Veilig(er) internet (2)

Ton Valkenburgh

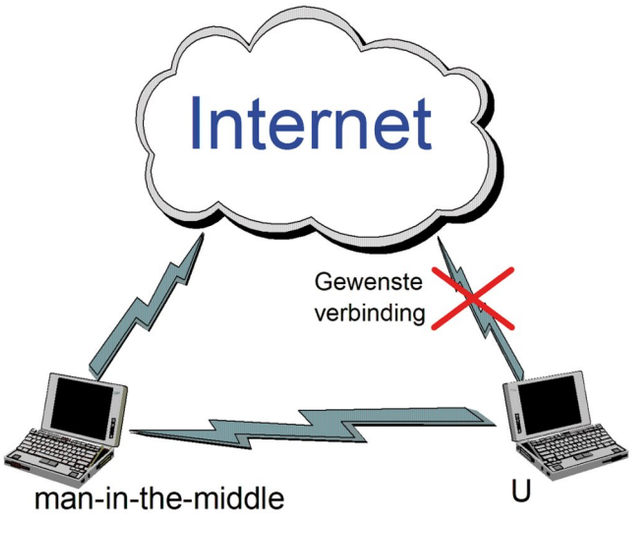
Veiligheid en privacy zijn steeds belangrijkere onderwerpen geworden. Gelukkig wordt er daarom ook aan de veiligheid van internet gewerkt. In SoftwareBus 2021-3 heb ik laten zien hoe o.a. in Android de verbindingsopbouw veiliger kan worden gemaakt. In dit artikel laat ik zien hoe dit te doen op de laptop.

Inleiding

Een veiligheidszwakte bij internet is de manier waarop een sessie wordt opgezet. We zijn allemaal gewend namen te gebruiken voor bijvoorbeeld een website. Deze naam moet worden omgezet in een fysiek IP-adres. Hiervoor gebruiken we domeinnaamservers: het zogenaamde domeinnaamsysteem (DNS).

In het internet zijn de domeinnaamservers gekoppeld. Als u een internetverbinding opzet wordt de naam van de gewenste site naar de domeinnaamserver van uw internetprovider gestuurd.

In de domeinnaamserver wordt deze naam omgezet in een fysiek IP-adres, bijvoorbeeld 23.45.7.100. Deze berichten zijn gewoon leesbaar. Als een wifi-toegangspunt met een identieke naam uw verbinding heeft overgenomen, kunnen al uw gegevens worden opge-vangen en meegelezen: een zogenaamde *man-in-the-middle* (het kan ook een vrouw zijn...).



Om dit te voorkomen is een aantal oplossingen bedacht, met elk hun voor- en nadelen. Op de voor- en nadelen van de verschillende opties ga ik in dit artikel niet in. Op internet is daarover genoeg te vinden.

Het zwakke punt zit in de opbouw van de verbindingssessie. Dan is zichtbaar waarnaar een verbinding wordt opgebouwd. Deze sessie zou versleuteld moeten zijn.

Voor de browser is DNS over HTTPS (DoH) van belang. Dat geeft een versleutelde sessieopbouw, gecombineerd met versleutelde gegevensoverdracht. Als je andere toepassingen gebruikt is DNS over TLS (DoT) van belang; dat zorgt altijd voor een versleutelde sessie-opbouw. De versleuteling van de gegevens moet door de betreffende toepassing gebeuren. Het is echter ook mogelijk om alle DNS-aanvragen via DoH te laten lopen.

Als u dus met een laptop op een terras zit en een openbare wifi-verbinding gebruikt, kan die worden gekaapt, maar dit is ook mogelijk als u thuis met wifi werkt.

Hoe we dit kunnen voorkomen wordt hieronder uitgelegd. Eerst een gedeeltelijke herhaling van DNS over HTTPS en daarna DNS over TLS voor Linux en DNS over HTTPS voor Windows.

Veilige DNS-servers

Ondersteuning voor DoH en DoT is nog vrij beperkt en u vindt het vooral bij wereldwijd werkende providers, zoals Google, Cloudflare, Quad9 en NEXTDNS (link 1).
Google kiezen zou hier het paard achter de wagen spannen zijn; de DNS-provider ziet namelijk wel het IP-adres. Een uitgebreide lijst van DNS-providers die versleuteld DNS ondersteunen is te vinden bij link 2.

Quad9 (link 3) is degene die ik prefereer. Het is een tegenwoordig in Zwitserland gevestigde non-profitorganisatie. De oprichters zijn IBM, Packet Clearing House en Global Cyber Alliance. Er wordt geen persoonlijke informatie bijgehouden en dus ook niets verkocht aan derden. De financiering is gebaseerd op donaties.

Zoals bij veel secure DNS-servers kan er ook worden gefilterd en is het op die manier mogelijk om o.a. malware buiten de deur te houden. Quad9 heeft ook een DNS-service die niet filtert, maar die dan ook geen versleuteling ondersteunt. In alle voorbeelden die ik hierna geef, gebruik ik Quad9 met filtering.

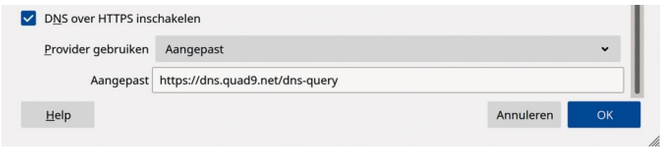
DNS over HTTPS

De juiste manier om naar een website te gaan is met het https-protocol. De verbinding tussen de browser en de website is dan versleuteld. Helaas ondersteunen nog niet alle websites dit.

Het verbreidt zich wel steeds meer, omdat het ondersteunen hiervan een hogere ranking in de zoekresultaten van Google geeft. De verbindingsopbouw is echter niet versleuteld. Als deze ook zou zijn versleuteld was uw privacy beter gewaarborgd. Normaal wordt voor de sessieopbouw het User Datagram Protocol (UDP) gebruikt. Hierbij wordt niet getest of een datapakket is aangekomen.

Bij DNS over HTTPS (DoH) wordt echter het Transmissie Control Protocol (TCP) gebruikt over een versleutelde verbinding. Hierbij wordt gecontroleerd of een datapakket is aangekomen en worden transmissiefouten gecorrigeerd.
Ook wordt de DNS-informatie versleuteld. Het resultaat is een betrouwbaardere verbindingsopbouw met grotere veiligheid en privacy.
DoH gebruikt poort 443, de standaardpoort voor https. DoH wordt o.a. standaard door Firefox ondersteund. Bij *Opties* >

Algemeen > Netwerkinstellingen > DNS over HTTPS inschakelen kiest u een DNS-service die het protocol ondersteunt. Bij Linux vindt u het bij *Bewerken > Voorkeuren > Algemeen> Netwerkinstellingen*.



Ook bij de Chrome-browser kan dit worden ingesteld. Helaas is dit niet mogelijk onder Linux. Google wil zich niet conformeren aan de Linuxstandaarden. Daarom missen we deze functie dus bij Chrome onder Linux. Er is gelukkig een eenvoudige oplossing die even veilig is.
Gebruik de hierna genoemde DNS over TLS en installeer de add-on HTTPS-Everywhere. Deze combinatie geeft dezelfde beveiliging als DNS over HTTPS.

Het nadeel van DoH in de browser is dat het alleen voor de browser werkt en niet voor bijvoorbeeld uw mail-sessies. Ook wordt op dit moment de Server Name Indication (SNI) nog niet versleuteld. Met SNI wordt aangegeven welke service men wil gebruiken bij een gemeenschappelijk IP-adres van de host. Als SNI niet is versleuteld, is de hostnaam waarmee men wil verbinden nog zichtbaar. Mozilla werkt op dit moment aan een oplossing voor Firefox (link 4).

DNS over TLS

Bij DNS over TLS (DoT) gaan alle DNS-sessies via een versleutelde TCP-verbinding. De opgebouwde verbinding is echter alleen versleuteld als dat door de betreffende toepassing wordt verzorgd, zoals uw mail-programma of browser. Sommige routers ondersteunen DoT. Dan werkt het voor al uw apparaten.

Voor mobiele apparaten – zeker die voor buitenshuis – zal het per apparaat moeten worden ingesteld. Juist hier is het belangrijk. Onderweg is de kans op een *man-in-the-middle* bij openbare netwerken namelijk groter.

Als u niet alleen een browser wilt gebruiken, maar bijvoorbeeld ook uw mail-programma, heeft u dus DoT-ondersteuning nodig. Android ondersteunt het standaard sinds versie 9.0. Hoe dit te activeren leest u in SoftwareBus 2021-3.

Ook vanaf iOS 14 zijn er oplossingen met apps uit de store. Voor Windows en Mac-OS zijn programma's beschikbaar om DoT te kunnen gebruiken. In Linux is de functie al sinds Ubuntu 18.04 beschikbaar.
DoT gebruikt poort 853. Sommige firewalls blokkeren standaard deze poort. Omdat de functie in Linux standaard aanwezig is, behandel ik die eerst. Daarna is de implementatie voor Windows eenvoudiger uit te leggen.

DoT in Linux

Ik heb deze functie getest in Kubuntu 20.04. Ik ben recentelijk naar deze distributie overgegaan. Over het waarom zal ik nog wel eens een artikel schrijven.

We zullen eerst de configuratie voor DoT moeten aanmaken. Dat doen we door het bestand **resolved.conf** aan te passen.

Open in de terminal met bijvoorbeeld de editor mousepad het bestand **resolved.conf:**

sudo mousepad /etc/systemd/resolved.conf

en pas het aan volgens onderstaande lijst.

```
[Resolve]
#DNS=
DNS=9.9.9.9 149.112.112.112
#FallbackDNS=
#Domains=
Domains=~.
#LLMNR=no
#MulticastDNS=no
#DNSSEC=no
DNSSEC=yes
#DNSOverTLS=no
DNSOverTLS=yes
#Cache=no-negative
#DNSStubListener=yes
#ReadEtcHosts=yes
```

Nadat het bestand is opgeslagen herstarten we de service met het commando:

sudo systemctl restart systemd-resolved

N.B.
- Achter het teken **#** staan de standaardwaarden;
- Ik heb twee IP-adressen van Quad9 ingevuld. Niets weerhoudt u ervan om ook voor alle zekerheid nog adressen van andere providers toe te voegen. De IP-adressen van de DNS-servers zet u achter elkaar met een spatie ertussen;
- DNSSEC=no maakt de afhandeling sneller;
- DNSOverTLS=yes is de beste keuze. Een alternatief is DNS OverTLS=opportunistic; dit zorgt er voor dat bij een fout een 'standaard' DNS-service wordt gebruikt. Realiseer u dan wel dat de *man-in-the-middle* die fout kan creëren en er toch tussen kan kruipen.

Met deze instellingen gebruiken zowel LAN-sessies als wifi-sessies de versleutelde DNS-server.
Helaas biedt de website van Quad9 op dit moment niet de mogelijkheid om te checken of de instellingen kloppen. We kunnen dit wel doen bij Cloudflare. Als DNS vullen we dan tijdelijk 1.1.1.1 in en gaan met de browser naar:

https://www.cloudflare.com/ssl/encrypted-sni/

Vergeet niet het na de test weer op Quad9 of de door u gekozen DNS-server te zetten.

Als u details van het netwerkverkeer wilt zien en/of controleren is Wireshark een handig programma. Het zit in de meeste distributies. Het behandelen van Wireshark valt echter buiten het bestek van dit artikel.

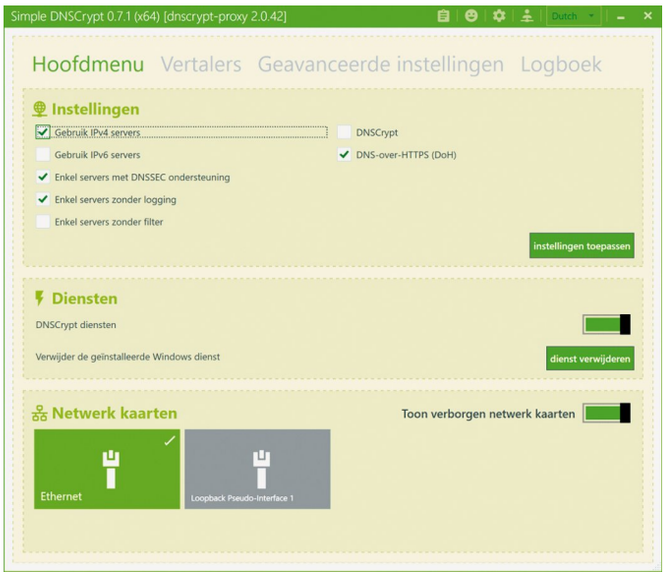
DoT of DoH in Windows 10

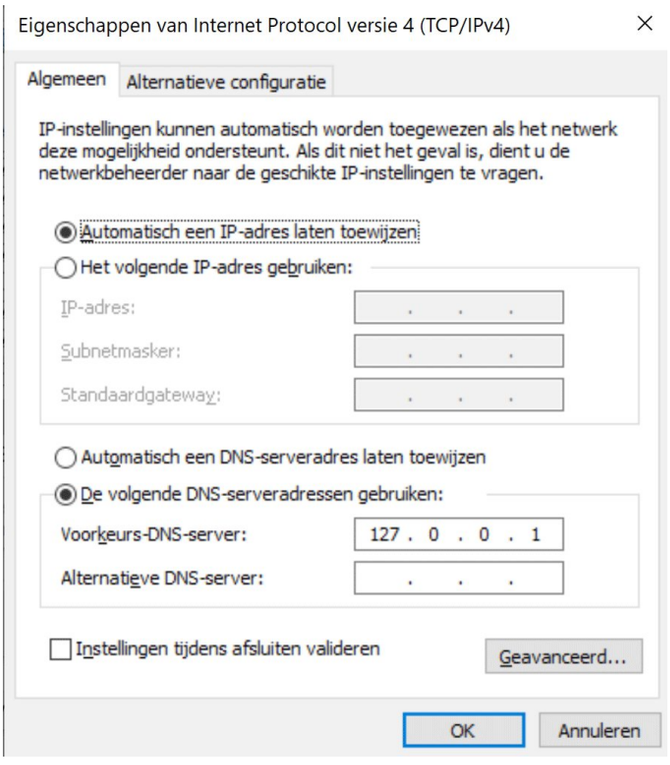
Microsoft heeft aangekondigd dat zij DoH en DoT gaan ondersteunen. In de Insider Fast Lane 20H2 is DoH al beschikbaar om te testen. Alle DNS-aanvragen gaan dan via DoH.

Als gebruiker heb je het voordeel dat de standaardpoort 443 wordt gebruikt. Iedere firewall zal die doorlaten. Helaas is de ondersteuning voor DoH niet in de reguliere versie 21H1 gekomen. Als u niet wilt wachten tot dit beschikbaar komt, is de enige mogelijkheid een extra programma te installeren. Goede kandidaten daarvoor zijn **Simple-DNSCrypt, Stubby en Unbound.**

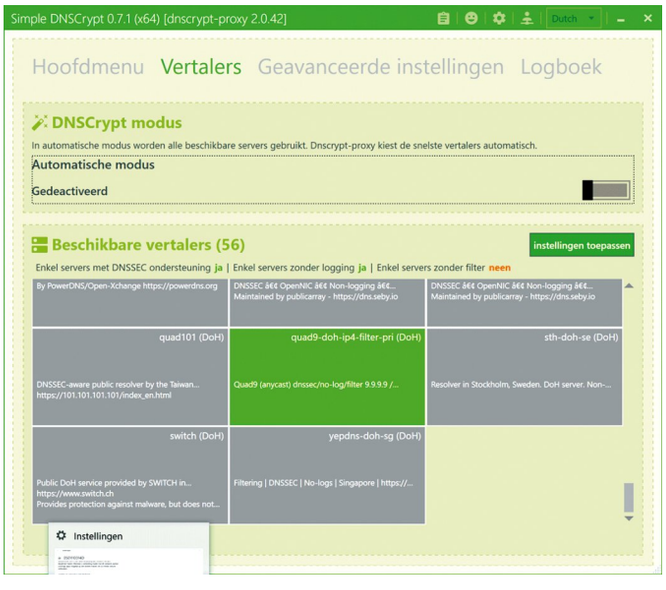
Simple-DNSCrypt (link 5) heeft mijn voorkeur omdat het eenvoudig is te configureren en, net als straks in Windows10, DoH ondersteunt.
Oorspronkelijk werd alleen DNSCrypt ondersteund, maar sinds versie 2.0 ook DoH. DNSCrypt is een ander protocol voor versleutelde autorisatie met DNS-servers. Het heeft het echter nooit gebracht tot een officiële norm. Er zijn echter wel DNS-servers die het ondersteunen.

Haal het programma op van de website en installeer het. Na de installatie gaan we configureren. In het Hoofdmenu halen we de vinkjes weg bij *Enkel servers zonder filter* en bij **DNSCrypt**. Daarna klikken we op de gewenste netwerkkaart en op *Toon verborgen netwerkkaarten*.





Bij het menu *Vertalers* kiezen we uit de lijst beschikbare vertalers (resolvers) **Quand9**.



Als alles goed is gegaan bij de netwerkkaart, is nu als DNS-server 127.0.0.1 ingesteld. Check dit door te gaan naar *Alle instellingen* > *Netwerk en internet* > *Eigenschappen*. Bij IP4-DNS-servers moet nu 127.0.0.1 staan.

Voor Wireless-adapters is het beter om het te koppelen aan de adapter; dan werkt het ook bij ieder netwerk waarmee u verbindt. Daarvoor kiest u *Adapteropties wijzigen*. Op de betreffende adapter klikt u met de rechter muisknop en kiest *Eigenschappen*. Selecteer *Internet Protocolversie 4* en klik op *Eigenschappen*.
Nu kunt u als voorkeurs-DNS-server 127.0.0.1 instellen.
Als we willen testen of het werkt, kiezen we bij **Simple-DNSCrypt** op het tabblad *Vertalers Cloudflare 1.1.1.1* i.p.v. **Quad9** en gaan we met de browser naar:

https://www.cloudflare.com/ssl/encrypted-sni/

Vergeet niet het na de test weer op **Quad9** of de door u gekozen DNS-server te zetten.

Simple-DNSCrypt heeft een aantal interessante opties:

- logboek waarin de aanvragen van de domeinnamen te zien zijn;
- zwarte en witte lijst voor domeinen en adressen.

Als Simple-DNSCrypt geen versleutelde DNS kan vinden, valt het terug op poort 53. Standaard zijn hiervoor **Quad9** en **Google** ingesteld. Ik raad aan de big-dataverzamelaar **Google** (8.8.8.8:53) te verwijderen.

Conclusie

Voor mobiele apparaten die buitenhuis worden gebruikt, zoals smartphones en laptops, is een veilige verbinding met internet belangrijk.
Bij het gebruik van openbare netwerken is het belangrijk om de *man-in-the middle* te vermijden.

De hier beschreven methodes maken uw verbindingsopbouw veiliger. De benodigde functies zijn redelijk makkelijk te configureren. Er is dus geen reden om dit na te laten.

Als u thuis werkt via wifi, raad ik aan ook uw systeem te configureren voor het gebruik van een versleutelde DNS.
Op dit moment is het aantal DNS-servers die deze functies ondersteunen beperkt. Het is te hopen dan het gebruik ervan snel standaard wordt bij alle internetserviceproviders.

Links

1. https://www.privacy-tools.nl/providers/dns-domein-providers/
2. https://dnscrypt.info/public-servers/
3. https://www.quad9.net/
4. https://blog.mozilla.org/security/2021/01/07/encrypted-client-hello-the-future-of-esni-in-firefox/
5. https://www.simplednscrypt.org/