MINI-GIDS

PCM

VOOROP IN TECH

EXPERTGIDS ENCRYPTIE

WAT IS DE ZIN VAN ENCRYPTIE? » WORKSHOP LET'S ENCRYPT »
IS WHATSAPP WEL VEILIG?

Wat moet ik beveiligen en hoe?

EVELINE MEIJER

ENCRYPTIE

HET VERSLEUTELEN VAN BERICHTEN DEDEN ZELFS DE OUDE EGYPTENAREN AL. TEGENWOORDIG DOEN WE DIT VOORAL DIGITAAL. LOGISCH, WANT STEEDS MEER VAN ONZE GEVOELIGE INFORMATIE IS DIGITAAL. MAAR WAAROM IS ENCRYPTIE NOG GEEN STANDAARD BIJ IEDER BEDRIJF? EN VOOR WELKE VORM VAN VERSLEUTELING KUN JE HET BESTE KIEZEN?

Het versleutelen van berichten en andere data lijkt tegenwoordig zo vanzelfsprekend. Maar dat is het helemaal niet. Zo voegden WhatsApp en Facebook Messenger dit jaar pas end-to-end-encryptie aan hun applicaties toe. Daarnaast vraagt bijvoorbeeld de Iomega-nas van Lenovo niet eens om een wachtwoord, bleek in een uitzending van Zembla in november 2016. Daardoor staat al je data dus onbeveiligd op het internet en iedereen die het ip-adres in handen heeft, kan daarbij. Dat bleek onder meer het geval bij een medewerkster van Europol die deze schijf gebruikte, waardoor er tal van gevoelige documenten op straat lagen.
Dat encryptie nog niet zo vanzelfsprekend is als het moet zijn, komt onder meer doordat er een aantal fabels zijn op dit gebied. Zo wordt er soms gedacht dat het versleutelen van internetverkeer en data een computer vele malen trager maakt. Maar de meeste computers van tegenwoordig gebruiken hun volledige capaciteit niet eens. De extra kracht die ze voor encryptie nodig hebben, halen ze niet weg uit je andere processen. In de praktijk merk je er dus niets van.
Daarnaast wordt er soms gedacht dat cybercriminelen niets te halen hebben, maar zoals velen weten zijn deze mensen geïnteresseerd in vrijwel iedere soort data. Met slechts een klein beetje gevoelige informatie – bijvoorbeeld klantgegevens – kunnen ze immers al veel bereiken. Vrijwel iedereen heeft tegenwoordig informatie die niet op straat mag komen te liggen en cybercriminelen maken er gretig gebruik van. Een reden dat velen van ons versleuteling gebruiken of willen gebruiken. En ook de overheid bemoeit zich daar tegenwoordig mee via de Meldplicht Datalekken. Deze wetgeving bepaalt dat een bedrijf het direct moet melden als er data uitlekken of gestolen worden. Blijkt dat het bedrijf niet zo veel mogelijk gedaan heeft om het lek te voorkomen – bijvoorbeeld alles versleutelen – dan moet er een flinke boete betaald worden. Maar het is ook voor een eindgebruiker prettig om te weten dat bestanden versleuteld zijn, al is het maar voor het geval er iets gebeurd. Maar hoe pak je dit nu aan?

“MET SLECHTS EEN KLEIN BEETJE **GEVOELIGE INFORMATIE – BIJVOORBEELD KLANTGEGEVENS –** KUNNEN HACKERS AL VEEL BEREIKEN”

VERSCHILLENDE VORMEN

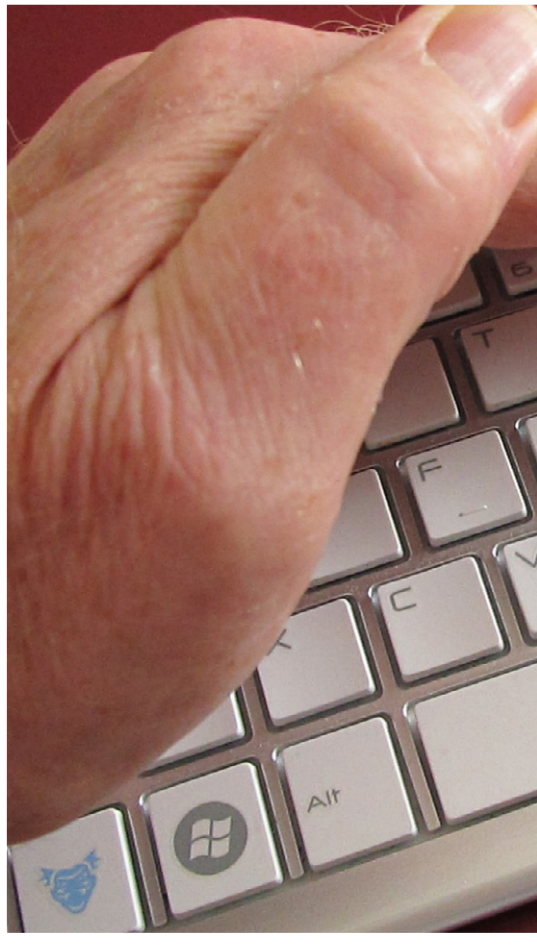
Het versleutelen van data en internetverbindingen kan op verschillende manieren. Veel mensen kennen vooral aes-256, een subversie van aes. Dit algoritme versleutelt bestanden in blokken van 128 bits, wat sneller is dan het bit voor bit versleutelen. Het getal 256 slaat op de lengte van de sleutel, die bij aes 128 bits, 192 bits of 256 bits lang kan zijn. Theoretisch gezien is een langere sleutel veiliger, omdat een computer meer tijd nodig heeft om deze te raden. Dit raden heet in het Engels brute forcing. Wordt er een sleutel van 256 bits gebruikt, dan zijn er 2^256 (2 tot de macht 256) mogelijke combinaties binnen de sleutel. En dat kunnen huidige computers domweg niet kraken, zelfs niet als gigantisch veel computers zouden samenwerken.
Maar er zijn meer soorten encryptie, die ieder weer voor een ander onderdeel doeleinde gebruikt worden. Waar aes vooral nuttig blijkt voor het versleutelen van bestanden op een computer, wordt de standaard rsa veel gebruikt bij het opzetten van een https-verbinding. Hierbij wordt een publieke sleutel gebruikt van de ontvanger om een bericht te versleutelen. Dit wordt eerst gedaan aan de hand van een omkeerbaar en niet-geheim protocol, bijvoorbeeld A=1 en B=2. Daarna worden de data aan de hand van een zogenaamde ontsleutelingsrelatie (C = n^e mod N) nogmaals omgezet, dit

De encryptiemethode aes is vooral geschikt voor het versleutelen van een computer

keer in onleesbare tekst. Die data worden vervolgens verstuurd en door de ontvanger weer omgezet in leesbare tekst met een eigen private sleutel.
Een laatste standaard is sha, dat onder meer gebruikt wordt om wachtwoorden op te slaan. Hierbij wordt een stuk tekst omgezet in een berichtensamenvatting van maximaal 1600 bits. Bijzonder hierbij is echter dat de versleutelde tekst nooit meer teruggezet kan worden naar de oorspronkelijke data. Maar versleutel je dezelfde tekst opnieuw, dan is de uitkomst altijd hetzelfde. In het geval van wachtwoorden slaat de computer dus deze samenvatting – of hash – op. Vul je het wachtwoord vervolgens in, dan versleutelt de computer het opnieuw en controleert hij of de uitkomst hetzelfde is als de opgeslagen variant.
Er zijn nog veel meer verschillende mogelijkheden binnen encryptie, maar deze drie zijn de huidige standaarden die door vrijwel iedereen – inclusief software en bedrijven – worden gebruikt. Hierbij geldt dat de een niet beter is dan de ander. Ze hebben gewoon verschillende doeleinden en werkwijzen. Dit heeft te maken met de vorm van encryptie waar ze binnen vallen: symmetrisch, asymmetrisch en hashing. Welke vorm je het beste kunt gebruiken, hangt af van wat je wilt bereiken.

SYMMETRISCH EN ASYMMETRISCH

Wil je weten hoe je data het beste kunt versleutelen, dan is het goed om te weten welke verschillende vormen van encryptie er zijn. Een zo'n variant is dus symmetrische encryptie. Deze vorm van versleuteling vereist dat je vooraf een sleutel met de ontvanger uitwisselt. Al je data worden namelijk met die sleutel van leesbare tekst omgezet naar onleesbare tekst, en alleen met dezelfde sleutel kun je dit weer terugdraaien. Die sleutel is vaak een reeks data, die het beste werkt als hij volledig willekeurig is. Het ingewikkeldste van symmetrische encryptie is dat de sleutel ook ergens opgeslagen moet worden, en alleen beschikbaar mag zijn voor de software die de sleutel nodig heeft. Het bekendste voorbeeld is het gebruik van een wachtwoord op de computer als je daar alles versleutelt: met de juiste combinatie van letters en cijfers kun je zelf de computer in, maar een ander die de sleutel niet kent, kan dat niet. Groot nadeel is natuurlijk dat als een ander deze sleutel wel in handen krijgt, de gehele beveiliging geen nut meer heeft.
Symmetrische encryptie wordt met name gebruikt voor services die versleutelde data opslaan voor een gebruiker (denk aan een back-up in de cloud), waarbij de sleutel in de handen van de gebruiker blijft. Een ander doel is het versleutelen van een computer of een opslagapparaat, of een veilig kanaal opzetten tussen twee end-points in een netwerk, mits er een veilige alternatieve route is om de sleutel uit te wisselen.
Asymmetrische encryptie doet ongeveer hetzelfde: het pakt data, maakt deze onleesbaar en maakt ze op verzoek en met de juiste sleutel weer leesbaar. Groot verschil is echter dat de sleutel van de ontvanger niet dezelfde is als die van de zender, waardoor er dus geen wachtwoord of andere sleutel verstuurd hoeft te worden. De data worden namelijk onleesbaar gemaakt met een publieke sleutel (public key), en een ontvanger kan zijn eigen private key gebruiken om het weer leesbaar te maken. De publieke sleutel mag dan ook openbaar gemaakt worden, wat eigenlijk zelfs de bedoeling is. De private key houd je echter voor jezelf, net als een wachtwoord. In sommige gevallen kan asymmetrische encryptie het ook toelaten om data te laten ondertekenen. Met de private key wordt in dat geval een handtekening gemaakt, de public key wordt daarna gebruikt om deze te verifiëren. Natuurlijk zijn er ook bij asymmetrische versleuteling nadelen. Via een zogenaamde man-in-the-middle-aanval is het namelijk mogelijk om in te breken op een versleutelde verbinding met deze techniek. In dat geval krijg je een publieke sleutel om een 'veilige verbinding' op te zetten, maar in werkelijkheid wordt er gecommuniceerd met een geheel andere partij. Deze partij geeft jou hun eigen publieke sleutel, en geeft degene waar je mee wilde communiceren een geheel andere publieke sleutel en doet alsof die van jou is. De data die jij verstuurt naar de andere kant kunnen zij vervolgens lezen en



"MET ASYMMETRISCHE ENCRYPTIE IS HET **MOGELIJK OM EEN BEVEILIGDE VERBINDING OP TE ZETTEN** MET SERVERS DIE OP AFSTAND STAAN"

opnieuw versleutelen voor ze deze doorsturen. Ze onderscheppen dus het bericht en kunnen meelezen. Vervelend als je bijvoorbeeld je bankgegevens aan het invoeren bent. Het enige wat je hiertegen kunt doen is er zeker van zijn dat je de juiste publieke sleutels hebt.

Asymmetrische encryptie wordt in de praktijk vooral gebruikt op het internet, bijvoorbeeld in de ssl-certificaten om een beveiligde verbinding (de https-verbinding) op te zetten tussen een browser en een website. Daarnaast is het hiermee mogelijk om een beveiligde verbinding op te zetten met servers die op afstand staan. Een computer gebruikt deze vorm van versleuteling ook als er software-updates zijn en hiervoor getekend moet worden. Daardoor weet het systeem namelijk zeker dat de software van een vertrouwde partij afkomstig is.

HASHING

Hashing is eigenlijk geen encryptie, maar een algoritme dat data zo erg door elkaar schudt dat het niet langer mogelijk is om te zien wat de originele data waren. Maar waar het bij encryptie mogelijk is om data ook weer leesbaar te maken, is dit bij hashing niet het geval. Eenmaal onleesbaar gemaakt, dan blijft het ook zo. Wat er wel gedaan kan worden is de originele gegevens opnieuw laten omzetten met hetzelfde algoritme. Komen de eerste en de tweede uitkomst overeen, dan weet je dat het om dezelfde data gaat. Het is namelijk niet mogelijk om dezelfde 'hash' uit twee verschillende stukken data te krijgen. De uitkomst van een soort data is altijd hetzelfde.

Het voornaamste doeleinde van hashing is het beschermen van wachtwoorden. Vul je dus een password in op je computer, dan controleert het systeem of de versleutelde uitkomst hetzelfde is met degene die opgeslagen staat. Het is echter niet mogelijk om de versleutelde uitkomst te gebruiken om binnen te dringen, en daarmee zien we meteen het voordeel van deze techniek. Steelt een hacker namelijk de hash op een computer, dan kan hij er nog niets mee. Hashing wordt onder meer gebruikt om wachtwoorden veilig op te slaan op een computer.

Encryptiestandaarden

Er zijn verschillende vormen van encryptie, en iedere vorm heeft weer een aantal standaarden. Aes is bijvoorbeeld een bekende standaard voor symmetrische encryptie. Maar er zijn ook andere standaarden binnen de symmetrische variant, bijvoorbeeld Twofish dat rond dezelfde tijd werd ontwikkeld als aes. Het grootste verschil tussen de twee is dat ze een ander netwerk gebruiken om dezelfde taak uit te voeren. Twofish is niet te breken, zelfs niet theoretisch, terwijl aes in sommige gevallen theoretisch gezien wel te kraken is. Dit is tot nu toe echter nog niet gebeurd. De voornaamste reden dat we aes gebruiken en Twofish veel minder, is omdat aes veel efficiënter is als het om hardware gaat. Deze standaard heeft minder geheugen nodig en minder tijd om data te versleutelen.

Op het gebied van asymmetrische encryptie wordt vooral rsa veel gebruikt. Net als aes is dit een standaard. Bij rsa maakt en publiceert een gebruiker een publieke sleutel gebaseerd op twee grote priemgetallen, samen met een hulpwaarde. De priemgetallen houd je geheim, omdat je met die cijfers en kennis van priemgetallen het bericht alsnog kunt kraken.

Sha is een standaard voor hash, die ontworpen is door de NSA. Met deze vorm van versleuteling wordt er een hash gemaakt van maximaal 1600 bits langs, die meestal weergegeven wordt als een hexadecimaal getal van veertig nummers. Sha wordt onder meer gebruikt bij ssl-certificaten. Op dit moment worden er nog veel sha-1 hashes gebruikt, maar dit is niet aan te raden. Op deze versie kan namelijk ingebroken worden. De nieuwste standaard is sha-3, en die is tot nu toe wel veilig.

WAT MOET IK GEBRUIKEN?

Zeggen dat je iets met aes versleuteld hebt en het daardoor veilig is, is dus iets te kort door de bocht. aes is namelijk een vorm van symmetrische encryptie en dus niet voor ieder doeleinde te gebruiken. In werkelijkheid wordt er bij veel vormen van beveiliging gebruikgemaakt van een combinatie tussen asymmetrische en symmetrische versleutelingen.

Gelukkig is het tegenwoordig vaak niet langer nodig om zelf alle soorten versleuteling uit te zoeken, want er zijn veel programma's en bedrijven die zich hierin specialiseren. Om te beginnen is het goed om je computer te versleutelen. Mocht iemand dan aanvallen, dan treft de hacker alleen versleutelde gegevens

Apple FileVault zit op iedere MacBook en iMac met macOS Lion of een nieuwere versie.

“VERSLEUTEL JE COMPUTERS. EVENTUELE HACKERS **TREFFEN DAN ALLEEN VERSLEUTELDE GEGEVENS** AAN”

aan. Dit kan vrij gemakkelijk met Microsoft BitLocker, mits je een Microsoft-computer gebruikt. Dit programma, dat vanaf Windows 7 in de Enterprise-edities van het besturingssysteem zit, versleutelt de gehele harde schijf op een computer. BitLocker is aan te zetten door naar Verkenner te gaan en met de rechtermuisknop op de map **C:** te klikken. In het menu staat dan **Zet BitLocker aan**. Microsoft vraagt op dat moment om een kopie van de herstelsleutel op te slaan, die nodig is om de versleuteling van de schijf af te halen. De sleutel is op te slaan in een Microsoft-account en in een bestand, maar ook af te drukken. Werk je met een iMac of MacBook, dan is Apple FileVault ook een mogelijkheid om de computer te versleutelen. Dit werkt vrijwel hetzelfde als Microsoft Bitlocker, maar de sleutel wordt alleen opgeslagen in een iCloud-account. Natuurlijk is het ook mogelijk om deze sleutel op te schrijven. Apple FileVault is aan te zetten via het **System Preferences**-menu onder **Security & Privacy**. Hier staat een **FileVault**-tabblad, waar je de computer kunt vergrendelen. Daarna moet de computer opnieuw worden opgestart.

In het geval van Linux wordt de harde schijf meestal versleuteld tijdens de installatie van het besturingssysteem, bijvoorbeeld met een tool als dm-crypt.
Daarnaast zijn er ook nog programma’s van derde partijen die een harde schijf kunnen versleutelen. Bekende voorbeelden zijn TrueCrypt, VeraCrypt en DiskCryptor. Daarnaast zijn er partijen die anti-virussoftware verkopen en direct encryptie in hun producten stoppen. Dit doen onder meer Symantec, Kaspersky, Sophos en ESET.

Usb apart versleutelen

Het is goed om te weten dat als er een bestand van een versleutelde computer naar een usb-stick gekopieerd wordt, het bestand in sommige gevallen niet langer versleuteld is. Daarom is het van belang om ook een usb-stick te versleutelen, wat onder meer kan met Microsoft BitLocker To Go of door een usb-stick met encryptie te kopen.

WEBSITE BEVEILIGEN

Tegenwoordig is het niet erg ingewikkeld meer om een website met een https-verbinding te beveiligen. Wat wel nodig is, is een ssl-certificaat, dat door verschillende partijen wordt aangeboden. Vaak kan dit ook via het bedrijf dat de website host. Voordeel daarvan is dat deze partij er ook vaak voor zorgt dat het certificaat op de juiste wijze geïnstalleerd wordt, waardoor er eigenlijk geen werk meer aan is. Daarnaast biedt Let’s Encrypt gratis ssl-certificaten aan. Iedereen met een domeinnaam kan hier gebruik van maken en het voordeel is dat het gemakkelijk te installeren is. De partij heeft namelijk software die je op een webserver zet, waarmee je kunt praten met Let’s Encrypt. Het installeren gaat ook via die software en als er updates zijn, worden die automatisch geïnstalleerd. Op de website www.letsencrypt.org wordt uitgelegd hoe dit precies in zijn werk gaat. Een nadeel is wel dat ze maar één soort certificaat uitgeven: de domain-validated certificates. Hiermee wordt echter alleen maar bewezen dat iemand het domein beheert, bijvoorbeeld doordat er gereageerd wordt op berichten naar het e-mailadres dat is

opgegeven. Veiliger certificaten, die bijvoorbeeld bewijzen dat de website van een specifiek bedrijf is, worden niet door Let's Encrypt uitgegeven. En deze certificaten maken gebruik van een andere vorm van encryptie, die veiliger is. Heb je bijvoorbeeld een website waar klanten misbruikgevoelige informatie in moeten vullen, dan is Let's Encrypt dus niet de partij waar je moet zijn en kun je beter een strengere beveiliging toepassen.

WIFI BEVEILIGEN

Het beveiligen van een draadloos netwerk kan tegenwoordig vrij gemakkelijk via de router. Nadeel is echter dat er drie verschillende manieren zijn om dit te doen, waardoor het lastig kan zijn om te besluiten welke er gekozen moet worden. De mogelijkheden zijn wpa2 (de standaard), wpa2-psk en wpa2-ent (Enterprise). Wpa2 gebruikt een aes-versleuteling. De twee andere mogelijkheden zijn ontworpen voor verschillende soorten netwerken. Wpa2-psk is bedoeld voor een netwerk in het eigen huis en voor zeer kleine bedrijven. Hierbij wordt ieder draadloos apparaat geauthenticeerd met dezelfde 256bit-sleutel. Dit betekent dat er een wachtwoord wordt aangemaakt die door iedere gebruiker ingevuld moet worden als hij het netwerk op wil. Wpa2-ent is officieel gezien bedoeld voor grote bedrijven, maar kan voor ieder bedrijfsnetwerk gebruikt worden. Het verschil is dat deze versie een nieuwe sleutel aanmaakt als een gebruiker inlogt met zijn unieke wachtwoord. Daarnaast worden wachtwoorden voor het netwerk zelf niet lokaal opgeslagen. Voordeel van de psk-modus is echter wel dat het eenvoudiger is om in te stellen. Voor wpa2-ent is namelijk ook een RADIUS-authenticatieserver nodig, die bij de psk-modus niet nodig is. RADIUS is vaak een server op het niveau van enterprises, maar er zijn ook wel opties voor kleinere bedrijven. Is er bijvoorbeeld al een Windows Server geïnstalleerd, dan kan er gebruikgemaakt worden van de Internet Authentication Service of de Network Proxy Server. Een andere optie is de RADIUS-server bij een hostingpartij neer te zetten, waarbij zij hem installeren en jij die moeite niet hoeft te doen. Maar voor welke van de opties er ook gekozen wordt, ieder apparaat dat verbinding wil maken met het netwerk moet in dezelfde modus staan.

Https

Bij https wordt de beveiligde verbinding opgebouwd met een 'handshake', die bestaat uit een combinatie van soorten encryptie. De handshake begint met een bericht vanuit de browser met daarin alle informatie die de server nodig heeft om een ssl-verbinding op te zetten. De server reageert vervolgens met eenzelfde soort bericht. Vervolgens gaat de server bewijzen dat hij echt is wie hij zegt dat hij is, en dat gebeurt met een ssl-certificaat. Hierin staat bijvoorbeeld de openbare sleutel, de digitale handtekening en informatie over de eigenaar. De browser controleert vervolgens of alle informatie klopt en of het certificaat echt is. En pas dan wordt er een sleutel uitgewisseld, namelijk die voor de symmetrische encryptie. Simpel gezegd wordt er dus eerst gebruikgemaakt van een asymmetrische versleuteling om een veilige verbinding op te zetten, zodat de sleutel voor de symmetrische verbinding veilig kan worden uitgewisseld.

ALLES OF NIETS

Wil je jezelf op digitaal gebied goed beveiligen, dan is het van belang om op ieder punt gebruik te maken van encryptie. Zoals gezegd verschilt het per onderdeel welke vorm van encryptie er gebruikt wordt (bijvoorbeeld aes-256 bij het versleutelen van een computer) en in veel gevallen kan er zelfs een combinatie gemaakt worden van de soorten encryptie. Maar slechts een klein onderdeel versleutelen is nooit goed genoeg, aangezien een hacker altijd wel een manier vindt om precies bij de onbeveiligde data uit te komen en deze te stelen. Gelukkig is het met de jaren steeds eenvoudiger geworden om data te versleutelen. Er zijn talloze bedrijven die zich hierin specialiseren en software aanbieden om het proces te versimpelen. Het beste is dan ook om te beginnen bij het begin: bedenk wat er allemaal versleuteld moet worden en begin boven aan dat lijstje. Het kost misschien een dag werk, maar het resultaat is dat een cybercrimineel het een stuk moeilijker zal vinden om bij je gevoelige informatie te komen. En dat is een prettig gevoel. «

"SLECHTS EEN KLEIN ONDERDEEL VERSLEUTELEN IS NOOIT GOED GENOEG"

WHATSAPP IS WÉL VEILIG!

WHATSAPP IS ONDERDEEL VAN FACEBOOK, EN ALS JE SOMMIGE DOEMDENKERS MOET GELOVEN MAAKT DAT DE CHAT-APP EEN RECHTSTREEKSE SPREEKBUIS NAAR HET SOCIALE NETWERK EN DE NSA. WIE ECHTER ZEGT DAT WHATSAPP NIET VEILIG GENOEG IS, OVERDRIJFT EN MAAKT HET GEBRUIKERS NODELOOS MOEILIJK.

Er is veel lof voor veilige chat-apps zoals Signal, zeker sinds we weten dat de NSA ons op grote schaal afluistert. Door de overname van WhatsApp door Facebook vroeg iedereen zich in 2014 (terecht) af of al die intieme berichtjes nog wel echt tussen jou en de ontvanger bleven. Signal was een van de eerste chat-apps die garandeerde dat je veilig blijft. En toen Telegram een paar jaar geleden in opkomst was, werd de chat-app door fans de hemel in geprezen voor zijn focus op privacy en veiligheid van de gebruiker. Dat was voordat de zelfgebouwde cryptografie van Telegram zo lek als een mandje bleek te zijn.
Die wanhopige drang naar superveilige en hyperanonieme chat-apps is echter behoorlijk overdreven. Toen WhatsApp in 2014 voor een recordbedrag door Facebook werd overgenomen, was de ophef groot. Het bedrijf dat grof verdiende met het aanleggen van profielen van zijn miljard gebruikers kreeg ineens de app in handen waarmee 900 miljoen mensen iedere dag de meest persoonlijke berichten naar elkaar stuurden. Was dat wel verenigbaar met elkaar? De beloofde massale exodus van gebruikers bleef uit, al wist het 'veilige' Telegram rond die tijd flink wat nieuwe gebruikers te werven.

GOEDE VERSLEUTELING

Uiteindelijk bleek die angst ongegrond. Sinds een jaar na de overname voegt WhatsApp end-to-end-encryptie toe aan alle berichten die je verstuurt. Aanvankelijk ging het alleen om berichten die onderling via Android-toestellen werden verstuurd, later werden daar iOS en groepsgesprekken aan toegevoegd en inmiddels is er geen enkel berichtje meer dat niet wordt versleuteld. Het gaat bovendien niet om de minste encryptie: WhatsApp maakt daarvoor gebruik van dezelfde versleuteling als Signal: een axolotl-protocol met AES-256-algoritme dat nagenoeg onkraakbaar is.

GEBRUIKSONVRIENDELIJK

Die encryptie is meer dan goed genoeg voor 99 procent van alle gebruikers. Waarom zijn er dan nog steeds zo veel gebruikers die roepen dat WhatsApp niet goed genoeg is? Is het omdat moederbedrijf Facebook nog steeds metadata kan aflezen en je contactenlijst te zien krijgt? Dat feit als een even groot risico bestempelen als het lezen van de daadwerkelijke inhoud van berichten is niet alleen naïef, maar ook gebruiksonvriendelijk. WhatsApp heeft bijna een miljard gebruikers, Signal significant minder. De app verbindt miljoenen mensen met elkaar, zelfs mensen die normaal nooit aan de smartphone wilden maar nu juist meer contact hebben met hun familie of vrienden. Is het dan realistisch om tegen gebruikers te zeggen dat ze moeten stoppen met WhatsApp?
Encryptie werkt. Dat bleek in maart wel weer, toen WikiLeaks een grote hoeveelheid informatie over de CIA naar buiten bracht. Wat bleek? De buitenlandse inlichtingendienst van de VS worstelde met de versleuteling van chat-apps. Niet alleen Signal was onkraakbaar, maar ook WhatsApp. Om de apps af te luisteren, moest de spionagedienst inbreken op het besturingssysteem van een verdachte … een peperduur en intensief karwei. Dat is hoe encryptie moet zijn én wat het gelukkig nu is: onkraakbaar, tenzij je heel specifiek achter één toestel aan gaat. «

BEVEILIG JE WEBSITE MET HTTPS

DE EERSTE WORKSHOP OP DE VOLGENDE TWEE PAGINA'S VORMT DE BASIS VOOR DRIE ANDERE HANDELINGEN. ZO KUNNEN WIJ JE HEEL UITGEBREID EN COMPLEET VERTELLEN OVER EEN PRODUCT OF DIENST EN KIES JE ZELF MET WELKE ONDERDELEN JE AAN DE SLAG GAAT. JE KIEST NA DE EERSTE WORKSHOP, ÉÉN OF MEERDERE ONDERDELEN DIE JE WILT GAAN VOLGEN.

PAGINA 58

OVERAL HTTPS MET LET'S ENCRYPT!

PAGINA 60

PAGINA 62

INSTAL-LEER ZELF JE CERTI-FICAAT

OPTIMALISEER JE SERVER-OMGEVING

PAGINA 64

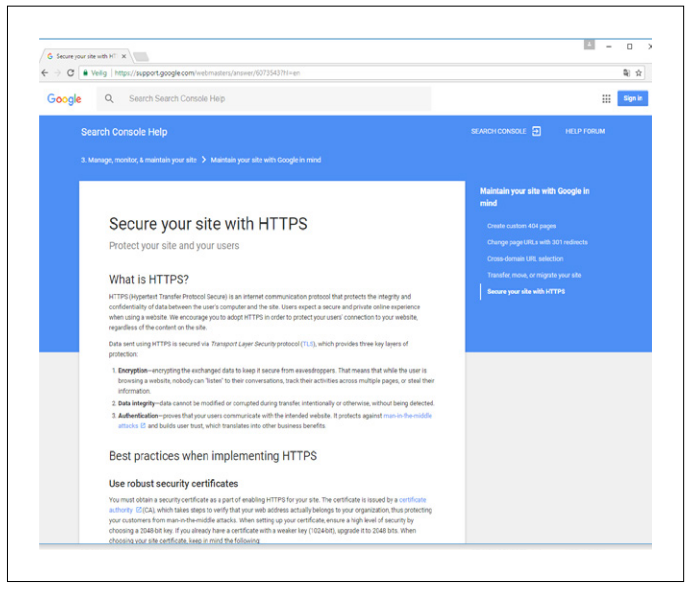
Let's Encrypt maakt het gratis én eenvoudiger

GERTJAN GROEN

BEVEILIG JE WEBSITE **MET HTTPS**

NIET ALLEEN COMPLEXE WEBSITES MET VEEL PRIVÉGEGEVENS, MAAR OOK EENVOUDIGE BLOGS VERDIENEN EXTRA BESCHERMING VIA HTTPS. DANKZIJ HET GEAUTOMATISEERDE PROCES VAN LET'S ENCRYPT KAN DIT KOSTELOOS EN ZONDER AL TE VEEL MOEITE. IN DEZE WORKSHOPSERIE LEES JE HOE JE DIT VOOR JE WEBSITE ACTIVEERT.

1



Google maakt zich sterk voor https-versleuteling.

Steeds meer websites schakelen over naar versleutelde verbindingen via https. Het is een stuk veiliger voor bezoekers maar ook voor jezelf als beheerder van een pagina. Het 'gewone' http-verkeer kan immers gemakkelijk afgeluisterd en gemanipuleerd worden. Bij https 'praat' de browser vrijwel zeker rechtstreeks met de webserver. Al het verkeer gaat door de veilige tunnel. Wie denkt dat een eenvoudige blog geen bescherming behoeft heeft het mis. Zo zijn er bijvoorbeeld partijen, waaronder enkele Amerikaanse internetproviders, die advertenties in websites 'injecteren'. Bij https worden ze buitenspel gezet.

GOOGLE VOORSTANDER VAN HTTPS

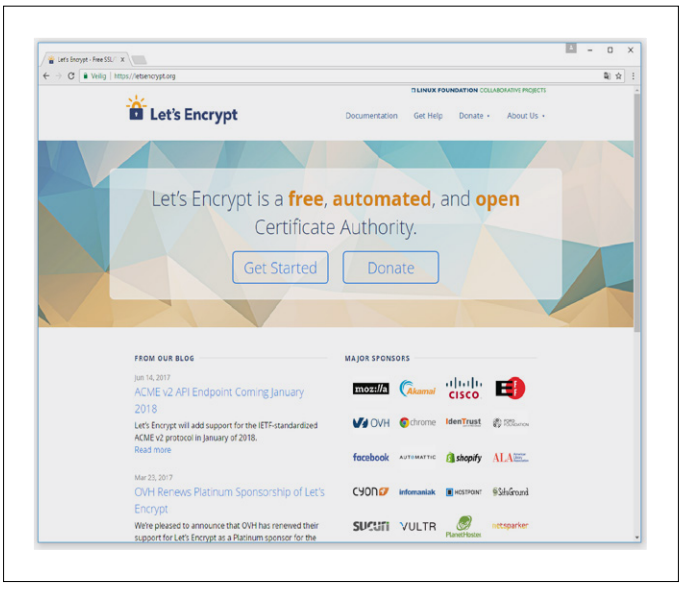
Google doet er alles aan om websitebouwers te bewegen over te stappen naar https. Zo heeft de populaire Chrome-browser een duidelijke voorkeur voor https-verkeer (zie kader volgende pagina). Ook beloont Google je met een betere ranking, al weegt het op dit moment nog niet zwaar mee in de resultaten. Het bedrijf stond ook aan de basis van het nieuwe internetprotocol http/2 dat voor een flinke snelheidswinst zorgt. Je hoeft ook niet bang te zijn voor hoge kosten. Het populaire Let's Encrypt verstrekt gratis certificaten die je via een tool bovendien zeer eenvoudig kunt installeren, activeren en vernieuwen. Genoeg redenen dus om voor het felbegeerde groene slotje te zorgen.

CERTIFICATEN

Er bestaan diverse certificaten. De eenvoudigste is een self-signed certificaat. Voor testdoeleinden is het een prima optie. Zo'n certificaat is ondertekend door de persoon die hem heeft gemaakt en niet door een vertrouwde autoriteit. Browsers geven daardoor een waarschuwing. Dit voorkom je met een certificaat dat is ondertekend door een certificaatautoriteit (ca). Deze uitgever moet door de browser als betrouwbaar worden beschouwd, anders verschijnt alsnog een waarschuwing. Een browser wordt standaard met een aantal certificaten geleverd. Het is een hiërarchische structuur of vertrouwensketen, die begint bij een root-certificaat en via tussenliggende of intermediate-certificaten uiteindelijk uitkomt bij jouw eigen certificaat. Naast commerciële aanbieders als Thawte en Symantec zijn er ook partijen die gratis certificaten verstrekken, waaronder Let's Encrypt dat in deze workshops centraal staat. Het heeft een eigen root-certificaat, ISRG Root X1, dat op het moment alleen nog in de laatste versie van Firefox is opgenomen. Het duurt normaliter zo'n drie tot zes jaar voordat een nieuw root-certificaat wordt vertrouwd en populaire browsers hiervan zijn voorzien. Tot die tijd zijn de certificaten die het uitgeeft gebaseerd op een tussenliggend certificaat dat is ondertekend door partner IdenTrust, dat reeds een aantal vertrouwde root-certificaten bezit (waaronder DST Root CA X3).

VERSCHILLENDE CERTIFICATEN

Zoek je een ondertekend certificaat, dan is een Domain Validated-certificaat (dv-certificaat, ook wel DomainSSL genoemd) de eenvoudigste en goedkoopste optie. Ze zijn geschikt voor iedere server die via een domeinnaam werkt, waaronder natuurlijk een webserver.



Let's Encrypt deelt gratis ssl-certificaten uit.

Chrome-browser heeft sterke voorkeur voor https

De Chrome-browser markeert het 'gewone' http-verkeer steeds nadrukkelijker als onveilig. Als zo'n pagina om een wachtwoord of creditcard vraagt dan staat er zelfs nadrukkelijk bij dat de verbinding niet veilig is. Het bedrijf zal deze waarschuwingen steeds verder uitbreiden. Zo zal op termijn zelfs een rood kruis verschijnen bij onbeveiligd verkeer. Dit kruis zie je nu alleen bij onvolledige https-implementaties. Verder werken sommige features om privacyredenen alleen nog via https. Een goed voorbeeld is de Geolocation-api waarmee een website de positie van de bezoeker kan opvragen.

Het groene slotje wordt steeds belangrijker voor het vertrouwen van bezoekers.

Zoals bij ieder certificaat moet je via een validatieproces aantonen dat je er recht op hebt. Bij dit type certificaat is dat eenvoudig: er is alleen een controle nodig dat je de domeinnaam bezit. Dat kan eenvoudig en zelfs automatisch, zoals Let's Encrypt laat zien. Naast dv-certificaten zijn er ook andere certificaten, waarvan Extendended Validation-certificaten (ev-certificaten) de interessantste zijn. Daarbij wordt de identiteit van de aanvrager uitvoeriger gecontroleerd. Als bonus wordt ook de bedrijfsnaam in de adresbalk getoond achter het groene slotje.

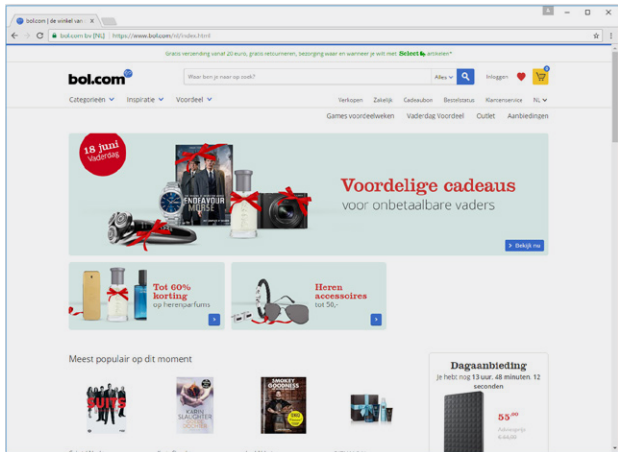
AANVRAAGPROCES VOOR CERTIFICATEN

Het normale proces voor het aanvragen van een certificaat is vrij omslachtig. Het begint met het genereren van een sleutelpaar. Als sleutelgrootte is 2048 bit momenteel de standaard en ook meer dan toereikend. Het sleutelpaar dat je aanmaakt omvat een privésleutel en publieke sleutel. De privésleutel houd je geheim en blijft op de server staan. De publieke sleutel wordt samengevoegd in een csr (Certificate Signing Request) samen met enkele andere gegevens over je organisatie en de domeinnaam. Die csr creëer je op de server en deel je vervolgens (online) met de certificaatautoriteit. Hierna krijg je het uiteindelijke certificaat. Dit certificaat installeer je op de server, in een map die niet toegankelijk is via het web. Een certificaat zegt alleen iets over het vertrouwensniveau. Het staat los van de parameters, zoals het gebruikte algoritme, van de uiteindelijke communicatie. Die parameters worden per verbinding in een 'handshake' afgesproken tussen je server en de browser van de bezoeker. Het is wel belangrijk dat je voor een correcte en veilige configuratie van de webserver zorgt.

WEBSERVER CONFIGUREREN

Heb je het certificaat ontvangen, dan dien je alleen nog je webserver, bijvoorbeeld Apache, in te stellen voor het gebruik van het certificaat. Ook moet je verschillende parameters opgeven voor de configuratie van het protocol Transport Layer Security (tls), voorheen bekend als ssl. Dat protocol is verantwoordelijk voor de versleuteling. Het is een vrij complex protocol en daarmee lastig te configureren. De Mozilla SSL Configuration Generator (**https://mozilla.github.io/server-side-tls/ssl-config-generator**) kan een goede hulp zijn. Je geeft aan wat voor webserver (bijvoorbeeld Apache of Nginx) en software je gebruikt, waarna je een werkende configuratie krijgt. De hele overstap van http naar https is al met al wel vrij gecompliceerd, wat vermoedelijk de belangrijkste reden is dat nog steeds veel websites het zonder die versleuteling doen. Gelukkig maakt Let's Encrypt het een stuk eenvoudiger.

Bij een Extended Validation-certificaat staat de bedrijfsnaam achter het groene slotje.



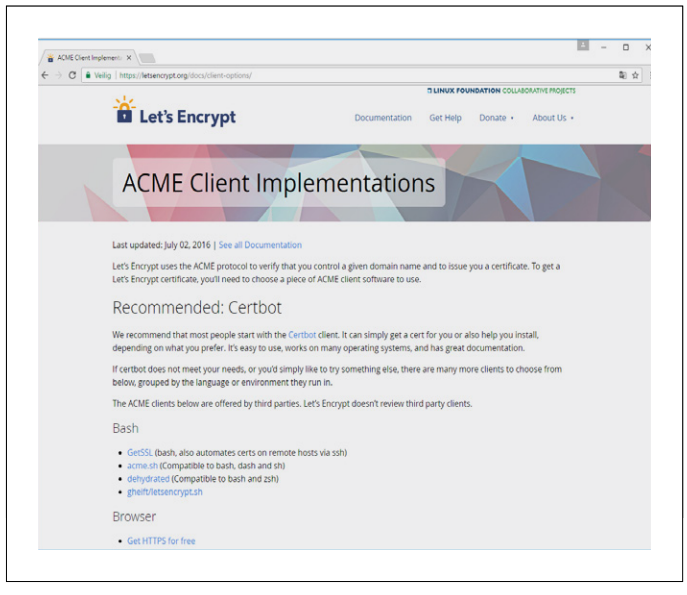
Geautomatiseerd proces neemt werk uit handen

GERTJAN GROEN

OVERAL HTTPS MET **LET'S ENCRYPT!**

LET'S ENCRYPT IS NIET HET EERSTE BEDRIJF DAT GRATIS CERTIFICATEN VERSTREKT. MAAR JUIST HET HOGE VERTROUWENSNIVEAU EN GEAUTOMATISEERDE PROCES MAAKT DE CERTIFICATEN GELIEFD ONDER WEBBOUWERS.

Let's Encrypt, een initiatief van onder meer Mozilla, Cisco, Akamai en Facebook, deelt gratis certificaten uit aan de bezitters van een domein. Wie al eens met certificaten heeft gewerkt weet hoe lastig het de aanvraag, installatie en configuratie kan zijn. Er is veel kennis van zaken en tijd voor nodig. Let's Encrypt neemt die klus vrijwel volledig uit handen: een geautomatiseerd proces kan het aanmaken van de certificaten en zelfs de configuratie daarvan in Apache voor je uit handen nemen.



Er zijn meerdere clients voor het acme-protocol van Let's Encrypt.

MEER GRATIS AANBIEDERS

Er zijn meer partijen die gratis certificaten verstrekken, zoals StartSSL van het Israëlische StartCom alsmede het Chinese WoSign, maar die worden door enkele misstanden minder vertrouwd. Bovendien wordt commercieel gebruik bij StartSSL uitgesloten. Het aanvraagproces is ook gecompliceerder. Let's Encrypt lijkt zijn zaken beter voor elkaar te hebben. Het proces van aanvragen is eenvoudig en laat geen ruimte voor geknoei. Als er al bugs zijn worden ze dankzij de transparante bedrijfscultuur openlijk gecommuniceerd.

GEAUTOMATISEERD PROCES

Bij Let's Encrypt installeer je een client op de server die werkt volgens het opensource acme-protocol (Automated Certificate Management Environment). Wij gebruiken de officiële client Certbot, maar op **www.letsencrypt.org/docs/client-options** vind je ook veel alternatieven. De client vraagt een certificaat aan bij de servers van Let's Encrypt. De server stuurt daarna een opdracht door om te bewijzen dat je de eigenaar van het bewuste domein bent. Dat kan een dns-record zijn of een bestand dat via een url op je server beschikbaar moet worden gemaakt. Na verificatie wordt het certificaat opgehaald, geïnstalleerd en wordt Apache geconfigureerd. Ook het vernieuwen kun je gemakkelijk automatiseren zoals we in de vierde workshop laten zien. Dat is belangrijk omdat de certificaten van Let's Encrypt maar drie maanden geldig zijn. In de praktijk biedt het vooral voordelen, ook wat veiligheid betreft. Je kunt het zien als een wachtwoord dat je regelmatig moet veranderen.

LET'S ENCRYPT VERSTERKT **ALLEEN CERTIFICATEN VOOR DOMEINNAMEN,** MAAR JE MAG WEL TOT 100 DOMEINNAMEN HIERIN OPNEMEN

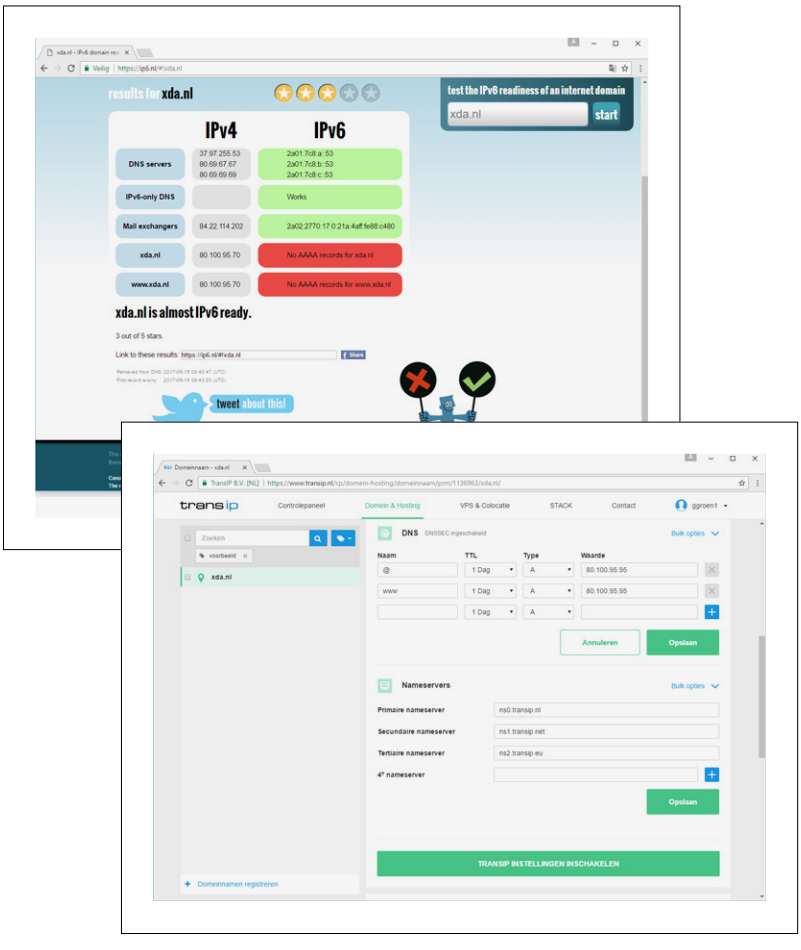
MEERDERE DOMEINNAMEN

Let's Encrypt verstrekt alleen Domain Validated-certificaten

Optimale snelheid met http/2

Het versleutelen van een website kan vertraging geven, maar dat hoeft niet. Google ontwikkelde als onderzoeksproject het verbeterde internetprotocol SPDY dat inmiddels de basis vormt voor versie 2 van http, bekend als http/2. Ongeveer 80 procent van de browsers ondersteunt de nieuwe standaard reeds (zie ook **www.caniuse.com/#feat=http2**). Het grootste voordeel is multiplexing. De bestanden die nodig zijn voor een webpagina, zoals afbeeldingen, css en javascript, kunnen via een veel kleiner aantal verzoeken opgehaald worden. Dat dit heel veel snelheidswinst kan geven zie je als je achtereenvolgens **http://www.httpvshttps.com** en **https://www.httpvshttps.com** bezoekt. Dat is wel een extreem voorbeeld, maar de praktijk laat ook zien dat er veel winst mee valt te behalen. Hoewel het protocol ook voor gewoon http-verkeer geschikt is, blijkt dat in de praktijk geen optie: hedendaagse browsers accepteren alleen https-verkeer via het protocol.

Controleer via bijvoorbeeld ip6.nl de bereikbaarheid via ipv4 en ipv6.



De dns-instellingen voor je domeinnaam moeten correct zijn.

(dv). Heel praktisch is dat één certificaat voor meerdere domeinnamen kan worden gebruikt. Dit heet ook wel Subject Alternative Name (san). In de workshop profiteren we hiervan door zowel domein.nl als www.domein.nl in het certificaat op te nemen. Een wildcard (*.domein) waarmee je direct alle subdomeinen afvangt is helaas niet mogelijk. Wel is de limiet erg ruim: een certificaat kan tot 100 domeinen bevatten. Er zijn wel enkele limieten, die je op **www.letsencrypt.org/docs/rate-limits** kunt nalezen, maar die zijn over het algemeen dusdanig ruim dat je je hier geen zorgen over hoeft te maken.

VEREISTEN VOOR INSTALLATIE

De belangrijkste vereiste om Let's Encrypt te installeren op je server is dat je als root kunt inloggen en het Python-programma kunt uitvoeren. Prettig om te weten is dat je voor het domein dat je achter https gaat zetten geen uniek ip-adres nodig hebt. Bij andere certificaten is dat soms wel een vereiste. Voor de autorisatie heeft Let's Encrypt wel van buitenaf toegang tot de server nodig. De poorten 80 (http) en 443 (https) moeten openstaan, maar die heb je sowieso nodig voor het onbeveiligde en beveiligde webserververkeer. Je kunt de installatie ook op een server in je eigen netwerk testen, maar dan moet je wel die twee poorten doorsturen van buitenaf via je router, met port forwarding. Verder is het bij de configuratie handig als je wat ervaring met Apache hebt. In deze workshop geven we ook veel tips voor deze populaire webserver. We nemen ook meteen de configuratie van http/2 mee, een belangrijke technologie om websites te versnellen (zie kader boven). Daarmee hoeft de overgang naar https geen tragere prestatie te geven. In de volgende workshop laten we zien hoe je dit alles installeert op een server met Ubuntu. Heb je geen eigen server? Steeds meer webhostingpartijen bieden ssl voor weinig of niets als extraatje. Zo is ssl op basis van Let's Encrypt sinds april dit jaar gratis actief voor alle accounts bij hostingprovider Antagonist (**www.antagonist.nl**) dat ook meteen overstapte naar http/2.

DNS-INSTELLINGEN

Nog een vereiste voor de validatie is dat de dns-instellingen voor je website correct zijn, maar dat verschilt niet van een gewone situatie. Het betekent concreet dat voor alle domeinen die je in het certificaat wilt opnemen het A-record naar het ipv4-adres van de server verwijst. Heb je ipv6 geconfigureerd op je server met een verwijzend AAAA-record, zorg dan dat dit correct is. Bij de validatie geeft Let's Encrypt vaak de voorkeur aan ipv6 als het beschikbaar is. Je kunt testen of de server bereikbaar is met bijvoorbeeld `curl -4 http://www.domein.nl` voor ipv4 en `curl -6 http://www.domein.nl` voor ipv6. Een alternatief is een website als **https://ip6.nl**. Het is ook een goede controle als je Let's Encrypt op een server in je eigen netwerk wilt testen. Hoewel we het in deze workshop buiten beschouwing laten, kun je bij Let's Encrypt ook gebruikmaken van dns-validatie (zie kader rechts).

Dns-validatie

Let's Encrypt ondersteunt de mogelijkheid voor dns-validatie. Hierbij kun je door het toevoegen van een txt-record aan de dns-instellingen bewijzen dat je de eigenaar van dat domein bent. Dat klinkt eenvoudig als je vertrouwd bent met dns, maar er zitten wel wat haken en ogen aan. Je zult nog altijd het aanvraagproces via je server moeten doen. Dat is met de dns-validatie gecompliceerder. En als je naast je hoofddomein ook extra domeinen aan een certificaat wilt toevoegen, moet je voor die extra domeinen ook een txt-record aanmaken, om te bewijzen dat je ook daar de eigenaar van bent.

Aan de slag met Ubuntu en Apache

GERTJAN GROEN

INSTALLEREN VAN JE CERTIFICAAT

IN DEZE WORKSHOP LATEN WE ZIEN HOE DE INSTALLATIE EN CONFIGURATIE VAN LET'S ENCRYPT WERKT IN COMBINATIE MET DE POPULAIRE WEBSERVER UBUNTU. WE KIEZEN VOOR EEN RECENTERE VERSIE VAN APACHE MET HTTP/2-ONDERSTEUNING.

We laten zien hoe je een certificaat kunt installeren voor een eenvoudige website die draait onder Apache. Als besturingssysteem gebruiken we een minimale installatie van Ubuntu 16.04 LTS (Xenial Xerus). Verder gaan we in deze workshop uit van Apache als webserver. We beginnen met een verse installatie, maar als je Apache al hebt draaien kun je de stappen aanpassen voor je eigen situatie.

OPTIMALE SNELHEID

We profiteren graag van de extra snelheid van http/2 bij het serveren van pagina's via https. Hiervoor is minimaal Apache 2.4.17 nodig. Hoewel Ubuntu daaraan voldoet bevat het niet de voor http/2 vereiste mod_http2-module, die nog als experimenteel te boek staat. Er is wel een mogelijkheid om die module toe te voegen, maar gezien enkele kwetsbaarheden is het verstandig om een recentere versie van Apache 2.4.x te installeren, bij voorkeur 2.4.25 of hoger. Die bevat ook meteen de mod_http2-module. Dat is waar we deze workshop dan ook mee beginnen.

LAATSTE VERSIE APACHE

Log voor onderstaande stappen in als root, bijvoorbeeld via ssh met Putty. Controleer met de opdracht `apt-cache policy apache2` welke versie van Apache eventueel reeds is geïnstalleerd en welke versies beschikbaar zijn via de huidige bronnen. Om over de laatste versie van Apache te beschikken voegen we een zogenoemde repository van een derde partij toe. Hiervoor zijn enkele tools nodig die je installeert met het commando:

```
apt-get install software-properties-common
python-software-properties
```

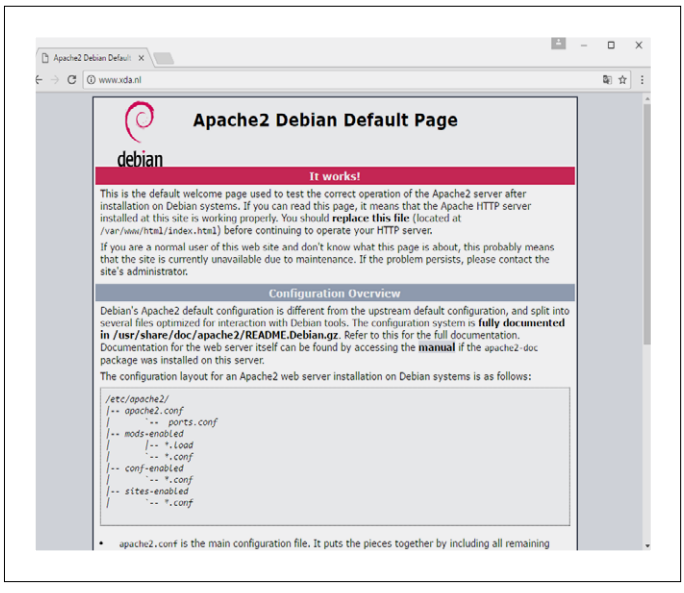
Voeg daarna de repository toe met `add-apt-repository ppa:ondrej/apache2`. Druk op Enter om door te gaan en werk bij met `apt-get update`. Het commando `apt-cache policy apache2` laat zien dat er nu een recentere versie is. Je kunt nu Apache installeren of, als het al eerder is geïnstalleerd, updaten met `apt-get install apache2`. Met `apachectl -v` zie je welke versie is geïnstalleerd. Momenteel is 2.4.25 beschikbaar. Werk met `apt-get upgrade` eventuele aanvullende pakketten bij.

VIRTUAL HOST

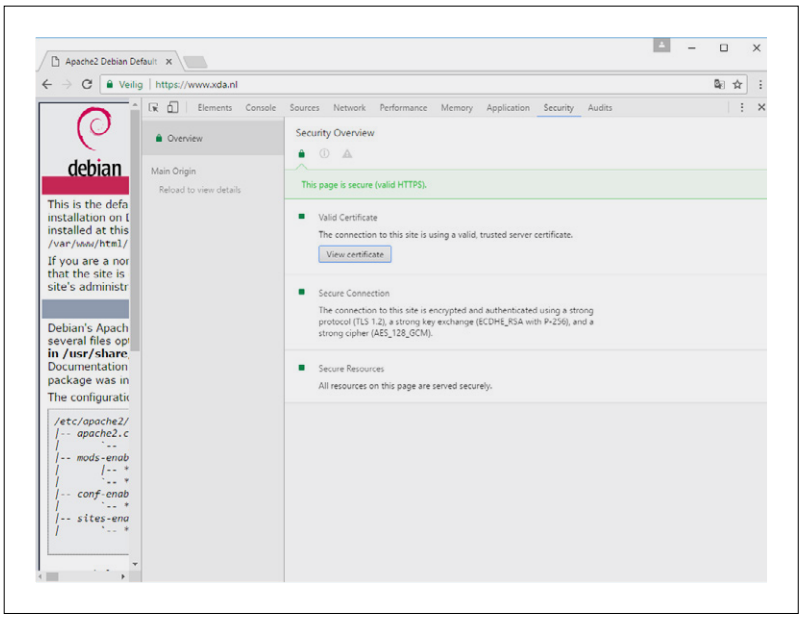
Apache maakt na de installatie een standaard virtual host aan in de map /var/www/html. Die pagina zie je als je het ip-adres van de server opent in een browser. De configuratie van deze virtual host vind je in **000-default.conf** in de map /etc/apache2/sites-available. Daar vind je ook **default-ssl.conf** voor de ssl-configuratie. Die twee bestanden kun je eventueel als basis voor andere virtual hosts gebruiken, als je meer dan één website wilt activeren. Je zou voor een https-website zelfs andere content kunnen laten zien dan voor de http-versie door via DocumentRoot naar een andere map te verwijzen. Belangrijk voor die virtual hosts, maar ook voor de aanvraag van certificaten is dat je de domeinnaam in dat configuratiebestand zet als **ServerName** samen met eventuele aliassen. In deze workshop gebruiken we **domein.nl** als basisdomein en **www.domein.nl** als alias. Geef dit in **000-default.conf** aan met **ServerName domein.nl** met in de regel daaronder **ServerAlias www.domein.nl** of eventueel **ServerAlias *.domein.nl** om meteen alle subdomeinen af te vangen.

MODULE ACTIVEREN

Om de http2-module met Apache te gebruiken, moet je deze eerst activeren met de opdracht `a2enmod http2`. Activeer ook ssl met `a2enmod ssl` en acti-



» De standaard-pagina van Apache.



Het certificaat van Let's Encrypt is in een handomdraai geïnstalleerd.

veer de virtual host met `a2ensite default-ssl`. Hiermee wordt een symlink aangemaakt in de map /etc/apache2/sites-enabled naar het bestand **/etc/apache2/sites-available/default-ssl.conf** zodat deze in de actieve Apache-configuratie wordt opgenomen. Herstart daarna Apache met `systemctl restart apache2` om de nieuwe configuratie actief te maken. Test de ssl-verbinding door **https://[ipadres]** in de browser te openen. Je krijgt een waarschuwing omdat het self-signed certificaat niet wordt vertrouwd. Zoals je in **default-ssl.conf** kunt zien, gaat het om het certificaat **/etc/ssl/certs/ssl-cert-snakeoil.pem**. Wil je meer inzicht, dan ga je in Chrome met Ctrl+Shift+I naar Hulpprogramma's voor ontwikkelaars. Het tabblad **Security** geeft details over het bewuste certificaat. Om van de waarschuwing af te komen gaan we het certificaat vervangen door een certificaat van Let's Encrypt.

CERTBOT INSTALLEREN

Voor het ophalen van het certificaat heb je een acme-client op je server nodig. Wij kiezen Certbot. De ontwikkelaars houden voor Ubuntu een speciale repository bij met de laatste versie, die nog actief wordt ontwikkeld. Met `add-apt-repository ppa:certbot/certbot` voeg je deze repository toe. Druk op Enter om door te gaan en werk bij met `apt-get update`. Installeer vervolgens Certbot met `apt-get install python-certbot-apache`. De tool kan voor Apache automatisch een certificaat aanvragen én installeren.

CERTIFICAAT AANVRAGEN

We willen dat het certificaat zowel voor het basisdomein **domein.nl** als voor **www.domein.nl** geldig is en moeten ze daarom beide opgeven. Dat kan met `certbot --apache -d domein.nl -d www.domein.nl`. Begin altijd met het basisdomein, in dit geval domein.nl, gevolgd door alle subdomeinen, met een limiet van 100. Een wildcard zoals je bij sommige duurdere certificaten ziet is helaas niet mogelijk bij Let's Encrypt, je zult dus alle namen op moeten geven. Certbot vraagt vervolgens om je e-mailadres en privacy-voorkeuren. Als laatste kies je of je toegang via zowel http als https wilt toestaan, of dat http-verzoeken moeten worden omgeleid naar https. Wij kiezen de eerste optie. In de volgende workshop laten we zien hoe je die tweede optie handmatig activeert.

DIRECT ACTIEF

Het nieuwe certificaat is direct actief. De bestanden die bij het certificaat horen vind je in de map /etc/letsencrypt/live/domein.nl. Je kunt het proberen door je website via https te bezoeken. De configuratie voor de https-website is opgenomen in **/etc/apache2/sites-available/000-default-le-ssl.conf**. Je ziet dat Let's Encrypt de algemene ssl-configuratie in **/etc/letsencrypt/options-ssl-apache.conf** zet. Die wordt met alle virtual hosts gedeeld, wat wel zo handig is als je enkele aanpassingen wilt maken.

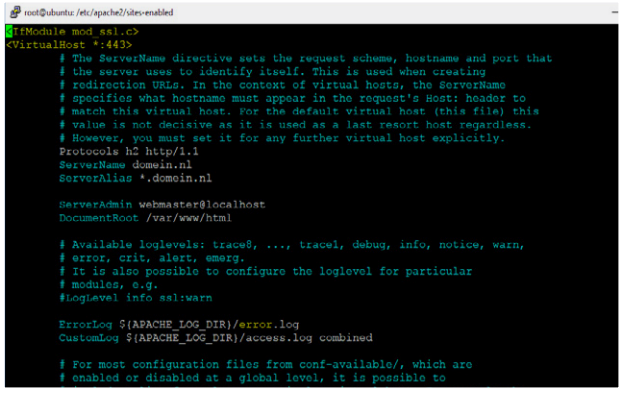
HTTP/2 AANZETTEN

We moeten, in het bestand **000-default-le-ssl.conf**, alleen nog expliciet http/2 aanzetten voor deze website. Daarvoor voeg je de regel **Protocols h2 http/1.1** toe. Het bestand ziet er dan als volgt uit (herstart hierna Apache):

```
<VirtualHost *:443>
  Protocols h2 http/1.1
  ServerName domein.nl
  ServerAlias *.domein.nl
  ...
</VirtualHost>
```

Je kunt http/2 overigens ook met een globale instelling direct voor iedere site actief maken, maar omdat het nog vrij nieuw is, is het verstandig dit per virtual host te doen. Via bijvoorbeeld **https://tools.keycdn.com/http2-test** kun je controleren of http/2 werkt, of je kunt de headers onderzoeken met de hulpprogramma's van Chrome en Firefox. Hoewel je http/2 ook voor de gewone http-versie kunt aanzetten heeft dat weinig zin, omdat browsers het protocol alleen voor https-verkeer ondersteunen. Een mooi extraatje van http/2 is de push-mogelijkheid waarmee je bestanden, zoals stylesheets of fonts, kunt voorladen.

De ssl-configuratie voor de website als virtual host.



Testen, optimaliseren en automatiseren

GERTJAN GROEN

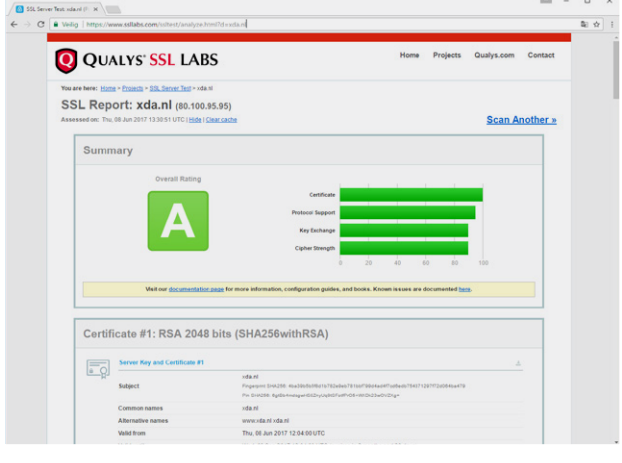
OPTIMALISEER JE OMGEVING

HET CERTIFICAAT WERKT, MAAR HOE NU VERDER? IN DEZE WORKSHOP GAAN WE HET CERTIFICAAT TESTEN, VERBETEREN WE DE SERVEROMGEVING EN LATEN WE ZIEN HOE JE HET CERTIFICAAT AUTOMATISCH KUNT VERNIEUWEN.

Een soms wat uitdagende, maar wel belangrijke stap is dat voor de https-versie alle bronnen veilig worden geladen, zoals alle afbeeldingen, stylesheets en javascript. Doe je dit niet, dan toont de browser een waarschuwing dat de pagina veilige en onveilige inhoud probeert te combineren. Voor je website kun je het gemakkelijk oplossen door een absoluut pad zoals http://www.domein.nl/css/style.css te veranderen van http naar https of door //domein.nl/css/style.css te gebruiken Bij laatstgenoemde optie zal de browser automatisch het juiste protocol (http of https) selecteren. Een relatief pad zoals css/style.css mag overigens ook altijd. Je bent er nu nog niet: ook externe elementen zoals reclame of koppelingen naar sociale netwerken moeten via https worden opgevraagd. Gelukkig zijn vrijwel alle grote advertentienetwerken en de belangrijkste adverteerders al overgeschakeld naar https.

CERTIFICAAT TESTEN

Je kunt het certificaat testen via bijvoorbeeld Qualys SSL Labs (**www.ssllabs.com/ssltest**). Het wordt hier uitvoerig getest en configuratiefouten komen direct naar voren. Er komt direct een rating uit. De laagste score die je kunt krijgen is F en de hoogste is A+. Een hogere rating gaat soms samen met beperktere compatibiliteit, maar bij de standaardinstellingen van Let's Encrypt blijkt de compatibiliteit dik in orde terwijl er toch een A-rating uit komt. Verderop lees je hoe je hier een A+-rating van kunt maken. Ook Observatory van Mozilla (**https://observatory.mozilla.org**) is een handige tool, die een ander scoremodel hanteert en ook weer andere tips geeft. Verder zijn er talloze blogs op internet die je helpen bij het verbeteren van je rating in bijvoorbeeld Apache of Nginx. Dat is overigens een voortdurend proces. De rating kan zomaar veranderen als nieuwe kwetsbaarheden worden ontdekt in software, protocollen of algoritmen. Het is dus belangrijk de configuratie af en toe te controleren en bij te werken net als de software, met name OpenSSL.



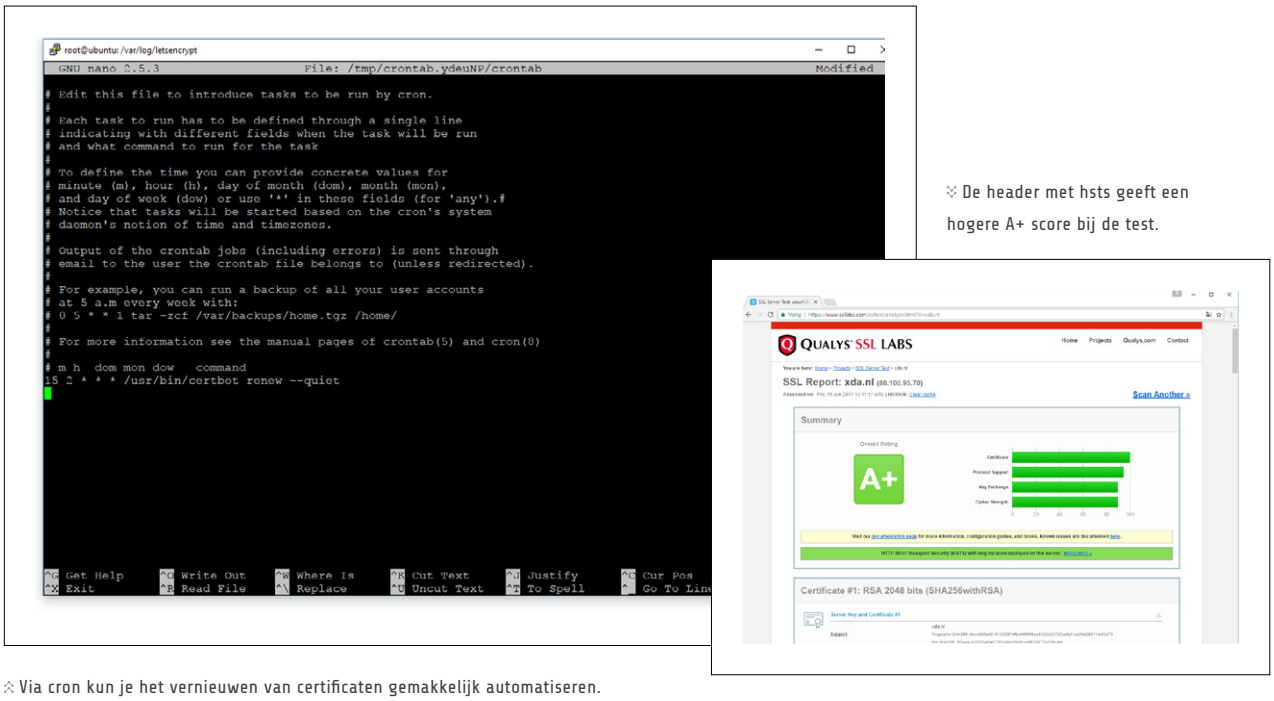
Via Qualys SSL Labs kun je een ssl-certificaat testen.

CERTIFICATEN VERNIEUWEN

Bovengenoemde testen laten ook zien hoe lang het certificaat geldig is, in dit geval drie maanden. Door die relatief korte geldigheid word je aangemoedigd om certificaten automatisch te vernieuwen en dat is precies wat we gaan doen. Dat gaat overigens eenvoudig via de opdracht `certbot renew`. Hiermee worden alle certificaten die op het systeem zijn geïnstalleerd gecontroleerd en indien nodig vernieuwd. Je kunt dit proces testen via `certbot renew --dry-run`. Met de hulp van cron kun je dit gemakkelijk automatiseren zodat je er geen omkijken meer naar hebt.

CRON-JOB TOEVOEGEN

Met cron kun je periodiek bepaalde taken op de server uitvoeren. We gaan hiermee dagelijks de certificaten te vernieuwen. Dat lijkt vaak, maar het is wat Let's Encrypt zelf aanbeveelt. Er kan immers een probleem zijn bij de servers van Let's Encrypt of op je eigen servers waardoor het vernieuwen een enkele keer niet lukt. Je hoeft niet bang te zijn dat je eerder tegen de (overigens zeer ruime) limieten aanloopt: alleen certificaten die binnen dertig dagen verlopen worden vernieuwd. Om de taak toe te voegen, bewerk



Via cron kun je het vernieuwen van certificaten gemakkelijk automatiseren.

De header met hsts geeft een hogere A+ score bij de test.

je een bestand genaamd crontab met `crontab -e`. Kies je favoriete teksteditor, bijvoorbeeld Nano. Voeg de volgende regel toe aan het bestand:

```
15 2 * * * /usr/bin/certbot renew --quiet
```

De taak wordt dagelijks om 2:15 uitgevoerd. Kies hier liefst zelf een (willekeurige) tijd, ergens 's nachts. De optie **--quiet** zorgt dat geen gebruikersinput wordt gevraagd. Controleer wel af en toe **/var/log/letsencrypt/letsencrypt.log** voor foutmeldingen.

BROWSER DOORVERWIJZEN

Wil je volop gebruik gaan maken van https, dan is het verstandig om al het http-verkeer door te wijzen naar de https-versie van je website. Dat doe je bij voorkeur met een aanpassing in de virtual host-configuratie, maar kan ook met de hulp van de rewrite-module. Voor de eerste optie open je de virtual host-configuratie voor de 'http-website', in ons voorbeeld is dat **/etc/apache2/sites-available/000-default.conf**. Hieraan voeg je een **Redirect** toe zoals in het voorbeeld hieronder. Als alles naar wens werkt kun je hier ook `Redirect permanent` van maken om aan te geven dat de wijziging permanent is.

```
<VirtualHost *:80>
  ServerName domein.nl
  ServerAlias *.domein.nl
  Redirect / https://www.domein.nl
</VirtualHost>
```

VERWIJZING IN .HTACCESS

Als tweede optie kun je enkele regels in een bestand met de naam .htaccess zetten in de root van de website (in dit geval /var/www/html). Hiervoor moet ook de rewrite-module actief zijn via `a2enmod rewrite`. Ook dien je in de Apache-configuratie (**/etc/apache2/apache2.conf**), onder **<Directory /var/www/>**, de optie **AllowOverride None** veranderen naar **AllowOverride All** zodat het .htaccess-bestand wordt gehonoreerd. Herstart Apache met `systemctl restart apache2`. Zet ten slotte onderstaande regels in **/var/www/html/.htaccess**. De **RewriteCond** bepaalt dat de regel daaronder (de daadwerkelijke rewrite) alleen plaatsvindt als een niet-https-bestand wordt opgevraagd.

```
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R,L]
```

AANGEPASTE HEADER MET HSTS

Om herhaalde omleidingen in de browser te voorkomen en daarmee de toegang iets te versnellen, is het handig om via een speciale header aan te geven dat de browser alle pagina's een bepaalde tijd direct via https moet laden. Hiervoor kun je HTTP Strict Transport Security (hsts) gebruiken. Je hebt bij Apache de module-headers nodig die je via `a2enmod headers` activeert. Herstart met het commando `systemctl restart apache2`. Zet onderstaande regel in het .htaccess-bestand om de harde omleiding gedurende een jaar (31536000 seconden) actief te maken:

```
Header always set Strict-Transport-Security "max-age=31536000" env=HTTPS
```

Omdat ook de beveiliging hiermee wordt aangescherpt, beloont Qualys SSL Labs je met een hogere A+-score. Zorg wel dat de omleiding minimaal een half jaar geldt (15552000 seconden). Maak de aanpassing overigens pas als je https-website volledig werkt (en blijft werken) en alle onderdelen netjes via https worden geladen.

HET ACTIVEREN VAN HTTP STRICT TRANSPORT SECURITY GEEFT JE EEN **NOG BETERE A+RATING BIJ QUALYS SSL LABS**