**HTTP Strict Transport Security (HSTS)**

HSTS is een web security policy mechanisme dat web cliënten beschermt tegen veel voorkomende aanvallen. Het is een extra beveiliging bovenop de voorzieningen die reeds geïmplementeerd werden in browsers. Het is ontwikkeld met de bedoeling van onervaren internetgebruikers extra bescherming te bieden, omdat zij de grootste doelgroep vormen voor hackers.

Indien de beste praktijken gevolgd worden tijdens het surfen op het internet, zou iedere web cliënt moeten controleren dat hij of zij verbonden is over een geëncrypteerde verbinding. Daarbovenop moet de legitimiteit van de website en encryptie gecontroleerd worden, dit kan men doen door het bijhorende certificaat te bestuderen. Indien dat certificaat ondertekend is door een herkende CA en het geldig is voor het betreffende (sub)domein is dit in orde.

Indien een website 1 of meerdere van deze praktijken niet goed opvolgt, zal de internetbrowser dit dikwijls aangeven door een duidelijk zichtbare waarschuwing. De browser kan eventueel ook het verkeer met die website blokkeren. Toch is het zo dat veel onervaren internetgebruikers deze waarschuwingen negeren, en indien nodig, de blokkering opheffen om verder te kunnen met minimaal tijdsverlies.

HSTS is hiervoor de oplossing. Het is een policy dat voorzien kan worden per domein om dergelijke problemen te voorkomen. Indien dit policy gebruikt wordt zal de browser zelf enkele beslissingen nemen in de plaats van de gebruiker. De websites die vallen onder die domeinen zullen enkel over een veilige, correcte HTTPS-verbinding bereikt worden, indien de cliënt gebruik maakt van een browser met HSTS ondersteuning.

HSTS checkt dat de voordien genoemde beste praktijken in orde zijn. Indien de gebruiker een HTTP-verbinding probeert op te stellen zal de browser dit vertalen naar een HTTPS-verbinding alvorens hij de GET request naar de betreffende webserver stuurt. Ook het certificaat van die HTTPS-verbinding zal geldig moeten zijn alvorens de browser zal toelaten van de connectie met de webserver op te zetten. De user hoeft in dit geval zelf geen beslissingen te maken of acties te ondernemen, de browser kan de veiligheid in zijn plaats garanderen.

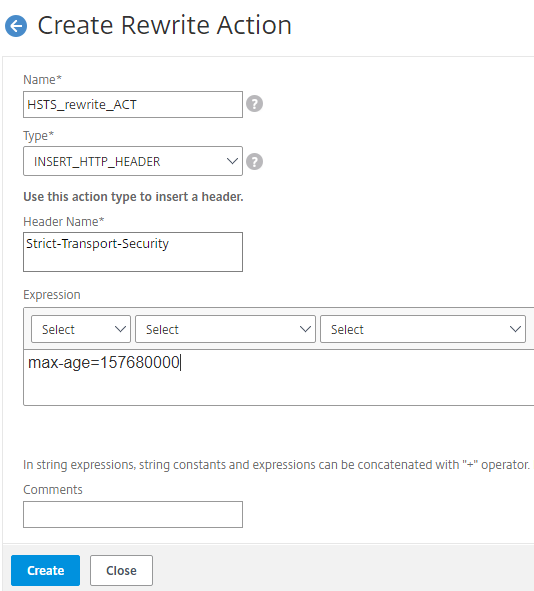
**Praktisch**

Een overzicht van de theoretisch achtergrond van HSTS kan je terugvinden in hoofdstuk xxx.

Er zijn twee verschillende implementaties van HSTS mogelijk. Eerst en vooral zullen sommige browsers een lijst van domeinnamen bezitten waarvoor de eigenaar het HSTS-policy wilt opleggen. Anderzijds is het mogelijk van een HSTS-header te voorzien in de HTTP-response. De eerste methode is niet schaalbaar en daarom kan die niet gebruikt worden om HSTS te voorzien voor elk domein dat dat hier gebruik van wilt maken. De tweede methode bezit een kwetsbaarheid die misbruikt kan worden door SSL-STRIP versie twee toe te passen. Indien de cliënt voor het eerst verbinding wilt maken met een bepaalde website, zal hij een DNS-aanvraag moeten sturen. Deze kan gemanipuleerd worden waardoor de request nooit terecht komt bij de correcte webserver en de HSTS-header nooit zal toegepast worden.

Hou er ook rekening mee dat HSTS geen garantie is voor een veilige internetverbinding, er zijn talloze andere kwetsbaarheden die de connectie tussen de cliënt en server onveilig kunnen maken.

Op basis van de opstelling die hier werd opgezet is het niet meer nodig om HSTS in te schakelen voor de publieke domeinnaam die gebruikt werd voor onze NetScaler en StorageZone. Dat komt omdat we er reeds voor gekozen hebben om de StorageZone te adverteren op basis van zijn secure HTTPS-domeinnaam, en de NetScaler van het nodige wildcard certificaat voorzien hebben. ShareFile zal dus altijd een HTTPS-connectie opzetten met NetScaler. Indien dat niet het geval was en men de optie voor HTTP-verbindingen zou openlaten, kan het handig zijn om verder na te denken over het implementeren van HSTS.



In NetScaler kan elke response van een HSTS-header regel voorzien worden, door de nodige rewrite actie aan te maken en toe te passen. Zo een rewrite actie bevat een HSTS-header regel die voorzien wordt van een maximumleeftijd. De maximumleeftijd wordt vermeld in seconden, het getal wordt dus enorm groot indien de regel jaren geldig moet blijven in de cliënten hun browser. In de afbeelding hierboven is dat 5 jaar. Deze header regelen heeft als naam “Strict-Transport-Security” en is van het type HTTP\_HEADER.

**SSL renegotiation**

SSL renegotiation is een proces waarbij de cliënt en server opnieuw een handdruk uitvoeren over een reeds bestaande TCP-verbinding. Dit is dikwijls het geval in webshops. Wanneer een gebruiker door wilt gaan naar de betaling voor enkele artikels die anoniem zijn toegevoegd aan het winkelmandje, moet hij authentiseren met zijn account voor de website. Dit doet hij door aan te melden en op die manier begaan cliënt en server opnieuw een handdruk. Omdat de gegevens uit dat winkelmandje bewaard moeten worden in de nieuwe TCP-verbinding, mag de oude verbinding niet verbroken worden. Het kan ook zijn dat om bepaalde redenen de uitwisselingseigenschappen van de verbinding aangepast moeten worden, bijvoorbeeld het gebruik van een ander algoritme. Met SSL renegotiation kan dit ook on the fly gebeuren, zonder de verbinding te moeten verbreken.

**Praktisch**

In hoofdstuk xxx kan extra theoretische achtergrond gevonden worden over SSL renegotiation.