DNS Sicherheit

Ausarbeitung zum Kryptologie-Projekt

Wilhelm Bartel (19INM-VZ) Justin Kromlinger (19INM-VZ)
Tom Wegener (18INM-TZ)

2021-01-13

Inhaltsverzeichnis

1 Grundlagen

Das Domain Name System (kurz: DNS) ist eine verteilte Datenbank zu Namensauflösung. Dabei werden Domain Namen wie zum Beispiel `htwk-leipzig.de` aufgelöst und Daten zugeordnet.

1.1 Aufbau

Das Domain Name System arbeitet mit sogenannten „Zonen“, die hierarchisch abgefragt werden. In einer Domain wird jede Zone dabei von rechts nach links durch einen „.“ getrennt, zum Beispiel `www.htwk-leipzig.de.`, wobei der letzte Punkt optional ist.

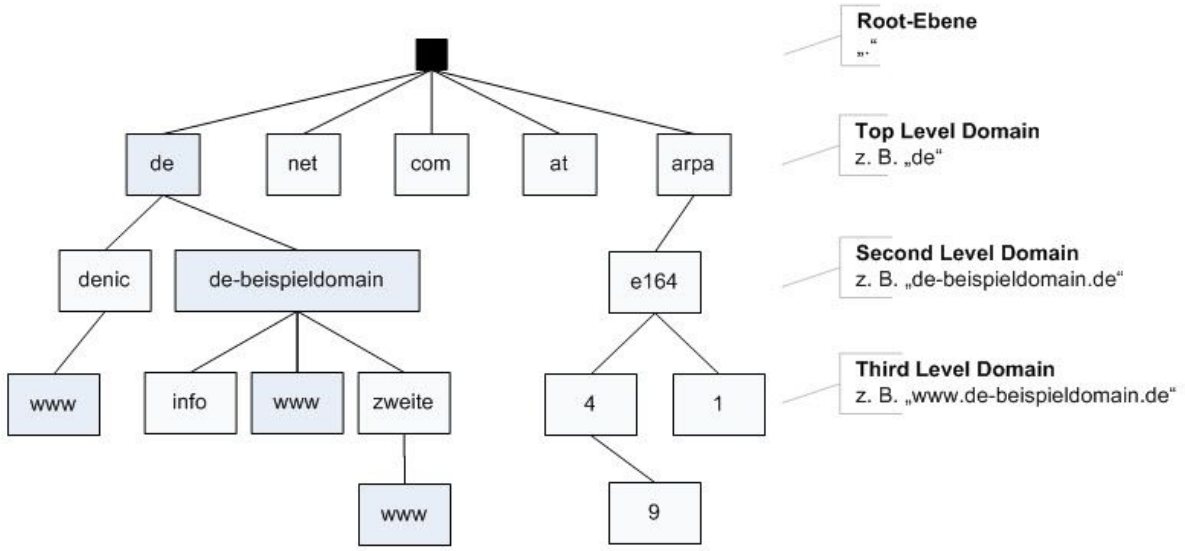

Abbildung 1: DNS Zonen (Denic 2020)

Jeder Knoten in diesem Grafen kann dabei ein eigener DNS-Server sein, der beliebig viele untergeordnete Knoten selbst verwaltet. Alle untergeordneten Zonen, die nicht selbst verwaltet werden, werden untergeordnete DNS-Server ausgelagert. Die Wurzel dieses DNS Baums bilden die sogenannten Root-DNS-Resolver.

1.2 Funktionsweise

1.2.1 Ressource Records

Abgefragt werden sogenannte Ressource Records, welche der Domain Daten zuweisen. Insgesamt sind 55 (Wikipedia 2020) Ressource Records definiert, mit vorgesehenen Funktionen. Darunter befinden sich Records wie zum Beispiel `A` oder `AAAA`, der die IPv4, respektive IPv6 Adresse des Servers nennt, der unter der Domain erreichbar sein soll. Des Weiteren existieren Ressource Records zum Verweis auf einen Mailserver (`MX`) und zur Angabe von Freitext (`TXT`).

1.2.2 Abfrage

Oftmals findet eine DNS Abfrage vor dem Aufruf einer Webseite statt. Dabei wird der `A` Record für die Domain der aufzurufenden Webseite abgefragt. Im Beispiel von `www.wikipedia.org` wird als Erstes ein Root-DNS-Resolver befragt, dieser leitet die Anfrage dann an den DNS Server der Zone `org` weiter, welcher wieder rum an den DNS Server der Zone `wikipedia.org` weiterleitet. Dieser antwortet dann mit dem gewünschten `A` Record von `www.wikipedia.org` und liefert damit die IP-Adresse des Servers.

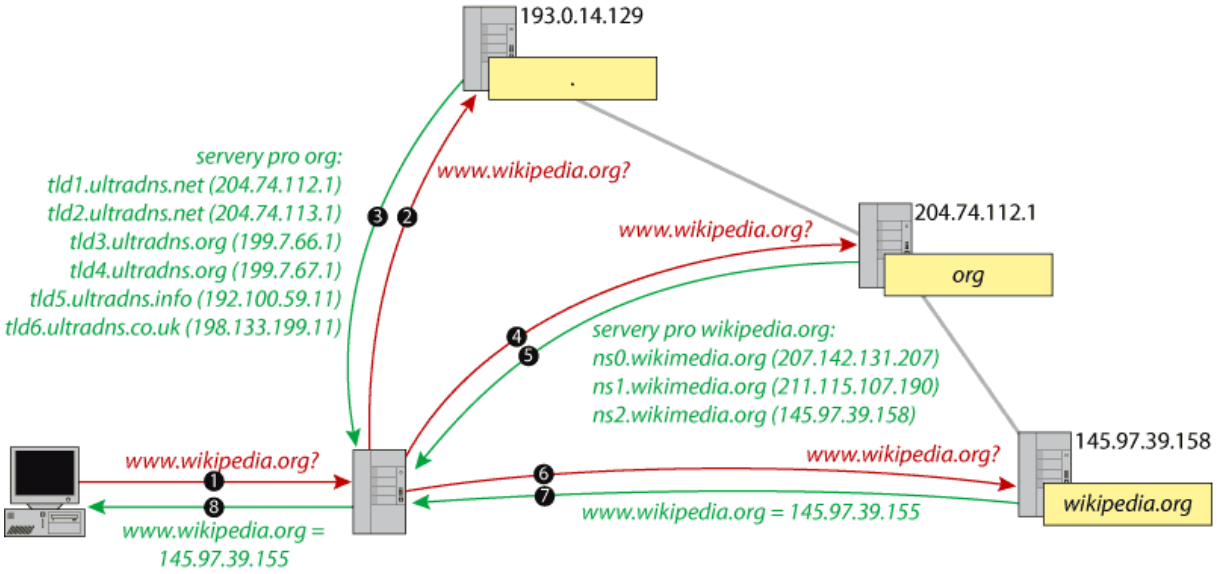

Abbildung 2: DNS Abfrage Reihenfolge (Wikimedia Commons 2006)

1.2.3 Caching

Einem Ressource Record wird immer eine Time-to-live (TTL) zugewiesen. Diese gibt an, wie lange ein anderer DNS Server den Record im Cache behalten darf. Dadurch kann ein DNS Server mit den Records anderer Zonen antworten, ohne dass die gesamte Zonenkette durchlaufen werden muss. Das spart Netzwerkauslastung und verringert Latenzen.

2 Angriffsmöglichkeiten

Die Angriffsmöglichkeiten auf das Domain Name System entstehen dadurch, dass alle Abfragen mit dem UDP Protokoll arbeiten und weder verschlüsselt noch signiert sind.

2.1 Integrität

Da die Antworten des Servers nicht signiert sind, könnten diese von einem Dritten verändert werden. Die Folgen davon sind, dass der Nutzer sich trotz der richtigen Domain am Ende mit einem anderen Server verbindet als gewünscht. Angreifer können dies dann weiter ausnutzen um zum Beispiel Zugangsdaten auslesen.

Eine Erweiterung dieses Vorgehens ist das sogenannte DNS Spoofing. Hierbei tauscht ein Angreifer den DNS-Server des Nutzers durch einen eigenen aus. Das ermöglicht ihm, vollständige Kontrolle über alle Server zu erlangen, auf die der Nutzer mittels Domains zugreift.

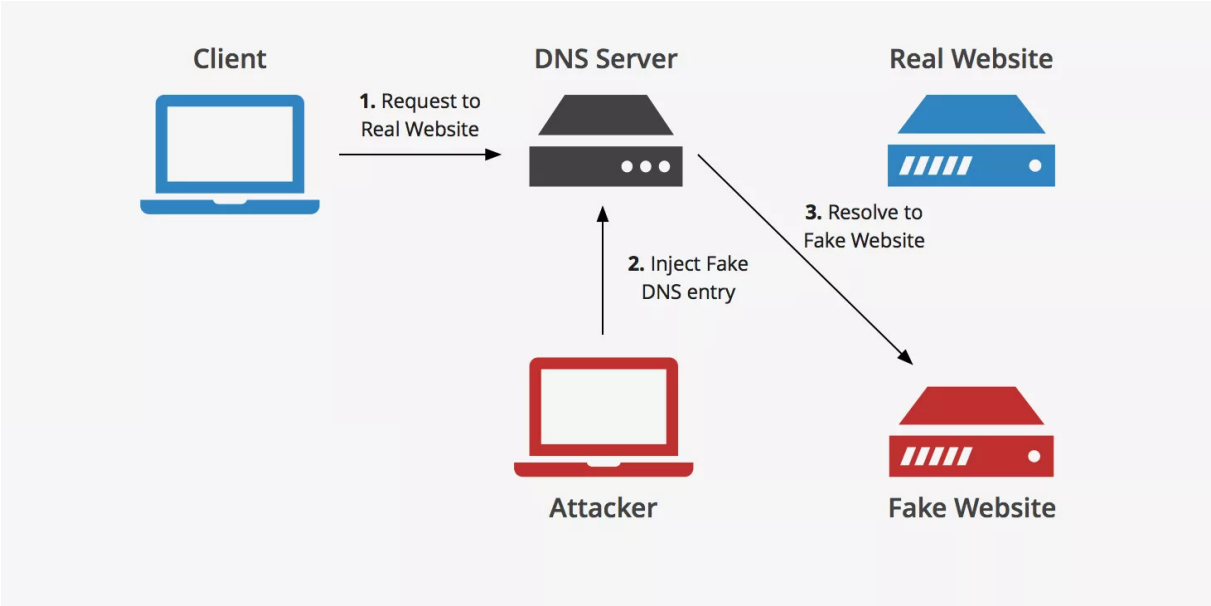

Abbildung 3: DNS spoofing (keycdn 2008)

2.2 Anonymität

Aufgrund der unverschlüsselten Kommunikation kann die gesamte Netzwerkkette die Anfragen nachvollziehen. Diese beinhaltet sowohl jeden abgefragten DNS-Server, als auch den Internetprovider. Dabei können dann Schlüsse gezogen werden, welche Webseiten der Nutzer besucht, auch wenn die Kommunikation zur Webseite dann verschlüsselt abläuft.

3 TSIG zum Update von DNS-Einträgen

Secret Key Transaction Authentication (TSIG, Transaction SIGnature) ist eine im RFC 2845(ISC 2000) beschriebene Erweiterung für DNS um die Authentizität von DNS-Partnern zu überprüfen und die Integrität von Updates sicherzustellen.

Beide Seiten werden im Voraus mit einem geteilten Geheimschlüssel ausgestattet. An ein Update wird dann ein zusätzlicher Record angehangen (TSIG) welcher im Datenfeld verschlüsselt einen Zeitstempel, den verwendeten Hashing Algorithmus und den Hash des Updates enthält. Durch die Verschlüsselung wird die Authentizität des Absenders und durch den Hash die Integrität der Daten überprüft. Der Zeitstempel stellt sicher, dass Update-Anfragen nicht wiederverwendet werden können, beide Seiten müssen also ihre Systemzeit synchronisieren. Bei einem erfolgreichen Update wird die Antwort des Empfängers ebenfalls mit einem TSIG Record versehen.

Damit mehrere Clients einen Server updaten können werden mehrere Schlüssel mit eigenem Namen gespeichert.

Feld	Bytes	Wert	Beschreibung
NAME	Max. 256	Variabel	Name des verwendeten Schlüssels
TYPE	2	TSIG (250)	
CLASS	2	ANY (255)	
TTL	4	0	TSIG Records werden nicht gecached
RDLENGTH	2	Variabel	Länge des RDATA Felds
RDATA	RDLENGTH	Variabel	Struktur aus Zeitstempel, Algorithmus und Hash

Um die Verteilung von Schlüsseln zu erleichtern wurde im RFC 2930(Motorola 2000) ein weiterer Resource Record (TKEY) definiert. Da die ursprüngliche Definition nur MD5 als Hashing Algorithmus erlaubte wurde im RFC 4635(Laboratories 2006) zusätzlich die Verwendung von SHA128 und SHA256 definiert.

4 DNSSEC

Die Domain Name System Security Extensions (DNSSEC) sind wie TSIG Sicherheitserweiterungen für DNS, statt Updates betreffen sie allerdings Abfragen von Resolvern. Sie wurden von der Internet Engineering Task Force (IETF) entwickelt und lösen Probleme der Authentizität, Integrität sowie Existenzverweigerung, nicht jedoch das Problem der Vertraulichkeit. RFC 3833(D. Atkins 2004) dokumentiert diese Probleme und wie DNSSEC sie zu lösen gedenkt.

DNSSEC arbeitet mit einer Vertrauenskette und Asymmetrischer Verschlüsselung – den öffentlichen bekannten Schlüsseln der Rootserver wird vertraut, diese signieren die öffentlichen Schlüssel einer ihnen bekannten Zone welche dann wiederum Sub-Zonen signieren können. Im Beispiel von wikipedia.com übergab der Domaineigentümer seinen generierten öffentlichen Schlüssel an den Verantwortlichen der .com TLD (Verisign), welcher diesen mit seinem privaten Schlüssel signiert. Dessen öffentlicher Schlüssel ist mit den privaten Schlüsseln der Rootserver signiert.

DNSSEC führt hierfür prinzipiell drei neue Ressource Records ein. DNSKEY enthält den öffentlichen Schlüssel, DS (delegation signer) enthält den Namen einer delegierten Zone (zum Beispiel wikipedia) und den zu erwartenden DNSKEY Eintrag. Abfragende können mit dem RRSIG (resource record signature) die Authentizität und Integrität der übertragenen Resource Records überprüfen – er enthält die mit dem privaten Schlüssel erstellte Signatur der Record Menge, welche mit dem öffentlichen Schlüssel im DNSKEY Eintrag verifiziert werden kann.

Hinsichtlich verwendeter Algorithmen ist mindestens RSA/SHA-256 Pflicht, Ed25519 wird empfohlen.

4.1 Signierte Existenzverweigerung

Ursprünglich wurden NXDOMAIN Antworten – eine angefragte Zone existiert nicht – dynamisch über einen privaten Schlüssel im Server signiert. Dies hat den Nachteil, dass infizierte Nameserver die von DNSSEC gelösten Probleme zunichtemachen. Stattdessen wurden die NSEC (next secure record) Einträge eingeführt. Diese enthalten nach Sortierung jeweils zwei aufeinander folgende Zone Einträge und sind vorab signiert, Nameserver müssen nun also keinen privaten Schlüssel vorhalten. Bei einer NXDOMAIN Antwort wird der NSEC Eintrag mitgeliefert, zwischen dessen beiden Paaren sich die Zone geordnet befinden würde, wodurch der Empfänger sicherstellen kann, dass die Zone nicht existiert.

Dies führt leider zu einem neuen Problem – dem sogenannten Zone Walking. Da mit NSEC über die Record Names einer Zone enumeriert werden kann, lassen sich darüber alle Namen einer Zone herausfinden. Um dies zu lösen wurde NSEC3 (next secure record version 3) eingeführt, welcher wie NSEC funktioniert, jedoch die Namen vor der Sortierung kryptografisch hasht, um die Enumerierung zu verhindern. Da dies mit einer lokalen Wörterbuchattacke angreifbar ist, werden mit NSEC5 Signed Key Hashes benutzt(J. Vcelak 2018).

5 DNSCrypt

DNSCrypt ist ein Protokoll zur Verschlüsselung und Authentizität-Prüfung von DNS-Verkehr, jedoch nur um Manipulation bzw. Fälschung der Antworten zu erkennen, nicht um die Anfragen an sich vor Dritten zu schützen.

DNSCrypt funktioniert dabei über ein Server-Client-System, welches über TCP oder UDP kommuniziert, dabei wird jedoch TCP präferiert. Am Anfang sendet der Client eine nicht authentifizierte Anfrage an einen Resolver auf dem DNSCrypt installiert ist. Die Anfrage enthält dabei die unterstützten DNSCrypt-Versionen und einen Identifier des angefragten Providers. Anschließend antwortet der Resolver mit einem Satz an signierten Zertifikaten, die von dem Client mit einem sogenannten „provider public key“ verifiziert werden müssen. Die Zertifikate enthalten Validitätsperiode, eine Seriennummer und die eine Kommunikationsweise. Anschließend wählt der Client das Protokoll mit der höchsten Seriennummer aus und nutzt die darin vorgegebenen Kommunikationsweise, also den beschriebenen Schlüssel-Austausch-Mechanismus und den Authentizierungs-Algorithmus und den öffentlichen Schlüssel (resolver public key). Die Informationen und die Identifikationsnummer des Zertifikates sowie der öffentliche Schlüssel werden anschließend dafür verwendet die DNS-Anfrage zu senden. Anschließend entschlüsselt und überprüft der Resolver die Anfrage und verschlüsselt die Antwort auf die gleiche Art und Weise (jedisct1 2017).

5.1 Verfügbarkeit

DNSCrypt wird unter Anderem von Anwendungen von AdGuard oder Surfshark unterstützt. Als Resolver können Server von zum Beispiel Adguard oder Cisco genutzt werden.

6 DNS-over-TLS (DoT)

Wie auch DNSCrypt löst DoT das Problem der Vertraulichkeit sowie Authentizität und Integrität von DNS Abfragen zwischen Nutzern und DNS Resolvern. Es ist ein von der IETF vorgeschlagener Standard der im RFC 7858(Laboratories 2006) definiert ist.

Um dies zu bewerkstelligen wird das normale DNS Protokoll über eine TLS-Verbindung mit dem angesprochenen Resolver verschlüsselt und läuft nicht mehr über 53/UDP, sondern standardmäßig über 853/TCP.

Durch den Umstieg auf TCP und den TLS-Overhead tritt ein gewisser Performance-Verlust ein, welcher lokal mit Caching negiert werden kann. Da nur die Verbindung zum Recursive Resolver über TLS erfolgen muss ist die Wahrung der Vertraulichkeit nur gegenüber Dritten gegeben, eine Ende-zu-Ende-Verschlüsselung ist nicht gegeben.

6.1 Verfügbarkeit

Viele bekannte DNS-Anbieter wie zum Beispiel Cloudflare, Quad9 oder Google bieten seit längerem zusätzlich DoT an.

Linux und Windows Nutzer können von NLnet Labs entwickelte Clients oder Resolver wie unbound nutzen. Auf Mobilgeräten kann unter Android seit Version 9 Systemweit ein DoT Eintrag für Mobilfunk und WLAN gesetzt werden, unter iOS steht seit Version 14 ebenfalls eine Einstellung bereit die aber nur über Drittanbieter-Anwendungen gesetzt werden kann.

7 DNS-over-HTTPS (DoH)

Wie DNSCrypt und DNS-over-TLS löst DNS-over-HTTTPS auch die Probleme der Vertraulichkeit und der Authentizität und Integrität von DNS-Abfragen zwischen User und DNS-Resolvern. Eine gewisse Ähnlichkeit des Nutzens wird auch häufiger erwähnt. DoH wurde im Oktober 2018 als RFC8484 (P. Hoffman (ICANN) 2018) von der IETF als vorgeschlagener Standard definiert.

Der Standard definiert, dass die DNS-Abfragen nicht über UDP verschickt werden, sondern über ein HTTP-GET oder -POST-Request an einen DNS-Server geschickt werden, dieser Request ist via TLS verschlüsselt. Die Anfragen sehen wie normaler HTTPS-Verkehr aus und nutzen auch den Port 443, dadurch wird es verunmöglicht die verschlüsselten DNS-Abfragen zu blockieren, ohne auch die restliche Kommunikation über HTTP/S zu blockieren.

Durch die Nutzung von HTTPS entsteht im Vergleich zu DoT ein geringer Overhead, jedoch ist DoH laut Mozilla nur geringfügig langsamer, manchmal auch schneller (Deckelmann 2019). (https://blog.mozilla.org/futurereleases/2019/04/02/dns-over-https-doh-update-recent-testing-results-and-next-steps/)

7.1 Verfügbarkeit

DNS-over-HTTPS wird ebenso wie DoT unter anderem von Google, Cloudflare und Quad9 unterstützt.

Das Protokoll wird eher von Anwendungs-Ebene unterstützt anstatt wie klassisches DNS auf System-Ebene. So ist es zum Beispiel im Firefox-Browser der Firma Mozilla, Edge von Microsoft oder im Chrome-Browser von Google integriert. Außerdem wird es von macOS 11 und iOS 14 unterstützt und in einigen Insider Preview Builds.

Linux-Nutzer können, wie bei DoT, auch auf den Recursive Resolver unbound setzen oder auch den „doh-client“ des github-users LinkTedd.

Literatur

D. Atkins, R. Austein, IHTFP Consulting. 2004. „Threat Analysis of the Domain Name System (DNS)“. https://tools.ietf.org/html/rfc3833.

Deckelmann, Selena. 2019. „DNS-over-HTTPS (DoH) Update – Recent Testing Results and Next Steps“. https://blog.mozilla.org/futurereleases/2019/04/02/dns-over-https-doh-update-recent-testing-results-and-next-steps/.

Denic. 2020. „Das Domain Name System (DNS)“. https://www.denic.de/wissen/domain-name-system-dns/.

ISC, Motorola, NAI Labs. 2000. „Secret Key Transaction Authentication for DNS (TSIG)“. https://tools.ietf.org/html/rfc2845.

J. Vcelak, S. Goldberg, CZ.NIC. 2018. „NSEC5, DNSSEC Authenticated Denial of Existence“. https://tools.ietf.org/id/draft-vcelak-nsec5-08.html.

jedisct1. 2017. „DNSCrypt version 2 protocol specification“. https://dnscrypt.info/protocol/.

keycdn. 2008. „DNS Spoofing“. https://www.keycdn.com/support/dns-spoofing.

Laboratories, Motorola. 2006. „HMAC SHA TSIG Algorithm Identifiers“. https://tools.ietf.org/html/rfc4635.

Motorola. 2000. „Secret Key Establishment for DNS (TKEY RR)“. https://tools.ietf.org/html/rfc2930.

P. Hoffman (ICANN), P. McManus(Mozilla). 2018. „DNS Queries over HTTPS (DoH)“. https://tools.ietf.org/html/rfc8484.

Wikimedia Commons, Pavel.satrapa at. 2006. „DNS Abfrage Bild“. https://commons.wikimedia.org/wiki/File:Dns-wikipedia.png.

Wikipedia. 2020. „Ressource Record“. https://de.wikipedia.org/wiki/Resource_Record.