Privacy-enhancing proxy

Michael Koopman s1401335 en Sven Konings s1534130

# Inleiding

Tijdens de challenge van de eerste week hebben wij een privacybevorderende proxy gemaakt. In dit document beschrijven we de veranderingen die wij hebben gemaakt, ten opzichte van de versie die op Blackboard stond.

# onRequest

Op het moment, dat de client een webpagina wordt opgevraagd, worden de volgende headers aangepast:

* Accept-Encoding: Om ervoor te zorgen, dat we de tekst die teruggestuurd wordt , ook uit kunnen lezen, geven we aan, dat we alleen deflate accepteren als encoding. Dit werd ons door een studentassistent aanbevolen. Helaas werken sommige sites hierdoor niet. Mocht de webserver geen deflate ondersteunen, dan wordt opnieuw een request gedaan, met normale encoding. Scripts en iFrames en sommige advertenties kunnen door de nieuwe request niet meer geblokkeerd worden op die website.
* User-Agent: Wij sturen nu alleen mee, dat we een Mozilla/5.0 client hebben, met Trident/7.0 als layout-engine. Het eerste is niet privacygevoelig, aangezien vrijwel alle browsers dit sturen. Dat laatste is nodig voor Google, anders kunnen we niets googelen.
* UA-CPU: We geven nu niet langer door welke CPU we gebruiken. Voor de meeste websites is het absoluut niet van belang welke processor je hebt.
* Referer: Is nu leeg, websites kunnen niet langer volgen waar je vandaan komt.
* Cookie: Stuurt nu niets meer mee. We geven hier gemak op voor privacy, doordat sites nu niet meer weten of je ze al eerder hebt bezocht, of dat je al ingelogd was.
* Host: Filtert een aantal advertenties eruit. De gefilterde hosts zijn alles waarin de volgende Strings voorkomen: “google-analytics”, “googleads”, “googletagmanager”, “connect.facebook.net” en “shackle.nl”.

# onResponse

In deze methoden worden alle scripts en iframes eruit gefilterd. De methode die hiervoor gebruikt wordt, heeft de naam removeSubString.

??TODO: Wat is gevoelige javascript