusätzliche vier Round Trip Times (RTT), Zeit, die eine Nachricht vom Client zum Server und wieder zurück, benötigt. Eine für den TCP Handshake, eine RTT für das Signalisieren der Unterstützung von TLS und weitere zwei RTT für den TLS-Sitzungsaufbau gebraucht. Um diesen erhöhten Aufwand und die Latenz zu minimieren, müssen aufgebaute Verbindungen für weitere DNS-Anfragen an den selben Server wiederverwendet werden. Um zusätzliche Wartezeit bei mehreren Anfragen zu verhindern müssen Clients diese direkt aufeinander folgend an den Server schicken und parallel vom Server bearbeitet werden. In Abbildung 3 werden

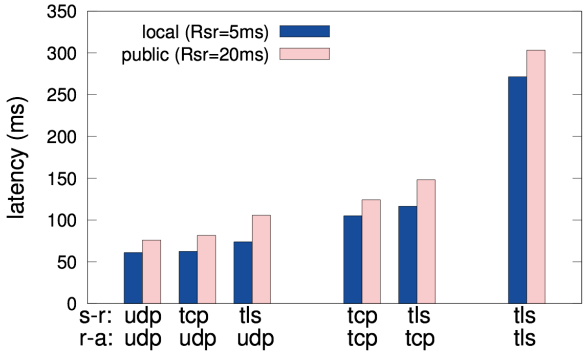

Abbildung 3: Durschnittliche Antwortzeiten von Stub Resolver zu rekursiven Nameserver des ISP und einem alternativen globalen mit verschiedenen Verbindungsarten zwischen Stub, rekursiv und autoritativen Resolvern [19]

verschiedene Verbindungsarten zwischen stub und rekursiven Resolver (s-r) und rekursiven Resolver und authoritativen Nameserver (r-a) verglichen. Dabei wird jeweils ein lokaler rekursiver Resolver des ISPs mit einer durchschnittlichen Antwortzeit (R_{sr}) von *5ms* und ein globaler mit einer durchschnittlichen Antwortzeit von *20ms* verwendet. Es ist gut ersichtlich, dass sich die zusätzlichen RTT die zur Erstellung einer TCP-TLS-Sitzung benötigt werden im Falle einer häufigen Wiederverwendung der Verbindung amortisieren. So steigt bei Verwendung von TCP-TLS anstatt von UDP als Verbindungsart von Stub zu rekursiven Resolver die Antwortzeit nur von durchschnittlich 62 ms auf 74 ms, da die Verbindung des Stub Resolvers für jede DNS Abfrage benutzt wird. Im Gegensatz dazu steigt bei zusätzlicher Verwendung von TCP-TLS für die Verbindung zwischen rekursiven und autoritativen Nameserver die Latenz von 74 ms auf 272 ms, da hier für viele verschiedene autoritative Nameserver eine Verbindung aufgebaut werden muss, die nicht mehr für weitere Abfragen benutzt wird. [8, 19]

Durch die Verwendung von TLS über TCP kann bei richtiger Wahl der TLS-Crypto-Suite Perfect Forward Secrecy hergestellt werden. Neben der Verschlüsselung der Nachricht wird bei TLS auch die Integrität dieser sichergestellt.

4.1.2 Confidential DNS

Confidential DNS ist ein von Wijngaards et al. [17] neu entwickeltes Kryptoprotokoll innerhalb des bestehenden DNS Protokolls. Dafür wird ein neues Resource Record vom Type ENCRYPT zur Keygenerierung und Übertragung von verschlüsselten Nachrichten eingeführt. Dabei werden mit einzelnen Flags nach dem ENCRYPT Record entsprechende Algorithmen, und die Art der folgenden Daten bestimmt. Das Protokoll kann sowohl für Kommunikation zwischen autoritativen und rekursiven Nameserver als auch zwischen rekursiven Resolvern und Stub Resolvern verwendet werden. Im folgenden Protokollverlauf wird die Partei, die die DNS-Anfrage startet mit Client bezeichnet, der Gegenüber mit Server.

Um eine verschlüsselte Übertragung zu starten, muss der Client zuerst Schlüsselmaterial vom Server anfordern, sofern kein passendes im Cache liegt. Dazu sendet er eine Anfrage mit dem Typ ENCRYPT und dem QNAME ”.än den Server, welcher diese wiederum mit eigenem Schlüsselmaterial beantwortet. Nachdem der Client einen Schlüssel des Servers hat, sendet dieser erneut eine Anfrage mit dem Typ ENCRYPT und dem Rootlabel als Anfragedomain mit einem zusätzlichen ENCRYPT Record in der Additional Section, welches verschlüsselt die konkrete Anfrage und Schlüsselmaterial für die folgende Kommunikation enthält. Die Antwort des Servers enthält ähnlich zur Anfrage ein RR ENCRYPT, in dem die Antwort mit dem vorher gelieferten Schlüsselmaterial verschlüsselt ist. Sowohl der Server als auch der Client können erhaltene Schlüssel zwischenspeichern und wiederverwenden. Schlägt die Übertragung fehl oder der Server liefert einen Error, so muss der Client auf unverschlüsselte Übertragung umstellen.

Dieses Protokoll stellt weder Perfect Forward Secrecy her, noch prüft es die Integrität und Authentizität der Pakete. Dadurch muss DNSSEC innerhalb der verschlüsselten RR verwendet werden um Integrität sicherzustellen, wodurch das in Abschnitt 3.2 beschriebene Problem, der Möglichkeit die Zone komplett auszulesen, weiterhin besteht. Auch schützt Confidential DNS nur gegen passive Angreifer. Aktive Angreifer hingegen können leicht die Kommunikation auf unverschlüsseltes DNS zwingen, indem sie Nachrichten fehlerhaft manipulieren und dadurch auf normales DNS zurückgegriffen wird.

4.1.3 DNSCurve

DNSCurve ist ein von Daniel J. Bernstein [3] entwickeltes Protokoll für das Domain Name System, welches die Ziele Integrität und Vertraulichkeit für die Kommunikation zwischen autoritativen und rekursiven Nameservern umsetzt. Kryptographische Grundlage bildet die durch das Kryptoframework NaCl bereitgestellte elliptische Kurve. NaCl bietet außerdem Krytoboxen an, die aufbauend auf einem privaten, einem öffentlichen Schlüssel und einer Nonce, einem Wert der für jedes Schlüssel Paar nur ein einziges mal verwendet werden darf, Nachrichten authentifizierend ver- und entschlüsselt. Das gemeinsame Geheimnis wird über Diffie-Hellman auf elliptischen Kurven bestimmt [2].

Jeder DNS Server besitzt einen eigenen privaten und zugehörigen öffentlichen Schlüssel. Der öffentliche Schlüssel wird in ein neues Label in den Domainnamen des DNS Servers ko-

diert und dadurch veröffentlicht. Wird eine Anfrage durch einen DNSCurve Client an einen DNSCurve kompatiblen DNS Server durchgeführt, so wird eine kryptographische Box mit dem eigenen privaten Schlüssel, dem öffentlichen Schlüssel des Empfängerservers und einer Nonce erstellt. In diese wird die originale Anfrage verpackt und zusammen mit dem eigenen öffentlichen Schlüssel und der Nonce and den Empfänger gesendet. Der Empfänger versucht nun mit dem empfangenen öffentlichen Schlüssel des Senders und der Nonce die Kryptobox zu entpacken. Sollte dabei ein Fehler auftreten, wird versucht, das Paket als normales DNS-Paket zu behandeln, um abwärtskompatibel zu bleiben. Konnte das Paket korrekt geöffnet werden, generiert der Server eine neue Nonce, verpackt die Antwort in einer Krytobox und versendet diese zusammen mit der ersten Nonce zurück an den Sender. Dieser versucht die Antwort zu öffnen, und verwirft diese, sofern die Kryptobox nicht korrekt entpackt werden kann. DNSCurve behält dadurch das normale DNS Protokoll bei und führt die Verschlüsselung ohne zusätzliche Nachrichten transparent ein. Durch das behandeln unverschlüsselter eingehender Pakete als normale DNS Anfragen, bleibt es abwärtskompatibel zum bestehenden DNS System, jedoch ohne verschlüsselte Verbindungen auf unverschlüsselte zu degradieren.

Da mit DNSCurve die komplette Verbindung auf Link-Level verschlüsselt ist, wirkt dies Methode effektiv gegen die in Abschnitt 3.1.1 beschriebenen Angriffsmöglichkeiten zwischen rekursiven und autoritativen Nameservern. Mit dem Einsatz von authentifizierender Verschlüsselung fällt die Notwendigkeit der Verwendung von DNSSEC weg und behebt dadurch die in Abschnitt 3.2 aufgezeigten Probleme im Bezug auf den Schutz des Zoneninhalts. Da die öffentlichen Schlüssel eines Nameserver in der jeweils darüberliegenden Zone eingetragen sind, kann auch die Authentizität der Schlüssel sichergestellt werden. Ein Problem besteht dabei lediglich die Schlüssel der Root-Zone zu authentifizieren. Außerdem bietet es, obwohl zur Erstellung des gemeinsamen Geheimnis ein Diffie-Hellman Verfahren verwendet wird, keine Perfect Forward Secrecy, da mit statischen Schlüsseln gearbeitet wird [2].

Neben der Spezifikation von DNSCurve, welche lediglich für den Einsatz zwischen rekursiven und autoritativen Nameservern ausgelegt ist, bietet OpenDNS mit DNSCrypt eine Erweiterung mit ähnlichen Designkriterien für die sichere Kommunikation zwischen Endnutzer und rekursivem Resolver an [15]. Auch hier bleibt jedoch die Angriffsmöglichkeit durch den Betreiber selbst bestehen, da dieser weiterhin alle Nachrichten sehen kann.

4.2 QNAME Minimierung

Stéphane Bortzmeyer [4] schlägt in einem IETF Draft vor die Anfragenamen auf das minimal Nötige zu reduzieren. Bei den aktuellen Implementierungen aller gängigen rekursiven Nameserver wird bei einer iterativen Namensauflösung bei jeder neuen Anfrage die komplette Ursprungsanfrage versendet. So wird an den Root-Server bei einer Anfrage der Domain *www.example.com* die vollständige Domain versendet, obwohl es ausreichend ist zu nach den Nameservern für die *com* Zone zu fragen. Durch die Minimierung der gefragten Domain (QNAME) in der Anfrage, sieht jeder Nameserver nurnoch die für ihn notwendigen und bekannten Details. Ein Problem bei dieser Methode ist die Bestimmung der Zonecuts, also jenen Stellen innerhalb der Domain, ab der ein anderer Nameserver zuständig ist, da dieser nicht an jedem Punkt sein muss. Dieses Verfahren ist mit der ursprünglichen Spezifikation des DNS-Protokolls in RFC1034 [14] und RFC1035 [13] kompatibel.

4.3 RangeQuery

Zhao et al. [18] schlagen vor den Anfragenamen zusammen mit einer Reihe anderer zufälliger Domainnamen zu versenden um die eigentliche Anfrage zu verschleiern. Dafür wird bei einer RangeQuery eine Liste $(name_1, ..., name_m)$ mit $m-1$ zufällig gewählten Domainnamen und dem geforderten Domainnamen versendet. Die $m-1$ zufälligen Domainnamen werden aus einer Menge mit n vorbestimmten gültigen Namen, die in unterschiedlichen vom Benutzer festgelegten Bereichen liegen, ausgewählt. So kann ein Angreifer nicht genau feststellen welcher Domainname das Ziel der Anfrage war. Während Zhao et al. [18] für jede DNS Anfrage die Liste der zufälligen Domain Namen neu zu bestimmen, um eine Erkennungswahrscheinlichkeit von $\frac{1}{m}$ für die echte Domain zu erreichen, widerlegen Lu und Tsudik [12] dies. Die Wahrscheinlichkeit sinkt mit der Anzahl an gesendeten RangeQuerys für die gleiche Domain, da bei mehreren Anfragen nur die Schnittmenge in allen Anfragen vorkommende Domainnamen in Frage kommen.

Abbildung 4 zeigt die Erkennungswahrscheinlichkeit nach k Anfragen mit jeweils einem Domain-Namen Pool der $n = 1024$ mit unterschiedlich großer Anfrageliste m. Es ist dabei leicht zu sehen, dass bei einer kleinen Anfrageliste bereits nach wenigen Wiederholungen die angefragte Domain mit hoher Wahrscheinlichkeit bestimmt werden kann. Lu und Tsudik [12] schlagen zur Lösung dieses Problems vor statt einer zufälligen Liste pro Anfrage eine Zufällige Liste pro Domain zu erstellen. Diese wird für alle weiteren Anfragen an diese Domain verwendet. Dies kann mit einem Pseudozufallszahlengenerator mit einem geheimen Startwert und der Domain erreicht werden.

RangeQuerys sind zusammen mit computational Private Information Retrieval (cPIR), welches im folgenden Abschnitt beschrieben wird, die einzige Methode um angefragte Domains vor rekursiven Nameservern zu verschleiern. Dabei muss bei RangeQuerys der erwünschte Sicherheitsgrad gegen den erzeugten Overhead an Anfragevolumen abgewogen werden.

4.4 PPDNS

Lu und Tsudik [12] beschreiben das Modell Privacy Preserving Domain Name System (PPDNS), bei der das bestehende DNS umgebaut wird, so dass Anfragen gestellt werden, bei der der Nameserver nicht die Möglichkeit hat, die angefragte Domain zu bestimmen. Dazu wird das hierarchische Domain Name System durch eine flache verteile Hashtabellenstruktur (DHT) ersetzt. Als Beispiel wird das bereits existierende Projekt CoDoNS [16] verwendet, in dem jeder Knoten und jede Domain hat einen eindeutigen Wert (ID) aus dem Raum der Hashfunktion hat und der Knoten, dessen ID sich am wenigsten von dem Hash des Domainnamens unterscheidet, diesen permanent speichert.

Der grundlegende Ablauf ist dem der RangeQuery aus Ab-

doi: 10.2313/NET-2014-08-1_18

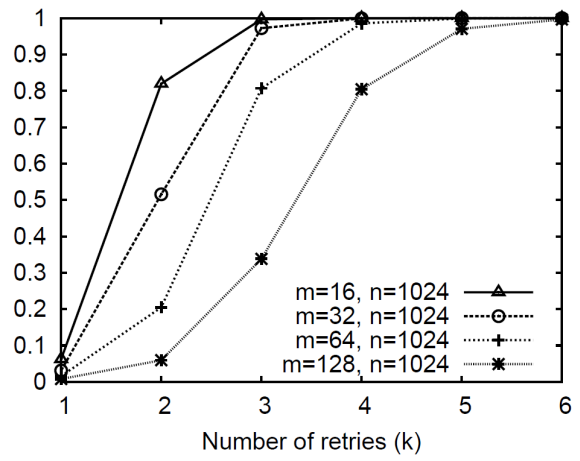

Abbildung 4: Wahrscheinlichkeit die angefragte Domain nach k Wiederholungen zu erkennen [12]

schnitt 4.3 ähnlich. Für eine DNS-Anfrage, bildet der Client einen Bereich, in dem der Hash der gesuchten Domain und $m-1$ weitere Domains liegen. Um den korrekten Bereich der Hashwerte zu berechnen, in dem m Domains existieren muss die Dichte der Domains im Wertebereich der Hashfunktion bekannt sein. Diese bekommt er von seinem konfigurierten Nameserver, der diese aufgrund der Gleichverteilung der Hashfunktion mithilfe seiner eigenen Daten berechnen kann. Nachdem der korrekte Hashbereich gebildet wurde, wird dieser als Anfrage an den Nameserver gesendet. Liegt der angefragte Bereich im Cache des Nameservers, kann sofort geantwortet werden, ansonsten führt er einzelne Bereichsanfragen entsprechend der Struktur der DHT Knoten durch.

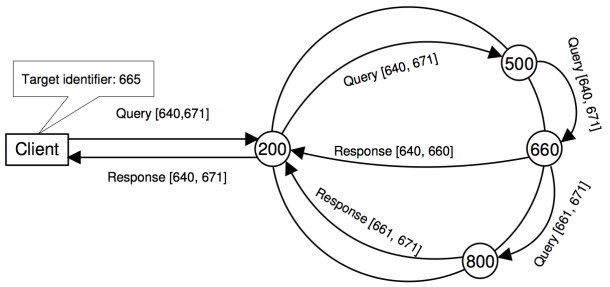

Abbildung 5: RangeQuery in einem DHT basiertem DNS [12]

In Abbildung 5 wird ein Ablauf einer Anfrage des Hashes 665 mit einer Hashfunktion durchgeführt, die einen Wertebreich von 1024 hat, mit einer Dichte von $\frac{1}{2}$ belegt ist und einem Sicherheitsparameter von $m = 16$ Domains pro Anfrage . Aus dem Anfrageziel 665 folgt eine Anfrage mit dem Bereich [640, 671], die an den konfigurierten Knoten mit der ID 200 gesendet wird. Da der Knoten diesen Bereich nicht im Cache liegen hat, wird die Anfrage weitergeleitet, bis alle Teilergebnisse des Bereichs vorliegen. Erst dann wird dem Client geantwortet und dieser kann aus dem Bereich die gewünschte Domain auswählen.

Durch RangeQuerys entsteht ein erheblicher Overhead, da auf jede Anfrage m Antworten versendet werden. Dadurch belegen RangeQuerys bei Verbindungen mit geringer Bandbreite einen Großteil der verfügbaren Übertragunsrate. Um diesen Overhead zu verkleinern, kann zusätzlich zu den RangeQuerys noch ein computational Private Information Retrieval (cPIR) Schema verwendet werden. Dafür wird der geforderte Bereich als eigene Datenbank gesehen, die in einzelne Unterbereiche unterteilt wird. Der Client kann nun einen bestimmten Index aus dieser Datenbank anfordern, ohne dass der Server erkennen kann welchen und ohne die komplette Datenbank zu senden. Lu und Tsudik [12] verwenden hierfür das cPIR Schema nach Gentry-Razam, welches die Ziele mit Arithmetik auf zyklischen Gruppen erreicht.

Lu und Tsudik [12] kommen zu dem Ergebnis, dass cPIR durch den großen Berechnungsaufwand auf aktueller Hardware nicht praktikabel einsetzbar ist. Bereits um einen Durchsatz von etwa 100 Anfragen pro Sekunde zu erreichen werden 100 Prozessoren benötigt. Im Gegensatz dazu ergeben die Simulationen mit reiner RangeQuery akzeptable Ergebnisse. So steigt die durchschnittliche Latenz einer RangeQuery innerhalb des PPDNS kaum an im Vergleich zu einer Anfrage mit lediglich einer einzelnen Domain. Da diese Simulationen jedoch nur auf Verbindungen innerhalb von Backbone Netzwerktopologien stammen, kann nicht davon ausgegangen werden, das dieses Ergebnis auch auf Endnutzer im privaten Bereich übertragbar ist, da hier die zugrundeliegenden Basislatenzen höher und die verfügbare Bandbreite sehr viel geringer ausfällt.

5. FAZIT

Nach Betrachtung möglicher Lösungsansätze zeigt sich, dass es kaum möglich ist, sich effizient vor den in Abschnitt 3.1.2 beschrieben Angriffe auf die Privatsphäre durch Betreiber von rekursiven Nameservern zu schützen, außer selbst die Namen iterativ aufzulösen, wodurch jedoch die autoritativen Nameserver mehr Informationen erhalten. Zwar existieren theoretische Modelle, in denen dieser Schutz gewährleistet werden kann, diese sind jedoch, wie in Abschnitt 4.4 gezeigt wurde, nicht ausreichend effizient um umgesetzt werden zu können. Des weiteren wird ein Umstieg auf ein solches System erschwert dadurch, dass ein kompletter, nicht kompatibler Umbau des Domain Name System nötig wäre. Auch die weitere Alternative der RangeQuerys innerhalb des normalen Domain Name Systems hat durch die Erhöhung des benötigten Traffics und Latenz kaum reelle Chance sich als Standard durchzusetzen.

Neben Effizienzgründen muss zusätzlich auch auf die Sinnhaftigkeit einer Einführung eines solchen Systems zur Verschleierung der DNS-Anfragen vor dem rekursiven Nameserver in Betracht gezogen werden. In vielen Netzwerksituationen, besonders im privaten Bereich, wird ein rekursiver Resolver des Internet Service Providers (ISP) verwendet. Das ist insofern problematisch, da der komplette Traffic durch das Netz des ISP muss. Dabei kann, sofern kein IP-Anonymisierungsdienst wie Tor verwendet wird, mindestens die IP bestimmt werden, mit der sich auf eine DNS-Anfrage folgend verbunden wird. Aus dieser kann unter Umständen per Reverse DNS Lookup ein Hostname gebildet werden und in Falle von RangeQuerys mit der ursprünglichen DNS-Anfrage verglichen werden. Auf diesen Weg kann der ISP

doi: 10.2313/NET-2014-08-1_18

die aufgelösten Namen zum Teil bestimmen obwohl diese in der originalen Anfrage verschleiert wurden.

Im Gegensatz zu den Lösungen, die auf Verschleierung der Anfrage vor den Serverbetreibern beruhen, finden Lösungsansätze für eine vertrauliche Datenübertragung zwischen zwei Parteien mehr und mehr Aufmerksamkeit und Attraktivität. Dabei hebt sich DNSCurve vor allem mit einem einfachen Protokoll und schneller Kryptographie auf Basis von elliptischen Kurven von Confidential DNS und t-DNS ab, welches durch die Verwendung von TCP und TLS als Basisprotokoll vor allem eine Erhöhung der Antwortzeiten bei der Kommunikation zwischen rekursiven und autoritativen Nameserver und der verbrauchten Ressourcen mit sich zieht. Auch lässt sich DNSCurve recht leicht in die bereits vorhandene Infrastuktur einbinden. Deshalb scheinen DNSCurve und DNSCrypt vielversprechende Kandidaten für einen standardisierten Einsatz in der Verschlüsselung von DNS-Traffic zu sein. So nutzt OpenDNS bereits seit Februar 2010 DNSCurve zur Verschlüsselung und Authentifikation von DNS-Traffic zwischen DNSCurve-fähigen autoritativen Nameservern [7]. Außerdem bietet es mit DNSCrypt bereits Software zur Verschlüsselung der Daten zwischen Stub Resolver und rekursive Nameserver an.

6. LITERATUR

[1] D. Atkins. Threat analysis of the domain name system (DNS). RFC 3833, RFC Editor, August 2004.

[2] D. J. Bernstein. Cryptography in NaCl. *Networking and Cryptography library*, 2009.

[3] D. J. Bernstein. DNSCurve: Usable security for DNS. http://dnscurve.org/, Jan 2011.

[4] S. Bortzmeyer. Possible solutions to DNS privacy issues. Internet-Draft draft-bortzmeyer-dnsop-privacy-sol-00, IETF Secretariat, December 2013.

[5] S. Bortzmeyer. DNS privacy considerations. Internet-Draft draft-bortzmeyer-dnsop-dns-privacy-02, IETF Secretariat, April 2014.

[6] M. Dempsky. DNSCurve: Link-level security for the domain name system. Internet-Draft draft-dempsky-dnscurve-01, IETF Secretariat, February 2010.

[7] M. Dempsky. OpenDNS adopts DNSCurve. http://blog.opendns.com/2010/02/23/opendns-dnscurve, February 2010.

[8] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, and D. Wessels. Starting TLS over DNS. Internet-Draft draft-hzhwm-start-tls-for-dns-00, IETF Secretariat, February 2014.

[9] S. Josefsson. DNSSEC Walker. http://josefsson.org/walker/, Nov 2005.

[10] P. Koch. Confidentiality aspects of DNS data, publication, and resolution. Internet-Draft draft-koch-perpass-dns-confidentiality-00, IETF Secretariat, November 2013.

[11] B. Laurie, G. Sisson, R. Arends, and D. Blacka. DNS security (DNSSEC) hashed authenticated denial of existence. RFC 5155, RFC Editor, March 2008.

[12] Y. Lu and G. Tsudik. Towards plugging privacy leaks in the domain name system. In *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on*, pages 1–10, Aug 2010.

[13] P. Mockapetris. Domain names - concepts and facilities. RFC 1034, RFC Editor, November 1987.

[14] P. Mockapetris. Domain names - implementation and specification. RFC 1035, RFC Editor, November 1987.

[15] OpenDNS. Introducing DNSCrypt. http://www.opendns.com/about/innovations/dnscrypt.

[16] V. Ramasubramanian and E. G. Sirer. Beehive: The design and implementation of a next generation name service for the internet. In *In ACM SIGCOMM*, 2004.

[17] W. Wijngaards. Confidential DNS. Internet-Draft draft-wijngaards-dnsop-confidentialdns-01, IETF Secretariat, March 2014.

[18] F. Zhao, Y. Hori, and K. Sakurai. Analysis of privacy disclosure in DNS query. In *Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on*, pages 952–957, April 2007.

[19] L. Zhu, Z. Hu, J. Heidemann, D. Wessels, A. Mankin, and N. Somaiya. T-DNS: Connection-oriented DNS to improve privacy and security. Technical Report ISI-TR-2014-688, USC/Information Sciences Institute, February 2014.