Nationaal Cyber Security Centrum
Ministerie van Veiligheid en Justitie

ICT-beveiligings-richtlijnen voor Transport Layer Security (TLS)

» *ICT-beveiligings-richtlijnen voor Transport Layer Security (TLS)*

Nationaal Cyber Security Centrum
Turfmarkt 147 | 2511 DP Den Haag
Postbus 117 | 2501 CC Den Haag

T 070-751 55 55
F 070-322 25 37

E info@ncsc.nl
I www.ncsc.nl

November 2014

Nationaal Cyber Security Centrum
Het Nationaal Cyber Security Centrum (NCSC) draagt via samenwerking tussen bedrijfsleven, overheid en wetenschap bij aan het vergroten van de weerbaarheid van de Nederlandse samenleving in het digitale domein.

Het NCSC ondersteunt de Rijksoverheid en organisaties met een vitale functie in de samenleving met het geven van expertise en advies, response op dreigingen en het versterken van de crisisbeheersing. Daarnaast voorziet het in informatie en advies voor burger, overheid en bedrijfsleven ten behoeve van bewustwording en preventie. Het NCSC is daarmee het centrale meld- en informatiepunt voor ICT-dreigingen en -veiligheidsincidenten.

Deze ICT-Beveiligingsrichtlijnen voor Transport Layer Security (TLS) zijn in 2014 gepubliceerd door het NCSC. Bij het opstellen van deze richtlijnen heeft het NCSC samengewerkt met:
- Nationaal Bureau voor Verbindingsbeveiliging, onderdeel van de AIVD.

De volgende partijen hebben waardevolle bijdragen geleverd:
- Atos
- Belastingdienst
- Bureau Forum Standaardisatie
- Capgemini
- Centric
- Dienst Publiek en Communicatie
- KPN
- Logius
- OpenFortress B.V.
- Rabobank
- Stichting NLnet

INHOUDSOPGAVE

Over de richtlijnen 4
Doel 4
Gebruik bij inkoop 4
Niveaus 4
Hoofdboodschap 5
Leeswijzer 5
Verwijzingen 5

Hoofdstuk 1 » Wat is Transport Layer Security? 6
Werking van TLS 7
Programmeerbibliotheken 7
Het belang van random numbers 8

Hoofdstuk 2 » Gebruiksadvies 10
Scenario 1: Controle over client en server 11
Scenario 2: Alleen controle over server 12
Aandachtspunten 12
Afwijken van het gebruiksadvies 12

Hoofdstuk 3 » Opties en instellingen 14
Versies 15
Cipher suites 15
RSA-sleutels 17
DH(E)-parameters en DSS-sleutels 17
ECDH(E)-parameters, ECDSA-sleutels en elliptische krommen 17
Overige opties 17

Hoofdstuk 4 » Richtlijnen 20
Versies 21
Cipher suites 21
Certificaat 21
Diffie-Hellman-parameters 21
Elliptische krommen 22
Overige opties 22

Appendix » Verdere overwegingen 24
Forward secrecy 25
Certificaatbeheer 25
Random number generators 25
Waar eindigt de TLS-verbinding? 26
Certificate pinning en DANE 26

Appendix » Namen van cipher suites in OpenSSL en GnuTLS 28

Appendix » Wijzigen van deze richtlijnen 30
Acute wijzigingen 31
Nieuwe versies 31

Appendix » Verklarende woordenlijst 32

Referenties en colofon 36

INLEIDING

Over de richtlijnen

Over de richtlijnen

Deze richtlijnen zijn bedoeld als advies bij het inkopen, opstellen en beoordelen van configuraties voor het Transport Layer Security-protocol (TLS). TLS is het meest gebruikte protocol voor het beveiligen van verbindingen op internet.

Doel

Deze richtlijnen bevatten geen stap-voor-stap-instructies voor het configureren van TLS[1]. Ze zijn wel technisch van aard. Dit document helpt een organisatie te kiezen tussen alle mogelijke instellingen van TLS, om zo te komen tot een veilige configuratie. Deze configuratie wordt vervolgens uitgevoerd door een beheerder of leverancier.

Gebruik bij inkoop

Organisaties die ICT-systemen inkopen, kunnen verwijzen naar dit document bij het stellen van eisen aan te leveren producten. Een leverancier wordt op die manier gevraagd om een veilige TLS-configuratie te leveren door te voldoen aan de richtlijnen uit dit document.

Niveaus

De beslissing over de juiste TLS-configuratie maakt elke organisatie uiteindelijk zelf. Het opstellen van een veilige configuratie is een lastig karwei. U dient voor elke optie te kiezen uit verschillende instellingen, soms talloos veel. Bij deze keuze houdt u natuurlijk veiligheid in het achterhoofd, maar ook compatibiliteit met de software van klanten of eindgebruikers speelt een rol. Deze richtlijnen vormen daarbij een gids.

Om de keuze voor een configuratie te vergemakkelijken, verdelen we de instellingen voor TLS-opties in deze richtlijnen in drie veiligheidsniveaus:

- Een instelling die **Onvoldoende** is, dient niet gekozen te worden. TLS-configuraties die deze instelling bevatten, zijn niet veilig.
- Is een instelling **Voldoende**, dan betekent dat 'voldoet nu nog'. Het is nu nog mogelijk om deze instelling in een veilige TLS-configuratie te gebruiken. Veel **Voldoende** instellingen zijn nodig voor compatibiliteit met oudere clientsystemen.
- De veiligste instellingen heten **Goed**. Heeft u de vrijheid om te kiezen welke instellingen u wilt ondersteunen, gebruik dan waar mogelijk alleen **Goede** instellingen.

De woorden 'onvoldoende', 'voldoende' en 'goed' hebben ook een betekenis in gewoon taalgebruik. Om het onderscheid duidelijk te maken, worden deze woorden in de richtlijnen in een **ander lettertype** weergegeven.

Van tijd tot tijd worden er aanvalstechnieken voor TLS gevonden. Deze aanvalstechnieken betreffen meestal **Voldoende** instellingen. Een instelling die door een ontdekte aanvalstechniek onveilig wordt, zal zijn **Voldoende**- of **Goed**-status verliezen en **Onvoldoende** worden. Als dit gebeurt, zal een addendum op deze richtlijnen worden uitgebracht. Zie voor een toelichting de appendix Wijzigen van deze richtlijnen.

1. Het boek 'Bulletproof SSL and TLS' van Ivan Ristic (ISBN 978-1907117046) biedt, naast veel achtergrondinformatie over TLS, ook stap-voor-stap-instructies voor het configureren van allerlei software voor veilig gebruik van TLS. De website https://bettercrypto.org/ geeft ook stap-voor-stap-instructies. Merk op dat de adviezen op die website in nuances afwijken van de adviezen in dit document.

Goede instellingen zullen waarschijnlijk toekomstbestendiger zijn dan **Voldoende** instellingen. Dit is echter niet te garanderen. Verder is geen enkele TLS-configuratie eeuwig houdbaar. Ook een TLS-configuratie die alleen uit **Goede** instellingen bestaat, zal uiteindelijk bijgewerkt moeten worden. Dit geldt als **Goede** instellingen door nieuwe aanvalstechnieken **Onvoldoende** worden.

BEAST-aanvalstechniek

In 2011 hebben onderzoekers de Browser Exploit Against SSL/TLS (BEAST)-aanvalstechniek ontdekt. Deze aanvalstechniek was mogelijk door de manier waarop initialisatievectoren (IV's) gebruikt werden in bulkversleuteling met cipher block chaining (CBC)-modus.

BEAST is een browserkwetsbaarheid. Moderne browsers bevatten inmiddels maatregelen waardoor ze niet meer kwetsbaar zijn voor de BEAST-aanvalstechniek. In veel gevallen is de aanvalstechniek dus niet meer praktisch toe te passen.

Omdat BEAST een browserkwetsbaarheid is, is het niet aan de serverbeheerder om BEAST te voorkomen. De serverbeheerder heeft ook geen goede middelen om BEAST tegen te gaan. De enige 'remedie' is om RC4 te gebruiken voor bulkversleuteling in plaats van een algoritme in CBC-modus. RC4 kent echter zijn eigen kwetsbaarheden en dient niet gekozen te worden. GCM is een veilig alternatief, maar dit wordt slechts zeer beperkt ondersteund.

Hoofdboodschap

TLS veilig configureren is belangrijk voor het goed beveiligen van netwerkverbindingen. TLS kent veilige en minder veilige instellingen. Oudere software ondersteunt niet altijd de veiligste instellingen. Gebruik waar mogelijk **Goede** instellingen, en vul deze aan met **Voldoende** instellingen om oudere software te ondersteunen. Moet u veel verschillende oudere software ondersteunen? Gebruik dan een breed palet aan **Voldoende** instellingen, en vul deze waar mogelijk aan met **Goede** instellingen. Gebruik geen **Onvoldoende** instellingen.

2. Op deze manier.

Leeswijzer

De kern van deze richtlijnen wordt gevormd door de hoofdstukken *Gebruiksadvies*, *Richtlijnen* en *Opties en instellingen*. Het hoofdstuk Gebruiksadvies is gericht op personen die zelf een veilige TLS-configuratie moeten opstellen. Het biedt een stapsgewijs handelingsperspectief om tot een veilige configuratie te komen. Het hoofdstuk Opties en instellingen somt relevante TLS-opties op. Het geeft aan wat de veilige instellingen voor elke optie zijn. Andere hoofdstukken verwijzen regelmatig naar het hoofdstuk Opties en instellingen voor details. Het hoofdstuk Richtlijnen is bedoeld voor personen die een bepaalde TLS-configuratie moeten beoordelen, zoals auditors. Het kan hierbij zowel gaan om een configuratie op papier als een configuratie in de praktijk.

Deze richtlijnen zijn op drie manieren te lezen:

- Ontwerpt u zelf een TLS-configuratie, lees dan het hoofdstuk *Wat is Transport Layer Security?*, gevolgd door het hoofdstuk *Gebruiksadvies*. Het hoofdstuk *Gebruiksadvies* zal u verwijzen naar relevante delen van het hoofdstuk *Opties en instellingen*.
- Wilt u weten hoe bepaalde instellingen voor TLS-opties de veiligheid van TLS beïnvloeden? Raadpleeg dan het hoofdstuk *Opties en instellingen*.
- Beoordeelt u een TLS-configuratie, lees dan het hoofdstuk *Wat is Transport Layer Security?*, gevolgd door het hoofdstuk *Richtlijnen*. Het hoofdstuk *Richtlijnen* zal u verwijzen naar relevante delen van het hoofdstuk *Opties en instellingen*.

Verwijzingen

In de richtlijnen worden meerdere soorten verwijzingen gebruikt:

- Technische termen worden niet altijd uitgelegd bij het eerste gebruik. Is een term op deze manier gemarkeerd, dan is deze te vinden in de verklarende woordenlijst achterin.
- Voor het opnemen van achtergrondinformatie worden voetnoten[2] gebruikt.
- De onderbouwing van de gegeven adviezen wordt gevormd door de referenties achterin. Waar een bepaalde referentie de onderbouwing vormt voor een advies, is deze referentie op de volgende manier aangeduid: (**I**).

HOOFDSTUK 1

Wat is Transport Layer Security?

Wat is Transport Layer Security?

Transport Layer Security (TLS) is een protocol voor het opzetten en gebruiken van een cryptografisch beveiligde verbinding tussen twee computersystemen, een client en een server. Nadat met het TLS-protocol een beveiligde verbinding is opgezet, kunnen applicaties de verbinding gebruiken om data uit te wisselen tussen client en server. TLS wordt gebruikt in een breed scala van toepassingen. Bekende voorbeelden zijn webverkeer (HTTPS), e-mailverkeer (IMAP en SMTP over STARTTLS) en bepaalde typen virtual private network (VPN).

Waarom TLS?

TLS beschermt communicatie tussen een client en een server. Communicatie beschermen is vooral belangrijk als er gevoelige informatie over een verbinding wordt verstuurd. Informatie kan gevoelig zijn vanwege de vertrouwelijkheid (bijvoorbeeld inloggegevens) en vanwege de integriteit (bijvoorbeeld een bankoverschrijving).

Persoonsgegevens worden vaak als gevoelig gezien. Bevat uw website een contactformulier of profielpagina's? TLS zorgt ervoor dat de verzonden gegevens onderweg niet ingezien kunnen worden door anderen.

In sommige gevallen is het gebruik van versleutelde verbindingen verplicht. Deze verplichting kan gesteld zijn in beleid van uw organisatie of uw sector, maar ook in wet- of regelgeving. De Norm ICT-beveiligingsassessments DigiD van Logius is een voorbeeld. Organisaties die diensten op basis van DigiD aanbieden, zijn verplicht deze diensten via HTTPS beschikbaar te maken.

TLS beveiligt alleen de inhoud van de communicatie. Informatie over het datatransport wordt niet beschermd. In dit opzicht verschilt TLS van IPsec. TLS werkt op de transport layer, IPsec werkt op de internet layer[3].
Er zijn momenteel zes verschillende versies van TLS. Drie daarvan dragen nog de oude naam: Secure Sockets Layer (SSL) 1.0, 2.0 en 3.0. Deze zijn ontwikkeld door Netscape. De drie nieuwere versies zijn TLS 1.0, 1.1 en 1.2. Deze zijn ontwikkeld door de Internet Engineering Task Force (IETF). De IETF onderhoudt TLS als open standaard. De recentste versie is TLS 1.2[4].

Een client of server kan meerdere versies van TLS ondersteunen. De versies zijn onderling niet compatibel. Elke versie kent zijn eigen opties, bijvoorbeeld op het gebied van versleuteling, authenticatie en sleuteluitwisseling.

Werking van TLS

Een verbinding tussen een client en server die met TLS beveiligd is, heet een TLS-sessie. Een TLS-sessie bestaat uit twee fasen: de handshake en de applicatiefase. Tijdens de handshake spreken client en server af op welke manier de TLS-sessie wordt opgezet. De uitgewisselde informatie is tijdens de handshake nog niet versleuteld. De volgende zaken worden onder andere tijdens de handshake afgesproken:

- Welke *versie* van TLS gaat gebruikt worden?
- Welke *cipher suite* gaat gebruikt worden?
- Welk *certificaat* biedt de server aan om zijn identiteit te bewijzen?
- Biedt de client ook een certificaat aan? Zo ja, welk?
- Welke *sleutel* gaat gebruikt worden om verdere data uit te wisselen?

De handshake wordt gestart door de client. Tijdens de handshake controleert de client de authenticiteit van het certificaat dat de server aanbiedt. Biedt de client ook een certificaat aan de server, dan controleert de server ook de authenticiteit van het certificaat dat de client aanbiedt.

Nadat de handshake is afgerond, begint de applicatiefase. Tijdens de applicatiefase is de TLS-sessie beschikbaar als een beveiligde tunnel voor dataverkeer. Applicaties kunnen van deze tunnel gebruikmaken om hun eigen verkeer te verzenden tussen de client en de server. Applicaties hoeven zich niet te bekommeren om de werking van de tunnel: zij kunnen deze gebruiken als abstract communicatiekanaal dat vertrouwelijkheid en integriteit van gegevens garandeert.

Programmeerbibliotheken

TLS wordt door veel verschillende softwareapplicaties gebruikt. Het programmeren van alle functionaliteit van TLS is veel werk en vergt specialistische kennis. Daarom bevat de meeste software geen eigen code voor TLS, maar maakt deze gebruik van een TLS-programmeerbibliotheek (TLS library).

Er zijn verschillende TLS-programmeerbibliotheken. Sommige zijn vrije software, andere zijn als gesloten

3. De transport layer en de internet layer zijn onderdeel van de Internet protocol suite, een model om netwerkverkeer mee te beschrijven. Dit ontwerp wordt beschreven in RFC 1122, te raadplegen op https://datatracker.ietf.org/doc/rfc1122/.
4. De specificatie van TLS 1.2 is vastgelegd in RFC 5246, te raadplegen op https://datatracker.ietf.org/doc/rfc5246/.

product beschikbaar. Ze kunnen apart worden geleverd, of worden meegeleverd met besturingssystemen of internetbrowsers. Bekende TLS-programmeerbibliotheken zijn bijvoorbeeld OpenSSL[5], GnuTLS[6], SChannel[7], NSS[8] en PolarSSL[9].

Deze richtlijnen vellen geen oordeel over de veiligheid van specifieke programmeerbibliotheken. Elke programmeerbibliotheek kent voordelen en nadelen. Niet elke instelling voor TLS is in elke programmeerbibliotheek beschikbaar. Over het algemeen valt wel te zeggen dat alle software programmeerfouten bevat, dus ook deze programmeerbibliotheken. Programmeerfouten kunnen leiden tot kwetsbaarheden. Het NCSC adviseert om bewust gebruik te maken van TLS-programmeerbibliotheken:

- Gebruik altijd de recentste versie van de gekozen bibliotheek. De maker heeft bij deze versie het langst de tijd gehad om kwetsbaarheden te verhelpen.
- Kies alleen instellingen waarvoor een business case bestaat. Dit voorkomt dat een programmeerfout in niet-noodzakelijke functionaliteit tot een kwetsbaarheid van het systeem leidt. Het hoofdstuk *Gebruiksadvies* helpt daarbij.

Het belang van random numbers

Random numbers (toevalsgetallen) spelen een belangrijke rol in veel toepassingen van cryptografie, ook in TLS. TLS gebruikt random numbers op meerdere plaatsen in het protocol.

De kwaliteit van de gebruikte random numbers vormt een pijler van de betrouwbaarheid van TLS. Het kiezen van de juiste instellingen voor TLS is belangrijk, maar geen enkele instelling kan het risico wegnemen van het gebruik van random numbers van lage kwaliteit.

Elk besturingssysteem en elke TLS-programmeerbibliotheek bevat methodes om random numbers te genereren. Daarnaast zijn er hardwaremodules verkrijgbaar om random numbers te genereren. Zulke hardwaremodules produceren sneller random numbers van hogere kwaliteit dan strikt softwarematige methodes.

U vindt meer achtergronden en adviezen over methodes voor het genereren van random numbers in de sectie Random number generators van de appendix Verdere overwegingen.

5. https://www.openssl.org/
6. http://www.gnutls.org/
7. http://msdn.microsoft.com/en-us/library/windows/desktop/aa380123%28v=vs.85%29.aspx
8. https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS
9. https://polarssl.org/

HOOFDSTUK 1 » WAT IS TRANSPORT LAYER SECURITY?

HOOFDSTUK 2

Gebruiksadvies

Gebruiksadvies

De richtlijnen uit dit document richten zich op de veiligheid van een TLS-configuratie. In de praktijk speelt niet alleen de veiligheid een rol bij de keuze van een configuratie. De TLS-configuratie van een server moet ook compatibel zijn met de TLS-configuratie van alle clients die met de server verbinden. De configuraties van client en server dienen op bepaalde aspecten overeen te komen.

Bij sommige opties sluiten meerdere keuzes elkaar niet uit. Dat geldt voor ondersteunde versies, cipher suites en elliptische krommen. Om een client en server te laten communiceren, dient er bij elk van deze opties minstens één keuze te zijn die client en server allebei ondersteunen. Bijvoorbeeld: als de client versies SSL 3.0, TLS 1.0 en TLS 1.1 ondersteunt, kan deze wel communiceren met een server die versies TLS 1.0 en TLS 1.1 ondersteunt, maar niet met een server die alleen versie TLS 1.2 ondersteunt. Hetzelfde principe geldt voor cipher suites en elliptische krommen.

Andere opties bepalen hoe lang een bepaalde cryptografische sleutel of andere waarde is. Dat geldt voor RSA-sleutels, DH-parameters, DSS-sleutels, ECDH-parameters en ECDSA-sleutels. Om een client en server te laten communiceren, dienen zowel client als server de gekozen lengte van de sleutel of parameter te ondersteunen. Oudere software ondersteunt niet altijd sleutels en parameters van voldoende lengte, en nieuwere software sluit soms juist sleutels en parameters van onvoldoende lengte uit.

Tot slot zijn er opties die slechts ‘aan’ of ‘uit’ kunnen staan. In dit document worden deze de ‘Overige opties’ genoemd. Er zijn geen gevallen bekend waarin het in- of uitschakelen van deze opties op de server kan leiden tot compatibiliteitsproblemen met clients.

Scenario 1: Controle over client en server

In sommige situaties heeft de partij die zeggenschap heeft over de configuratie van een server ook zeggenschap over de configuratie van alle clients die met de server verbinden. Een voorbeeld is een webserver waarop een interne webapplicatie wordt aangeboden. Deze is immers alleen bereikbaar voor clients van de organisatie zelf. Het NCSC adviseert om in zulke gevallen zo veel mogelijk voor **Goede** instellingen te kiezen: dit is de veiligste en meest toekomstvaste optie. In het hoofdstuk Opties en instellingen vindt u welke instellingen dat zijn.

Stappenplan

1. Inventariseer welke instellingen voor TLS-opties er beschikbaar zijn op de server.
2. Inventariseer welke clients er verbinding zullen maken met de server.
3. Inventariseer bij elk type client welke instellingen voor TLS-opties deze ondersteunt[10].
4. Kies **Goede** instellingen voor de volgende opties:
   a. TLS-versies voor de server. Zorg ervoor dat elke client een versie ondersteunt die de server ook ondersteunt.
   b. Cipher suites voor de server. Zorg ervoor dat elke client een cipher suite ondersteunt die de server ook ondersteunt.
   c. Lengtes van parameters en sleutels voor de server. Zorg ervoor dat de gekozen lengte door elke client wordt ondersteund.
   d. Elliptische krommen voor de server, indien van toepassing. Zorg ervoor dat elke client een elliptische kromme ondersteunt die de server ook ondersteunt.
   e. Overige opties voor de server, tenzij er sterke overwegingen zijn om een **Voldoende** instelling te kiezen. Deze overwegingen volgen uit de client-inventarisatie van stap 2.
5. Ondersteunt de server voor een bepaalde optie geen **Goede** instelling die door de clients wordt ondersteund? U heeft drie mogelijkheden:
   a. Vervang de clients door een type dat wel een **Goede** instelling voor deze optie ondersteunt.
   b. Vervang de server door een type dat wel een **Goede** instelling voor deze optie ondersteunt.
   c. Kies voor de desbetreffende optie een **Voldoende** instelling die zowel door client als server wordt ondersteund.
6. Configureer de server met de gekozen instellingen. De TLS-configuratie is onderdeel van de configuratie van software die de TLS-verbinding gebruikt. Wilt u bijvoorbeeld HTTPS aanbieden, dan configureert u TLS in de configuratie van de webserversoftware[11].
7. Test of deze configuratie inderdaad werkt met alle typen clients. Ondervindt u compatibiliteitsproblemen? Ga dan terug naar stap 5.
8. Documenteer de gekozen instellingen. Besteed daarbij in elk geval aandacht aan de redenen om te kiezen voor **Voldoende** in plaats van **Goede** instellingen.

10. Een overzicht van de TLS-configuraties van verschillende typen clients is te vinden op https://www.ssllabs.com/ssltest/clients.html.
11. In sommige gevallen vormt een ander netwerkapparaat het eindpunt van de TLS-sessie. Meer informatie hierover vindt u in de sectie Waar eindigt de TLS-verbinding? van de appendix Verdere overwegingen.

Scenario 2: Alleen controle over server

In andere gevallen heeft de partij die zeggenschap heeft over de configuratie van een server geen zeggenschap over de configuratie van (alle) clients die met de server verbinden. Een voorbeeld is een webserver waarop een publieke website wordt aangeboden. Het NCSC adviseert om in deze gevallen voor zo veel mogelijk opties **Goede** instellingen te kiezen, en deze aan te vullen met **Voldoende** instellingen. In het hoofdstuk *Opties en instellingen* vindt u welke instellingen dat zijn.

Stappenplan

1. Inventariseer welke typen clients moeten kunnen verbinden met de server. Dit is een eis die meestal in overleg met de zakelijk eigenaar wordt vastgesteld.
2. Inventariseer welke instellingen voor TLS-opties er beschikbaar zijn op de server.
3. Kies **Goede** en **Voldoende** TLS-versies voor de server.
4. Kies **Goede** en **Voldoende** cipher suites voor de server. Ondersteun een breed palet aan **Voldoende** cipher suites: deze zijn vaak nodig voor compatibiliteit met oudere clients[12]. Ondersteun echter niet meer cipher suites dan nodig voor compatibiliteit.
5. Kies **Voldoende** lengtes van parameters en sleutels. Kies **Goede** lengtes als u zeker weet dat alle clients deze lengte ondersteunen[12].
6. Kies, indien van toepassing, **Goede** elliptische krommen. Heeft u reden aan te nemen dat sommige clients deze krommen niet ondersteunen[12]? Kies dan ook **Voldoende** elliptische krommen. Ondersteun echter niet meer krommen dan nodig voor compatibiliteit.
7. Kies voor elke overige optie zijn **Goede** instelling, tenzij er, op basis van de vereiste compatibiliteit met clients, sterke overwegingen zijn om een **Voldoende** instelling te kiezen[12].
8. Configureer de server met de gekozen instellingen. De TLS-configuratie is onderdeel van de configuratie van software die de TLS-verbinding gebruikt. Wilt u bijvoorbeeld HTTPS aanbieden, dan configureert u TLS in de configuratie van de webserversoftware.
9. Bedenk welke software clients zoal gebruiken om verbinding te maken. Test of clients met deze software verbinding kunnen maken.
10. Ondervindt u compatibiliteitsproblemen? Ga na welke opties deze problemen opleveren. Meestal zijn dergelijke problemen op te lossen door een **Goede** instelling te vervangen door of aan te vullen met een **Voldoende** instelling.

Aandachtspunten

- De richtlijnen in dit document zijn van invloed op de selectie van een certificaatleverancier. Niet elke certificaatleverancier kan namelijk elk type certificaat leveren. Bespreek uw TLS-configuratie daarom met de beheerders van certificaten en public key infrastructures (PKI's) binnen uw organisatie.
- Het controleren van TLS-configuraties kan onderdeel zijn van penetratietests en de reguliere auditcyclus binnen een organisatie. Er bestaan tools en websites waarmee u ook zelf een dergelijke controle kunt uitvoeren[13]. De resultaten van een dergelijke controle kunt u vergelijken met deze richtlijnen. Zo bent u bevindingen tijdens een penetratietest of audit voor.
- Besturingssystemen bevatten vaak meerdere TLS-programmeerbibliotheken. Zorg dat u weet welke programmeerbibliotheek de serversoftware gebruikt, en zorg dat die up-to-date is.

Afwijken van het gebruiksadvies

Het NCSC adviseert om TLS-configuraties te allen tijde in te richten op basis van dit gebruiksadvies. In uitzonderlijke situaties kan het echter voorkomen dat dit niet haalbaar is. Houd daarbij het volgende in gedachten:

- Zorg voor een onderbouwde risicoanalyse voor het afwijken van deze richtlijnen. Het afwijken van deze richtlijnen zal een negatieve invloed op het beveiligingsniveau hebben. Waarom is dat in dit geval acceptabel? Hoe heeft u deze afweging gemaakt? En welke aanvullende maatregelen treft u om ontstane risico's weg te nemen? Documenteer de afwijkingen en resultaten van deze afwegingen.
- Iets is meestal beter dan niets. Ook een verbinding die door een **Onvoldoende** TLS-configuratie wordt beschermd, kan voor sommige aanvallers totaal onleesbaar zijn. Het niet kunnen voldoen aan alle richtlijnen mag geen reden zijn om TLS volledig uit te schakelen.
- Het beoordelen van een TLS-configuratie vereist uitgebreide kennis. Wijkt u af van dit gebruiksadvies, bespreek uw TLS-configuratie en de resulterende risico's dan met een inhoudelijk specialist.

12. Een overzicht van de TLS-configuraties van verschillende typen clients is te vinden op https://www.ssllabs.com/ssltest/clients.html.
13. Voorbeelden van dergelijke tools zijn SSLScan (http://sourceforge.net/projects/sslscan/), sslyze (https://github.com/iSECPartners/sslyze) en SSL Diagnos (http://sourceforge.net/projects/ssldiagnos/). Op de website Qualys SSL Labs kan een dergelijke controle ook online worden uitgevoerd (https://www.ssllabs.com/ssltest/index.html).

HOOFDSTUK 2 » GEBRUIKSADVIES

HOOFDSTUK 3

Opties en instellingen

Opties en instellingen

De getallen tussen haakjes verwijzen naar de referenties achterin.

Versies

Recentere versies van TLS zijn veiliger dan oude versies. De oudste drie versies van TLS, SSL 1.0, SSL 2.0 en SSL 3.0, zijn niet veilig meer te gebruiken. De recentste versie, TLS 1.2, biedt de beste bescherming.

Versie	Status
TLS 1.2	**Goed** (2; 3)
TLS 1.1	**Voldoende** (2; 3)
TLS 1.0	
SSL 3.0	**Onvoldoende** (3; 1)
SSL 2.0	
SSL 1.0	

Cipher suites

De veiligheid van een cipher suite hangt af van de veiligheid van de algoritmes waar hij uit bestaat. De richtlijnen over te gebruiken cipher suites vallen daarom uiteen in richtlijnen over te gebruiken algoritmes voor de vier doelstellingen.

Een **Goed** algoritme biedt minstens 128 bits security-equivalent. Een **Voldoende** algoritme biedt minstens 112 bits security-equivalent (**2; 4**).

Een **Goede** cipher suite is een cipher suite die alleen uit **Goede** algoritmes bestaat. **Goede** cipher suites voldoen per definitie aan richtlijn B2-1 t/m B2-4. **Goede** cipher suites zijn in elk geval:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_GCM_SHA256

Een **Voldoende** cipher suite is een cipher suite die bestaat uit **Voldoende** algoritmes en eventueel enkele **Goede** algoritmes. **Voldoende** cipher suites voldoen per definitie aan richtlijn B2-1 t/m B2-4. **Voldoende** cipher suites zijn in elk geval:

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

Het NCSC adviseert om cipher suites in de hierboven genoemde volgorde aan te bieden. Op die manier krijgen de snelste en veiligste cipher suites de voorkeur.

De volgorde van deze lijsten is als volgt bepaald. Ten eerste zijn cipher suites die forward secrecy bieden geplaatst boven cipher suites die dat niet bieden. Ten tweede zijn cipher suites gebaseerd op ECDSA geplaatst boven cipher suites gebaseerd op RSA (hiervoor moet dan wel een certificaat op basis van ECDSA worden gebruikt). Ten derde is sleuteluitwisseling met ECDH geplaatst boven sleuteluitwisseling met RSA. Ten vierde zijn versies van algoritmes met langere sleutels geplaatst boven versies met kortere sleutels. De cipher suites met forward secrecy en langere sleutels zijn veiliger dan cipher suites zonder forward secrecy of met kortere sleutels. De cipher suites met cryptografie op basis van elliptische krommen zijn sneller dan de cipher suites gebaseerd op RSA.

Algoritmes voor certificaatverificatie

Het verifiëren van een certificaat gebeurt met behulp van digitale handtekeningen. Om de authenticiteit van de verbinding te garanderen, moet een betrouwbaar algoritme voor certificaatverificatie gebruikt worden. Dit algoritme wordt bepaald door de certificaatleverancier. Na verificatie van het certificaat plaatst de eigenaar van het certificaat een digitale handtekening met de bijbehorende geheime sleutel op zijn bijdrage aan de sleuteluitwisseling.

Algoritme	Status
ECDSA	GOED (1; 4; 2)
RSA	
EXPORT-varianten	ONVOLDOENDE (1; 4; 2)
PSK	
Anon	
NULL	

Dit zijn de gebruikelijkste algoritmes voor certificaatverificatie. Het algoritme DSS is **VOLDOENDE**, maar zeer ongebruikelijk en daarom niet in de tabel opgenomen. Veel DSS-certificaten hebben sleutels van minder dan 2048 bits, wat in strijd is met richtlijn B3-4. Levert uw certificaatleverancier certificaten met DSS, dan is het de moeite waard een leverancier te zoeken die RSA- of ECDSA-certificaten levert.

Algoritmes voor sleuteluitwisseling

Nadat de authenticiteit van het verstrekte certificaat is vastgesteld, begint de sleuteluitwisseling. Deze vindt plaats met behulp van eerder door de partijen gekozen publieke sleutels. De DHE- en ECDHE-algoritmes bieden daarnaast de mogelijkheid om een tijdelijke (ephemeral) sleutel te gebruiken. Zie voor meer informatie over tijdelijke sleutels de sectie *Forward Secrecy* in appendix *Verdere overwegingen*.

Algoritme	Status
ECDHE	GOED (1; 4; 2)
ECDH	
RSA	
DHE	VOLDOENDE[14] (1; 4; 2)
DH	
SRP	ONVOLDOENDE (1; 4; 2)
KRB5	
PSK	
NULL	

Algoritmes voor bulkversleuteling[15]

De gegevens worden tijdens de applicatiefase versleuteld met een algoritme voor bulkversleuteling. Het bekendste algoritme voor bulkversleuteling is AES. TLS ondersteunt AES met twee sleutellengtes (128 en 256 bits) en meerdere operatiemodussen (onder meer CBC en GCM). GCM is de veiligste operatiemodus.

Algoritme	Status
AES-256-GCM	GOED (1; 2; 3)
AES-128-GCM	
AES-256-CBC	VOLDOENDE (1; 2; 3)
AES-128-CBC	
3DES-CBC	
IDEA	ONVOLDOENDE (1; 4; 2)
DES	
RC4	
NULL	

Algoritmes voor hashing

TLS gebruikt hashfuncties voor twee doeleinden: het genereren van random numbers en het uitvoeren van berichtauthenticatie in de applicatiefase. Voor beide doeleinden is het belangrijk dat de gekozen hashfunctie veilig is.

Algoritme	Status
SHA-512	GOED (1; 4; 2)
SHA-384	
SHA-256	
SHA-1	VOLDOENDE (1; 4; 2)
MD5	ONVOLDOENDE (1; 4; 2)

Let op: SHA-1 is alleen **VOLDOENDE** voor het genereren van random numbers en het uitvoeren van berichtauthenticatie. Voor het genereren van een certificaathandtekening is het **ONVOLDOENDE**. Zie ook richtlijn B3-2.

14. Er zijn meerdere implementaties van TLS die geen sleuteluitwisseling met DH(E) toestaan met publieke sleutels van tenminste 2048 bits. Gebruik van DH(E) zal daarom in veel gevallen in strijd zijn met richtlijn B4-1.
15. Dit zijn de gebruikelijkste algoritmes voor bulkversleuteling. Andere **VOLDOENDE** algoritmes zijn: CAMELLIA-128-CBC, CAMELLIA-256-CBC, SEED en ARIA. Andere **GOEDE** algoritmes zijn AES-128-CCM, AES-256-CCM, CAMELLIA-128-GCM en CAMELLIA-256-GCM. Deze zijn echter niet gebruikelijk. Ondersteunt een systeem deze algoritmes, dan is het de moeite waard na te gaan of dit wel nodig is.

RSA-sleutels

De veiligheid van RSA voor versleuteling en digitale handtekeningen hangt samen met de lengte van de gebruikte geheime sleutel.

Lengte van RSA-sleutel	Status
Minstens 3072 bits	GOED (4; 2)
Minstens 2048 bits	VOLDOENDE (1; 4; 2)
Minder dan 2048 bits	ONVOLDOENDE (4; 2)

DH(E)-parameters en DSS-sleutels

De veiligheid van Diffie-Hellman voor digitale handtekeningen (DSS) en sleuteluitwisseling (DH en DHE) hangt samen met de lengte van de gebruikte publieke en geheime sleutel.

Lengte van *publieke* sleutel voor DSS of DH(E)	Status
Minstens 2048 bits	VOLDOENDE[16] (4; 2)
Minder dan 2048 bits	ONVOLDOENDE (4; 2)

Lengte van *geheime* sleutel voor DSS of DH(E)	Status
Minstens 224 bits	VOLDOENDE[17] (4; 2)
Minder dan 224 bits	ONVOLDOENDE (4; 2)

ECDH(E)-parameters, ECDSA-sleutels en elliptische krommen

Met cryptografie op basis van *elliptische krommen* is het mogelijk om hetzelfde niveau van veiligheid te bereiken als met traditionele cryptografie, maar met een fractie van de sleutellengte. De veiligheid van een elliptische-krommensleutel van 256 bits staat gelijk aan die van een RSA-sleutel van 3072 bits.

Lengte van ECDSA-sleutel of ECDH(E)-parameter	Status
Minstens 256 bits	GOED (4; 2)
Minstens 224 bits	VOLDOENDE[18] (4; 2)
Minder dan 224 bits	ONVOLDOENDE (4; 2)

Niet alle elliptische krommen zijn geschikt om mee te rekenen in TLS. Sommige leveren erg korte sleutels op, andere zijn geoptimaliseerd voor gebruik in hardwareimplementaties en weer andere bevatten meer structuur dan nodig is. De keuzes in onderstaande tabel zijn daarop gebaseerd.

Elliptische kromme	Status
brainpoolP512r1	GOED (2)
brainpoolP384r1	
brainpoolP256r1	
secp521r1	
secp384r1	
secp256r1	
secp224r1	VOLDOENDE (2)
Andere krommen[19]	ONVOLDOENDE (2)

Overige opties

Compressie

Het gebruik van compressie kan een aanvaller informatie bieden over geheime delen van versleutelde communicatie. Een aanvaller die in staat is om een deel van de verzonden data te bepalen, kan door middel van een groot aantal verzoeken stukje bij beetje de oorspronkelijke data reconstrueren. Data die meer herhalingen bevat, is beter te comprimeren dan data die geen herhalingen bevat. De aanvaller kan daarom achterhalen of het door hem bepaalde deel vaker voorkomt in de verzonden informatie.

TLS-compressie wordt zo weinig gebruikt dat het geen kwaad kan het uit te schakelen. Voor applicatiespecifieke compressie ligt dat anders. Bij het HTTP-protocol is compressie bijvoorbeeld een middel om bandbreedte efficiënt te gebruiken.

Maak uw eigen afweging over het gebruiken van applicatiespecifieke compressie. Kiest u ervoor uw applicatie compressie te laten gebruiken, ga dan na of het mogelijk is maatregelen te treffen op applicatieniveau tegen deze aanvalstechniek. Een voorbeeld van een dergelijke maatregel is het beperken van de mate waarin een aanvaller de respons van de server kan beïnvloeden.

Compressieoptie	Status
Geen compressie	GOED (1; 3)
Applicatiespecifieke compressie	VOLDOENDE[20] (1)
TLS-compressie	ONVOLDOENDE[21] (1; 3)

16. Het gebruiken van DHE-parameters van ten minste 2048 bits kan een negatieve invloed op de prestaties van het systeem hebben.
17. Het gebruiken van DHE-parameters van ten minste 2048 bits kan een negatieve invloed op de prestaties van het systeem hebben.
18. Hoewel ECDSA-sleutels en ECDH(E)-parameters van 224 bits Voldoende zijn, is het erg ongebruikelijk om minder dan 256 bits te gebruiken. Gebruikt uw systeem of leverancier slechts 224 bits, dan is het de moeite waard na te gaan waarom dit nodig is.
19. Bedoeld worden de krommen sect163k1, sect163r1, sect163r2, sect193r1, sect193r2, sect233k1, sect233r1, sect239k1, sect283k1, sect283r1, sect409k1, sect409r1, sect571k1, sect571r1, secp160k1, secp160r1, secp160r2, secp192k1, secp192r1, secp224k1, secp256k1 en de mogelijkheid om zelf krommen te kiezen.
20. Het inschakelen van applicatiespecifieke compressie (zoals HTTP-compressie) maakt de configuratie kwetsbaar voor de BREACH-aanvalstechniek. Het uitschakelen van applicatiespecifieke compressie kan een negatieve invloed hebben op de prestaties van het systeem.
21. Het inschakelen van TLS-compressie maakt de configuratie kwetsbaar voor de CRIME-aanvalstechniek.

Renegotiation

De TLS-specificatie geeft de mogelijkheid om uit de applicatiefase te stappen en een nieuwe handshake af te dwingen. Dit verschijnsel, dat renegotiation heet, was oorspronkelijk erg onveilig uitgevoerd. Er is een reparatie aan de standaard uitgevoerd en inmiddels is veilige renegotiation mogelijk. De oude uitvoering, die inmiddels insecure renegotiation heet, moet worden uitgeschakeld. Ook is er voor zover bekend geen noodzaak om client-initiated renegotiation in te schakelen. De optie maakt een server wel vatbaar voor DDoS-aanvallen. Schakel deze optie daarom ook uit.

Insecure renegotiation	Status
Uit	GOED (1; 3)
Aan	ONVOLDOENDE (1; 3)

Client-initiated renegotiation	Status
Uit	GOED (1)
Aan	ONVOLDOENDE (1)

OCSP stapling

De client kan de geldigheid van het X.509-certificaat van de server navragen bij de certificaatleverancier die het certificaat heeft geleverd. Hij gebruikt daarvoor het OCSP-protocol. Het OCSP-protocol verschaft de certificaatleverancier informatie over clients die met de server in kwestie communiceren: dit kan een privacyrisico vormen. Een server kan de OCSP-informatie ook zelf verstrekken (OCSP stapling). Dit lost het privacyrisico op en levert ook prestatiewinst op.

OCSP stapling	Status
Aan	GOED (1; 3)
Uit	VOLDOENDE (1; 3)

HOOFDSTUK 3 » OPTIES EN INSTELLINGEN

HOOFDSTUK 4

Richtlijnen

Richtlijnen

In de richtlijnen wordt regelmatig verwezen naar instellingen die **Voldoende** of **Goed** zijn. De bedoelde instellingen zijn te vinden in het hoofdstuk *Opties en instellingen*.

Versies

Recentere versies van TLS zijn veiliger dan oude versies. De oudste versies van TLS bevatten kwetsbaarheden die niet kunnen worden gerepareerd. Deze moeten daarom worden vermeden. Een TLS-configuratie kan meerdere versies ondersteunen.

Nummer	Richtlijn
B1-1	Alle ondersteunde versies van TLS zijn **Voldoende** of **Goed**.

Cipher suites

TLS gebruikt cryptografische algoritmes voor vier verschillende doeleinden:

- Certificaatverificatie: RSA, ECDSA, DSS, etc.
- Sleuteluitwisseling: ECDH(E), DH(E), RSA, etc.
- Bulkversleuteling: AES, CAMELLIA, 3DES, etc.
- Hashing: SHA-384, SHA-256, SHA-1, MD5, etc.

Om een verbinding op te zetten gebruikt TLS voor elk van deze doeleinden een algoritme. Een volgens de TLS-standaard geldige samenstelling van algoritmes heet een cipher suite. Er zijn honderden verschillende cipher suites. Een TLS-configuratie kan meerdere cipher suites ondersteunen.

Nummer	Richtlijn
B2-1	Alle ondersteunde cipher suites bevatten een **Voldoende** of **Goed** algoritme voor certificaatverificatie.
B2-2	Alle ondersteunde cipher suites bevatten een **Voldoende** of **Goed** algoritme voor sleuteluitwisseling.
B2-3	Alle ondersteunde cipher suites bevatten een **Voldoende** of **Goed** algoritme voor bulkversleuteling.
B2-4	Alle ondersteunde cipher suites bevatten een **Voldoende** of **Goed** algoritme voor hashing.

Certificaat

TLS biedt de server de gelegenheid zijn identiteit aan te tonen met behulp van een X.509-certificaat. Alleen als de server een certificaat gebruikt, weet de client dat hij communiceert met de server en niet met een derde partij die de communicatie wil afluisteren of manipuleren. Het aanvragen en beheren van certificaten is geen onderdeel van deze richtlijnen. Zie voor enkele aanwijzingen de sectie *Certificaatbeheer* in appendix *Verdere overwegingen*.

Nummer	Richtlijn
B3-1	De server biedt een certificaat aan ter authenticatie.
B3-2	De ondertekende fingerprint van het certificaat is gemaakt met een **Goed** algoritme voor hashing.
B3-3	Als de server een certificaat aanbiedt met een RSA-sleutel, is de lengte van deze sleutel ten minste **Voldoende**.
B3-4	Als de server een certificaat aanbiedt met een DSS-sleutel, is de lengte van de de publieke sleutel ten minste **Voldoende** en de lengte van de geheime sleutel ten minste **Voldoende**.
B3-5	Als de server een certificaat aanbiedt met een ECDSA-sleutel, is de lengte van deze sleutel ten minste **Voldoende**.
B3-6	Als het eigen certificaat niet direct door het stamcertificaat (root CA) is ondertekend, biedt de server tussencertificaten (intermediate CA's) tussen het stamcertificaat (root CA) en het eigen certificaat aan ter authenticatie.

Diffie-Hellman-parameters

De cipher suite specificeert hoe de client en server samen aan een sleutel voor communicatie in de applicatiefase komen. Diffie-Hellman (DH) is een methode om samen tot een sleutel te komen. Diffie-Hellman kent ook een versie met elliptische krommen (ECDH). Zowel DH als ECDH kent een variant met een tijdelijke (ephemeral) sleutel (DHE en ECDHE). Zie voor meer informatie over tijdelijke sleutels de sectie *Forward Secrecy* in appendix *Verdere overwegingen*.

Nummer	Richtlijn
B4-1	Als gebruik wordt gemaakt van DH of DHE voor sleuteluitwisseling, dan is de lengte van de gebruikte publieke parameters ten minste **Voldoende**.
B4-2	Als gebruik wordt gemaakt van DH of DHE voor sleuteluitwisseling, dan is de lengte van de gebruikte geheime parameters ten minste **Voldoende**.
B4-3	Als gebruik wordt gemaakt van ECDH of ECDHE voor sleuteluitwisseling, dan is de lengte van de gebruikte parameters ten minste **Voldoende**.

Elliptische krommen

Van berekeningen met elliptische krommen zegt men dat ze plaats vinden ‘op’ een elliptische kromme. De kromme vormt de context waarin gerekend wordt. Om veilig te kunnen rekenen, is het gebruik van een geschikte kromme noodzakelijk. Niet elke kromme biedt dezelfde veiligheid.

Nummer	Richtlijn
B5-1	Alle gebruikte elliptische krommen zijn **Voldoende** of **Goed**.

Overige opties

Naast de al besproken opties kent TLS nog talloze andere opties. We bespreken alleen de opties die van invloed kunnen zijn op de veiligheid van TLS en die standaard niet altijd goed zijn ingesteld.

Compressie

Het gebruik van compressie kan een aanvaller informatie bieden over geheime delen van versleutelde communicatie. Omdat de data eerst gecomprimeerd en daarna pas versleuteld wordt, kan de mate van compressie informatie verschaffen over de data die wordt verzonden.

Nummer	Richtlijn
B6-1	De instellingen voor compressie zijn **Voldoende** of **Goed**.

Renegotiation

De TLS-specificatie geeft de mogelijkheid om uit de applicatiefase te stappen en een nieuwe handshake af te dwingen. Dit verschijnsel heet renegotiation.

Nummer	Richtlijn
B6-2	De instellingen voor renegotiation zijn **Voldoende** of **Goed**.

HOOFDSTUK 4 » RICHTLIJNEN

APPENDIX » VERDERE OVERWEGINGEN

Forward secrecy

Forward secrecy is een techniek om de vertrouwelijkheid van TLS-communicatie te blijven waarborgen als de geheime sleutel van een certificaat naderhand gestolen wordt. Cipher suites die DHE of ECDHE gebruiken als algoritme voor sleuteluitwisseling, bieden forward secrecy.

Wordt forward secrecy gebruikt, dan gebruiken client en server hun eigen sleutels niet direct voor bulkversleuteling. In plaats daarvan wordt een tweede sleutel afgesproken, een tijdelijke (ephemeral) sleutel, die alleen voor die sessie geldt. Naderhand worden alle gebruikte waarden verwijderd. Uit de geheime sleutel van het certificaat valt de gebruikte tijdelijke sleutel niet af te leiden.

Het scenario waar forward secrecy tegen beschermt, bestaat uit twee stappen. Eerst weet een aanvaller de door TLS beschermde communicatie af te luisteren. Naderhand achterhaalt hij de geheime sleutel van het servercertificaat, bijvoorbeeld door hacking of een rechterlijke vordering.

Certificaatbeheer

Het aanvragen en beheren van certificaten is geen onderdeel van deze richtlijnen. Desondanks is goed certificaatbeheer een belangrijke voorwaarde voor veilige inzet van TLS. Daarom sommen we enkele aandachtspunten op. Verdere aanwijzingen zijn te vinden in het factsheet 'Veilig beheer van digitale certificaten' van het NCSC[22].

- **Genereren geheime sleutel** Gebruik een goede random number generator voor het genereren van de geheime sleutel. Zorg dat u deze sleutel genereert op een vertrouwd systeem, zoals een Hardware Security Module (HSM) of een computer die geheel losgekoppeld is van het internet. Genereert u de sleutel op een losgekoppelde computer, zet deze dan daarna op de server die het certificaat aan zal bieden.
- **Certificaatleverancier** Kies een betrouwbare certificaatleverancier voor het leveren en ondertekenen van het certificaat. Nederlandse overheidspartijen kunnen gebruikmaken van certificaten van PKIoverheid en zijn daar in sommige gevallen toe verplicht.
- **Domeinnamen** Het certificaat bevat een lijst van domeinnamen (fully qualified domain names, FQDN) waarvoor het geldt. Zorg dat het certificaat alle domeinnamen noemt waarvoor het gebruikt zal worden, inclusief subdomeinen.
- **Extended validation** Veel certificaatleveranciers leveren ook Extended Validation (EV)-certificaten. Een EV-certificaat biedt meer betrouwbaarheid over de identiteit van de eigenaar, maar is vaak wel duurder dan een gewoon certificaat. Een risicoanalyse kan helpen bepalen of in uw geval een EV-certificaat gewenst is.
- **Bestanden op de server** De beheerder van de server dient tussencertificaten (intermediate CA's) tussen het stamcertificaat (root CA) en het eigen certificaat ook op de server aan te bieden. De server biedt deze ook aan bij het opzetten van een TLS-verbinding. De geheime sleutel van het eigen certificaat moet goed beschermd worden. Weet een aanvaller de geheime sleutel te achterhalen, dan kan hij onderschept TLS-verkeer inzien of manipuleren. Eventueel kan de geheime sleutel worden ondergebracht in een HSM. Een HSM biedt hardwarematige bescherming tegen het stelen van de geheime sleutel.
- **Administratie** Houd een administratie bij van alle certificaten die binnen de organisatie in gebruik zijn. Vermeld daarbij ook de verloopdatum van certificaten, zodat u deze tijdig kunt vervangen. Verlopen certificaten dienen nooit gebruikt te worden. Is het nodig het certificaat te vervangen, dan is eenvoudig terug te vinden waar het in gebruik is.

Random number generators

Om cryptografische algoritmes veilig te gebruiken, is een goede bron van entropie en een goede generator voor random numbers (pseudo-random number generator, PRNG) nodig. De bron van entropie levert willekeurige data en vormt de invoer voor de PRNG. De PRNG zet deze willekeurige data om in uniform verdeelde random numbers. Deze eis speelt in het bijzonder in twee gevallen:

- het genereren van sleutels voor certificaten, en
- het genereren van tijdelijke sleutels voor forward secrecy.

Het genereren van entropie vormt vaak een knelpunt als een server die TLS aanbiedt zwaar wordt belast. Het toevoegen van een hardwaremodule (hardwarematige random number generator) aan de server zorgt ervoor dat er altijd voldoende entropie beschikbaar is.

De meeste besturingssystemen en TLS-programmeerbibliotheken bevatten een goede random number generator. Van de volgende random number generator is bekend dat hij onveilig is:

- Dual EC DRBG[23] [24]

Het is raadzaam na te gaan of dit niet de random number generator is die uw TLS-programmeerbibliotheek standaard gebruikt.

22. https://www.ncsc.nl/dienstverlening/expertise-advies/kennisdeling/factsheets/factsheet-veilig-beheer-van-digitale-certificaten.html
23. http://csrc.nist.gov/publications/nistbul/itlbul2013_09_supplemental.pdf
24. http://dualec.org/

Waar eindigt de TLS-verbinding?

Het model van een client die met een server verbindt, klopt niet met de opstelling die veel organisaties gebruiken. Het ontsleutelen van TLS-verkeer kan bijvoorbeeld worden gecentraliseerd, waarna het ontsleutelde verkeer binnen het interne netwerk verder wordt gerouteerd. Deze opstelling geeft mogelijkheden tot het naverwerken van het netwerkverkeer. Houdt bij het gebruik in een dergelijke opstelling in gedachten dat TLS het verkeer slechts beschermt tot het punt waar het wordt ontsleuteld. Moet vertrouwelijkheid ook binnen het eigen netwerk gegarandeerd blijven, tref dan aanvullende maatregelen. Een mogelijke maatregel is om voor dit laatste stuk een nieuwe TLS-sessie te gebruiken.

Sommige bedrijven die anti-DDoS-maatregelen voor organisaties bieden, vragen om de geheime sleutels van certificaten die in gebruik zijn voor TLS. Zij kunnen dan uw verkeer ontsleutelen en filteren. Wilt u van deze diensten gebruikmaken, geef dan niet zomaar uw sleutel af. Ga na of het afgeven van de geheime sleutel niet in strijd is met intern beleid of sectorale wetgeving. Overweeg om een andere aanbieder te zoeken die het afgeven van uw sleutels niet vereist[25]. Inventariseer de risico's die komen kijken bij het afgeven van de geheime sleutel. Tref contractuele maatregelen om de verminderde technische controle te compenseren en audit regelmatig of de dienstverlener de maatregelen ook hanteert.

Certificate pinning en DANE

Een client die een TLS-sessie met een server start, controleert het X.509-certificaat van de server. De client controleert de keten van digitale handtekeningen die het certificaat verbindt met het stamcertificaat (root CA). Dit systeem is kwetsbaar, omdat de meeste software meer dan honderd stamcertificaten vertrouwt. Geeft een certificaatleverancier vervalste certificaten uit, dan komt de integriteit van het hele stelsel in gevaar.

Heeft u controle over de software van de client en de server? Dan kunt u door middel van certificate pinning vastleggen wat het enige certificaat is dat de client van de server moet accepteren. De client hoeft de keten van handtekeningen niet meer te controleren: hij herkent het certificaat, of niet. Een gecompromitteerde certificaatleverancier vormt dan geen risico voor deze verbinding. De verbinding van een mobiele app naar een server is een scenario waarin certificate pinning effectief kan zijn.

DNS-based Authentication of Named Entities (DANE) is een techniek om clients in staat te stellen de authenticiteit van een certificaat te controleren door middel van het Domain Name System (DNS). De beheerder van een certificaat publiceert informatie over dat certificaat in een speciaal DNS-record, een TLSA-record. Clients kunnen het certificaat nu niet alleen via de certificaatautoriteit, maar ook via het TLSA-record controleren op authenticiteit. Merk op dat het traditionele DNS-stelsel niet betrouwbaar genoeg is om DANE veilig te gebruiken. Gebruik van DNSSEC is daarvoor noodzakelijk.

25. Zie voor een voorbeeld van een techniek om dit te bereiken: https://blog.cloudflare.com/keyless-ssl-the-nitty-gritty-technical-details/.

APPENDIX » VERDERE OVERWEGINGEN

APPENDIX » NAMEN VAN CIPHER SUITES IN OPENSSL EN GNUTLS

Status	Naam volgens IANA[26]	Naam in OpenSSL[27]	Naam in GnuTLS[28]
GOED	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	ECDHE-ECDSA-AES256-GCM-SHA384	TLS_ECDHE_ECDSA_AES_256_GCM_SHA384
	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	ECDHE-ECDSA-AES128-GCM-SHA256	TLS_ECDHE_ECDSA_AES_128_GCM_SHA256
	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDHE-RSA-AES256-GCM-SHA384	TLS_ECDHE_RSA_AES_256_GCM_SHA384
	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDHE-RSA-AES128-GCM-SHA256	TLS_ECDHE_RSA_AES_128_GCM_SHA256
	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	ECDH-ECDSA-AES256-GCM-SHA384	-
	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	ECDH-ECDSA-AES128-GCM-SHA256	-
	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384	ECDH-RSA-AES256-GCM-SHA384	-
	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256	ECDH-RSA-AES128-GCM-SHA256	-
	TLS_RSA_WITH_AES_256_GCM_SHA384	AES256-GCM-SHA384	TLS_RSA_AES_256_GCM_SHA384
	TLS_RSA_WITH_AES_128_GCM_SHA256	AES128-GCM-SHA256	TLS_RSA_AES_128_GCM_SHA256
VOLDOENDE	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	ECDHE-ECDSA-AES256-SHA384	TLS_ECDHE_ECDSA_AES_256_CBC_SHA384
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	ECDHE-ECDSA-AES128-SHA256	TLS_ECDHE_ECDSA_AES_128_CBC_SHA256
	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	ECDHE-ECDSA-AES256-SHA	TLS_ECDHE_ECDSA_AES_256_CBC_SHA1
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	ECDHE-ECDSA-AES128-SHA	TLS_ECDHE_ECDSA_AES_128_CBC_SHA1
	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	ECDHE-ECDSA-DES-CBC3-SHA	TLS_ECDHE_ECDSA_3DES_EDE_CBC_SHA1
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDHE-RSA-AES256-SHA384	TLS_ECDHE_RSA_AES_256_CBC_SHA384
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	ECDHE-RSA-AES128-SHA256	TLS_ECDHE_RSA_AES_128_CBC_SHA256
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	ECDHE-RSA-AES256-SHA	TLS_ECDHE_RSA_AES_256_CBC_SHA1
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDHE-RSA-AES128-SHA	TLS_ECDHE_RSA_AES_128_CBC_SHA1
	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	ECDHE-RSA-DES-CBC3-SHA	TLS_ECDHE_RSA_3DES_EDE_CBC_SHA1
	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	ECDH-ECDSA-AES256-SHA384	-
	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	ECDH-ECDSA-AES128-SHA256	-
	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	ECDH-ECDSA-AES256-SHA	-
	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	ECDH-ECDSA-AES128-SHA	-
	TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	ECDH-ECDSA-DES-CBC3-SHA	-
	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	ECDH-RSA-AES256-SHA384	-
	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	ECDH-RSA-AES128-SHA256	-
	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	ECDH-RSA-AES256-SHA	-
	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA	ECDH-RSA-AES128-SHA	-
	TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	ECDH-RSA-DES-CBC3-SHA	-
	TLS_RSA_WITH_AES_256_CBC_SHA256	AES256-SHA256	TLS_RSA_AES_256_CBC_SHA256
	TLS_RSA_WITH_AES_128_CBC_SHA256	AES128-SHA256	TLS_RSA_AES_128_CBC_SHA256
	TLS_RSA_WITH_AES_256_CBC_SHA	AES256-SHA	TLS_RSA_AES_256_CBC_SHA1
	TLS_RSA_WITH_AES_128_CBC_SHA	AES128-SHA	TLS_RSA_AES_128_CBC_SHA1
	TLS_RSA_WITH_3DES_EDE_CBC_SHA	DES-CBC3-SHA	TLS_RSA_3DES_EDE_CBC_SHA1

26. http://www.iana.nl/assignments/tls-parameters/tls-parameters.xhtml
27. https://www.openssl.org/docs/apps/ciphers.html
28. http://gnutls.org/manual/html_node/Supported-ciphersuites.html

APPENDIX » WIJZIGEN VAN DEZE RICHTLIJNEN

Wijzigen van deze richtlijnen

Deze richtlijnen zullen worden aanpast aan nieuwe versies van TLS en nieuwe inzichten over de (on)veiligheid van bepaalde instellingen. De veiligheid van TLS is een voortdurend onderwerp van onderzoek. De komende jaren zullen er dan ook meer kwetsbaarheden in het protocol worden gevonden. Ook zullen er nieuwe versies van TLS worden opgesteld.

Acute wijzigingen

Is een acute wijziging nodig van deze richtlijnen, dan zal deze worden uitgebracht als addendum op de recentste versie van de richtlijnen. Dit kan bijvoorbeeld gebeuren als uit onderzoek blijkt dat een bepaalde TLS-instelling niet meer veilig is.

Een addendum zal worden gepubliceerd op de website van het NCSC en worden verzonden aan partners van het NCSC. Daarnaast zal de publicatie van een addendum worden gemeld via het Twitteraccount van het NCSC (@ncsc_nl) of een tegen die tijd geëigend medium.

Nieuwe versies

Grotere wijzigingen worden doorgevoerd in nieuwe versies van deze richtlijnen. Een nieuwe versie van de richtlijnen bevat ook de informatie die in eerdere addenda is uitgebracht. Nieuwe versies worden op dezelfde manier verspreid als addenda: via de website van het NCSC, verzonden aan NCSC-partners en via het Twitteraccount van het NCSC.

APPENDIX » VERKLARENDE WOORDENLIJST

3DES Zie Bulkversleuteling

AES Zie Bulkversleuteling

Bulkversleuteling Bulkversleuteling is het proces waarbij data tijdens de applicatiefase wordt versleuteld met de tijdens de handshake afgesproken sleutel. De versleuteling vindt plaats met een algoritme voor symmetrische encryptie. Bekende voorbeelden van zulke algoritmen zijn AES, 3DES en CAMELLIA. DES en RC4 zijn verouderde algoritmes voor bulkversleuteling.

CA Zie Certificaatverificatie

CAMELLIA Zie Bulkversleuteling

CBC Zie Operatiemodus

Certificaat Zie Certificaatverificatie

Certificaatverificatie De server biedt de client tijdens een TLS-sessie een certificaat aan. Dit certificaat is digitaal ondertekend door een certificaatautoriteit (CA). Een certificaatautoriteit is een door de client vertrouwde partij. De client controleert de digitale handtekening van de certificaatautoriteit. Hij weet daardoor of het certificaat inderdaad door de certificaatautoriteit is uitgegeven. Het algoritme voor certificaatverificatie is het algoritme waarmee de certificaatautoriteit zijn digitale handtekening heeft geplaatst. Bekende voorbeelden zijn RSA, DSS en ECDSA.

Cipher suite Een cipher suite is een samenstelling van algoritmes voor gebruik in TLS. Een cipher suite bevat voor vier bewerkingen een algoritme: certificaatverificatie, sleuteluitwisseling, bulkversleuteling en hashing. De client en de server spreken met TLS samen een cipher suite af voor ze versleuteld gaan communiceren.

DDoS-aanval Een Distributed Denial of Service (DDoS)-aanval is een aanval waarbij een computer met een stortvloed aan verzoeken onbeschikbaar wordt gemaakt. De verzoeken zijn afkomstig van niet een, maar een groot aantal computersystemen.

DES Zie Bulkversleuteling

DH Zie Sleuteluitwisseling

DHE Zie Sleuteluitwisseling

Diffie-Hellman Zie Sleuteluitwisseling

DNS Het Domain Name System (DNS) is een gedistribueerd systeem voor het beantwoorden van informatie-verzoeken over domeinnamen. Een typisch verzoek kan zijn wat het IP-adres van een computer met een bepaalde domeinnaam is, of welke computer de e-mail voor een bepaalde domeinnaam afhandelt. Met behulp van DNS Security Extensions (DNSSEC) kan de betrouwbaarheid van informatie in DNSSEC beter worden gewaarborgd. DNSSEC maakt het gebruik van DNS voor nieuwe toepassingen mogelijk.

DNSSEC Zie DNS

DSS Zie Certificaatverificatie

ECDH Zie Sleuteluitwisseling

ECDHE Zie Sleuteluitwisseling

ECDSA Zie Certificaatverificatie

ECC Zie Elliptische krommen

Elliptische krommen Een elliptische kromme is een wiskundige structuur waarmee gerekend kan worden. Een ander voorbeeld van een dergelijke wiskundige structuur zijn de gehele getallen. Met behulp van elliptische krommen is het mogelijk om cryptografische berekeningen uit te voeren. Zulke cryptografie noemen we Elliptic Curve Cryptography (ECC). ECDSA, ECDHE en ECDH zijn algoritmes op basis van ECC.

Elliptic curve Zie Elliptische krommen

Forward Secrecy Forward secrecy is een techniek om de vertrouwelijkheid van TLS-communicatie te blijven waarborgen als de geheime sleutel van een certificaat naderhand gestolen wordt. Cipher suites die DHE of ECDHE gebruiken als algoritme voor sleuteluitwisseling, bieden forward secrecy.

GCM Zie Operatiemodus

Geheime sleutel Zie Sleutel

Handshake De handshake is de fase van het TLS-protocol waarin de client en server afspraken maken over de manier waarop ze gegevens uit zullen gaan wisselen. Na de handshake volgt de applicatiefase, waarin client en server gegevens versleuteld uitwisselen.

Hashfunctie Een hashfunctie is een wiskundige functie die gegeven data verhaspelt tot een digitale vingerafdruk. Uit het resultaat is de invoer in het algemeen niet meer af te leiden. Hashfuncties worden in TLS gebruikt voor het genereren van toevalsgetallen en voor het authenticeren van berichten tijdens de applicatiefase. Voorbeelden van hashfuncties zijn MD5, SHA-1, SHA-256, SHA-384 en SHA-512.

Hashing Zie Hashfunctie

HTTPS HTTP Secure (HTTPS) is een protocol dat bestaat uit het opzetten van een TLS-sessie waarover HTTP-verkeer wordt uitgewisseld. Communicatie met een webserver is op die manier niet in te zien of te manipuleren.

IETF De Internet Engineering Task Force is een orgaan dat verantwoordelijk is voor het opstellen van internetstandaarden. Deze internetstandaarden worden geformuleerd in zogeheten Requests For Comments (RFC's). De IETF heeft geen bevoegdheid om het gebruik van de gestelde standaarden af te dwingen.

Key Zie Sleutel

Library Zie Programmeerbibliotheek

MD5 Zie Hashfunctie

Operatiemodus Een algoritme voor bulkversleuteling kan werken op blokken data (block cipher) of op de stroom data (stream cipher). Bij gebruik van een block cipher dienen de versleutelde blokken op een veilige manier aaneengeregen te worden. De operatiemodus is de manier waarop deze blokken aaneengeregen worden. Voorbeelden van operatiemodussen zijn CBC en GCM.

PKI Zie Public Key Infrastructure

Private key Zie Sleutel

Programmeerbibliotheek Een programmeerbibliotheek is software die bepaalde functionaliteit beschikbaar stelt voor programmeurs van andere software. Door het gebruik van een programmeerbibliotheek kan een programmeur voortbouwen op het werk van anderen. Hij hoeft niet alle functionaliteit zelf te bouwen. TLS wordt in software meestal gebruikt in de vorm van een programmeerbibliotheek.

Public key Zie Sleutel

Public Key Infrastructure Een Public Key Infrastructure (PKI) is een hiërarchische ordening van certificaten waarbij de hogere certificaten de authenticiteit van de lagere certificaten bevestigen met een digitale handtekening. Vertrouwt een client de hoogste certificaten in de PKI, dan kan hij ook vertrouwen op de lagere certificaten door de tussenliggende digitale handtekeningen te controleren. De certificaten die een certificaatautoriteit uitgeeft, samen genomen met het stamcertificaat (root CA), vormen een PKI.

Publieke sleutel Zie Sleutel

RC4 Zie Bulkversleuteling

RSA RSA is een algoritme voor sleuteluitwisseling en voor certificaatverificatie. Zie aldaar.

Security-equivalent Security-equivalent is een maat om de cryptografische sterkte van versleutelingsmethoden te vergelijken. Security-equivalent wordt uitgedrukt in bits. De sterkte van een versleutelingsmethode hangt af van het gebruikte algoritme, de sleutellengte en de stand der techniek voor het kraken van de methode. Bijvoorbeeld: ECDSA met een sleutellengte van 256 bits en AES met een sleutellengte van 128 bits hebben allebei een security-equivalent van 128 bits.

SHA-1 Zie Hashfunctie

SHA-256, SHA-384, SHA-512 Zie Hashfunctie

Sleutel Een sleutel is een stuk geheime data waarmee cryptografische berekeningen uitgevoerd kunnen worden. Zijn gegevens bijvoorbeeld versleuteld, dan kunnen ze met behulp van de bijbehorende sleutel ontsleuteld worden. In symmetrische algoritmes voor versleuteling is de hele sleutel geheim. In asymmetrische algoritmes voor versleuteling bestaat de sleutel uit twee delen, een publiek deel en een geheim deel. Het publieke deel van de sleutel heet ook wel de publieke sleutel. Dit deel wordt niet geheim gehouden. Het geheime deel van de sleutel heet dan de geheime sleutel.

Sleuteluitwisseling De client en server in een TLS-sessie hebben een sleutel nodig om bulkversleuteling te kunnen doen. Het uitwisselen van een sleutel gebeurt met behulp van een algoritme voor sleuteluitwisseling. Hiervoor is een speciaal algoritme nodig omdat de verbinding tijdens de handshake nog niet versleuteld is. Voorbeelden van algoritmes voor sleuteluitwisseling zijn RSA, DH, DHE, ECDH en ECDHE.

SSL Secure Sockets Layer (SSL) is de oude naam voor Transport Layer Security (TLS). Hoewel TLS al sinds versie TLS 1.0 (1999) geen SSL meer heet, wordt de naam nog steeds veel gebruikt.

VPN Een Virtual Private Network (VPN) is een netwerk dat bestaat uit computers die onderling verbonden zijn via niet-vertrouwde verbindingen. Door het toepassen van versleuteling kunnen de computers onderling toch in vertrouwen gegevens uitwisselen.

Referenties

1. **Qualys, Inc.** SSL/TLS Deployment Best Practices. *Qualys SSL Labs*. [Online] 17 september 2013. [Citaat van: 1 augustus 2014.] https://www.ssllabs.com/projects/best-practices/.

2. **European Union Agency for Network and Information Security.** 2013 recommendations. *Algorithms, Key Sizes and Parameters Report*. [Online] oktober 2013. [Citaat van: 1 augustus 2014.] https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report.

3. **National Institute of Standards and Technology.** Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. *NIST Special Publication 800-52*. [Online] april 2014. [Citaat van: 1 augustus 2014.] http://csrc.nist.gov/publications/PubsSPs.html.

4. **National Institute of Standards and Technology.** Recommendation for Key Management - Part 1: General (Revision 3). *NIST Special Publication 800-57*. [Online] juli 2012. [Citaat van: 1 augustus 2014.] http://csrc.nist.gov/publications/PubsSPs.html.

Colofon

Uitgave
Nationaal Cyber Security Centrum, Den Haag
Turfmarkt 147 | 2511 DP Den Haag
Postbus 117 | 2501 CC Den Haag

T 070-751 55 55
F 070-322 25 37

E info@ncsc.nl
I www.ncsc.nl

November 2014

Nationaal Cyber Security Centrum
Het Nationaal Cyber Security Centrum (NCSC) draagt via samenwerking tussen bedrijfsleven, overheid en wetenschap bij aan het vergroten van de weerbaarheid van de Nederlandse samenleving in het digitale domein.

Het NCSC ondersteunt de Rijksoverheid en organisaties met een vitale functie in de samenleving met het geven van expertise en advies, response op dreigingen en het versterken van de crisisbeheersing. Daarnaast voorziet het in informatie en advies voor burger, overheid en bedrijfsleven ten behoeve van bewustwording en preventie. Het NCSC is daarmee het centrale meld- en informatiepunt voor ICT-dreigingen en -veiligheidsincidenten.

Nationaal Cyber Security Centrum
Turfmarkt 147 | 2511 DP Den Haag
Postbus 117 | 2501 CC Den Haag

T 070-751 55 55
F 070-322 25 37

E info@ncsc.nl
I www.ncsc.nl

November 2014