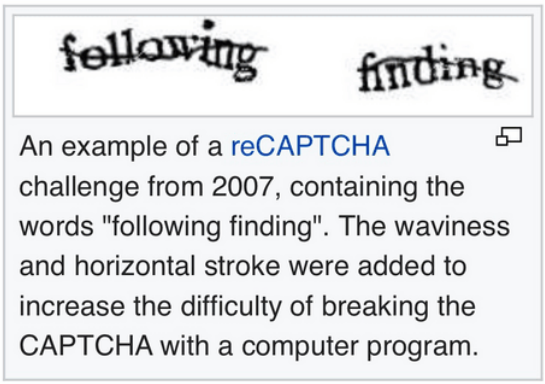
CAPTCHA Alternatief

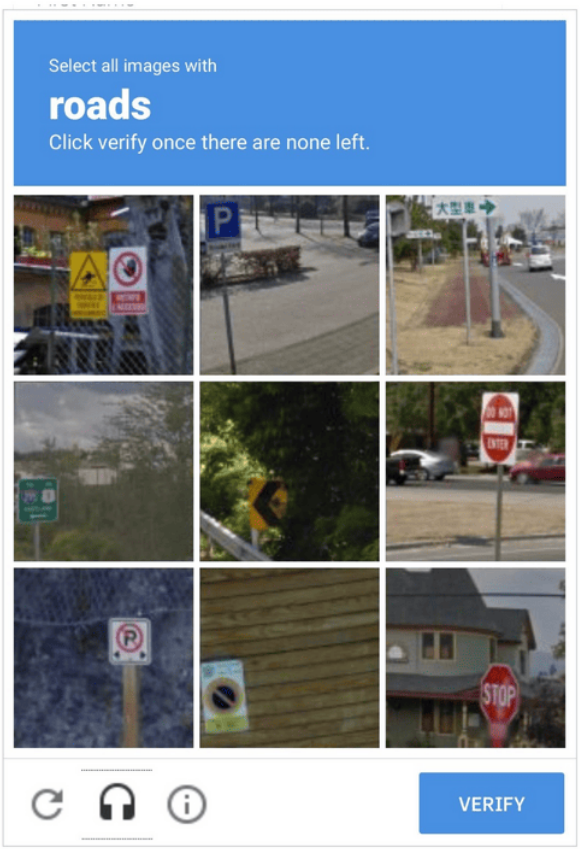
Bescherm je formulieren op een gebruiksvriendelijke manier

Willem L. Middelkoop
30 juli 2022



An example of a reCAPTCHA challenge from 2007, containing the words "following finding". The waviness and horizontal stroke were added to increase the difficulty of breaking the CAPTCHA with a computer program.

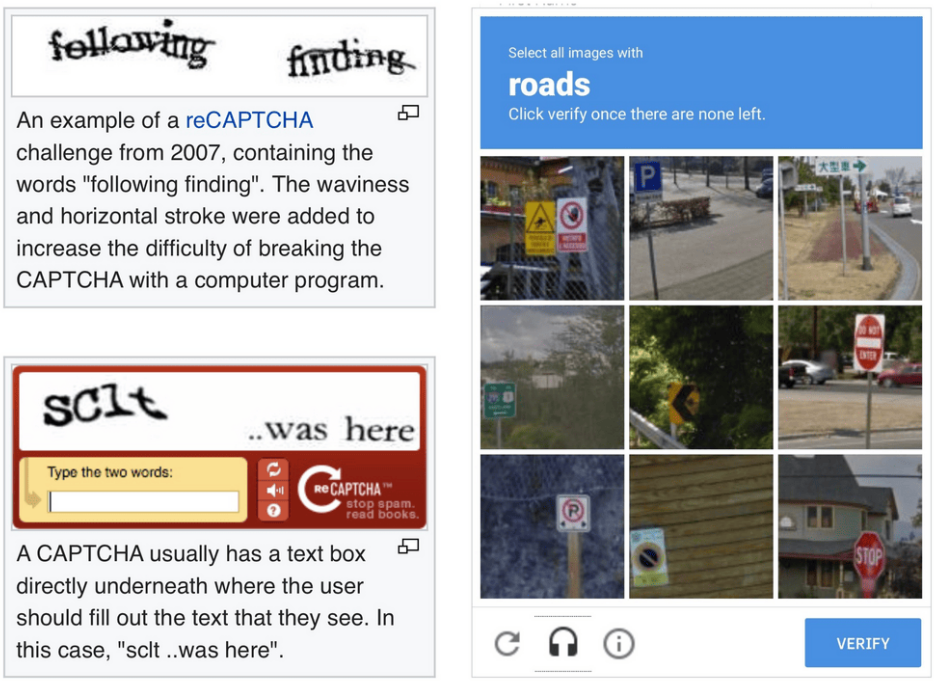
A CAPTCHA usually has a text box directly underneath where the user should fill out the text that they see. In this case, "sclt ..was here".



De kans is groot dat je wel eens een CAPTCHA bent tegengekomen: ze beschermen webformulieren door je te vragen vreemd weergegeven tekens over te typen of door je te vragen foto's te selecteren die een bepaald ding bevatten. Waarom worden ze gebruikt en is er een gebruiksvriendelijk alternatief?

Wat is CAPTCHA?

CAPTCHA's zijn ontworpen om computers en mensen van elkaar te onderscheiden en vereisen vaak dat je iets doet wat computers niet gemakkelijk kunnen. Vandaar dat de afkorting staat voor: Completely Automated Public Turing test to tell Computers and Humans Apart. Webbouwers gebruiken CAPTCHA's om hun webformulieren te beschermen tegen geautomatiseerde spam en misbruik.

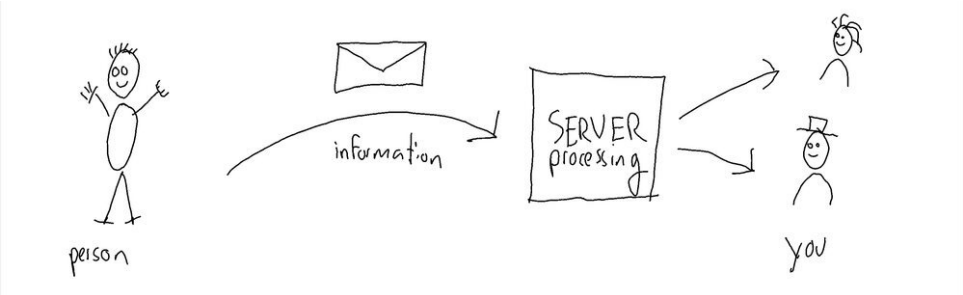


Typische CAPTCHA's die je kunt tegenkomen op het web (afbeelding: Wikipedia en Google)

Waarom heb je formbescherming nodig?

Een typisch webformulier gebruikt twee stappen:

- **informatie invoeren:** Je voert wat informatie in en klikt vervolgens op ”VERZENDEN” (of iets dergelijks) om door te gaan.
- **informatie verwerken:** De informatie die je hebt ingevoerd wordt verwerkt door een server, die je bericht of verzoek mogelijk doorstuurt naar iemands e-mail.



Een typisch webformulier: server verwerkt informatie van een persoon en stuurt deze door naar andere personen, zoals uzelf. Kwaadwillende actoren kunnen de afzender gemakkelijk vervalsen, waardoor geautomatiseerd misbruik van uw verwerkingsserver mogelijk wordt.

Het probleem met deze typische aanpak is dat de eerste stap gemakkelijk kan worden geautomatiseerd door kwaadwillende actoren. Ze gebruiken een andere computer om herhaaldelijk informatie naar de server te verzenden, waardoor deze gedwongen wordt om

ladingen informatie te verwerken, vaak met spamberichten. Hackers zijn vooral geïnteresseerd in het vinden van webformulieren waar ze het "AAN"-adres kunnen manipuleren, omdat ze dan jouw server kunnen gebruiken om spamberichten naar iedereen ter wereld te verzenden.

Een CAPTCHA beschermt je formulier door te voorkomen dat een computer automatisch informatie naar je server verzendt.

Problemen met CAPTCHA

Eerlijk gezegd vind ik ze lelijk en omslachtig, en ik ben niet alleen. Ze introduceren een drempel voor mensen: verlaging van conversieratio's, click-through rates, enz. Bovendien zijn ze vaak erg onvriendelijk voor mensen met suboptimaal zicht.

Critici klagen ook over Google als een van de meest populaire leveranciers van gratis CAPTCHA-tests. Men kan zich afvragen waarom ze zo'n dienst gratis aanbieden? Het is een uitstekende manier voor hen om in de loop van veel (commerciële) klantreizen te komen en hun algoritmen te voeden met meer gegevens om mensen te profileren. Het is iets om te overwegen als je privacy en AVG/cookie wetgeving waardeert.

Alternatief

Daarom ben ik begonnen met het creëren van een effectief alternatief voor CAPTCHA's om te gebruiken op de webformulieren die ik bouw voor mijn klanten. Mijn aanpak is tweeledig:

- **verbergen:** de meeste kwaadwillende actoren zoeken naar mogelijke doelen door te zoeken naar elke FORM-tag, die vaak wordt gebruikt om webformulieren te bouwen. Mijn formulieren hebben geen FORM-tag, maar gebruiken andere veelvoorkomende elementen zoals DIV: een perfecte camouflage. Als ze het niet kunnen vinden, kunnen ze het niet kraken.
- **versterken:** mijn webformulieren gebruiken een extra stap om het moeilijker te maken om hun misbruik te automatiseren. Ik noem het "een postzegel halen" voor het verzenden.

Cryptografische pre-submit handtekening: de postzegel

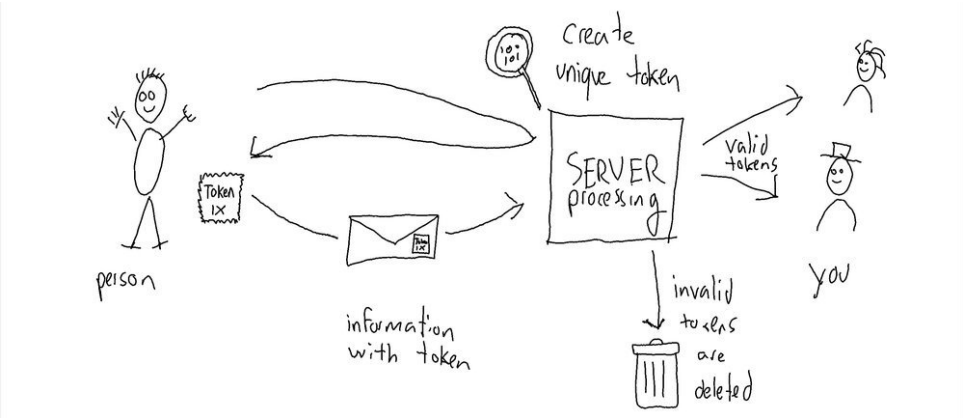
Je moet begrijpen dat kwaadwillende actoren economisch denken: ze zullen alleen zoveel moeite doen als de potentiële winst het waard is. Als je het hen moeilijk genoeg maakt om je webformulier te misbruiken, is de kans groot dat ze verder gaan en er een punt achter zetten. Voordat ik uitleg wat ik heb gedaan, moet ik je waarschuwen:

Disclaimer: *Je moet het doel van je formbescherming overwegen, in mijn geval is het (aanzienlijk) verminderen van het aantal spamberichten met behoud van gebruiksvriendelijkheid. Dit stelt me in staat om na te denken over een systeem dat niet noodzakelijkerwijs "bomvrije" bescherming biedt, maar als je nucleaire geheimen probeert te beschermen, is deze aanpak misschien niet geschikt voor jou. Overweeg altijd je dreigingsmodel!*

Om webformulieren te beschermen introduceer ik een extra stap die ik "een postzegel halen" heb genoemd. De stappen van het beschermde webformulier worden dan als volgt:

- **informatie invoeren:** Je voert wat informatie in en klikt vervolgens op "VERZENDEN" (of iets dergelijks) om door te gaan.

- **postzegel halen:** Vlak voordat de daadwerkelijke informatie wordt verzonden, neemt je apparaat contact op met de server om een cryptografische token aan te vragen die overeenkomt met de kenmerken van de daadwerkelijke informatie *en* de afzender. Dit gebeurt in slechts milliseconden, mensen zullen het niet merken. Beschouw de token als een postzegel, specifiek voor het pakket dat je verzendt - slechts eenmaal te gebruiken.
- **informatie verwerken:** De informatie die je hebt ingevoerd wordt gecombineerd met de cryptografische token en naar de server verzonden. Als de token geldig is, wordt de informatie zoals gewoonlijk verwerkt. Ongeldige tokens worden simpelweg genegeerd, wat belangrijk is om reverse engineering erg moeilijk te maken door geen feedback te geven (de server zegt altijd "OK" :-))!



Door een uniek, eenmalig token te introduceren voor elk bericht, wordt het voor kwaadwillende actoren veel moeilijker om het verzenden van berichten te automatiseren - dit beschermt uw formulier tegen spammers.

Sinds 2016 gebruik ik deze aanpak en de post-processing servers hebben letterlijk miljoenen berichten verwerkt, met extreem weinig berichten (<100) die spam zijn. Bij nadere inspectie is het meeste spam dat doorkomt "handwerk": mogelijk een indicatie van kwaadwillende actoren die het mechanisme onderzoeken. Omdat het te veel moeite lijkt om het systeem te kraken, is het zeer effectief in het bereiken van mijn doel om de webformulieren te beschermen.

Conclusie

Hoewel CAPTCHA's begrijpelijkerwijs erg populair zijn, is het niet altijd nodig om ze op je webformulieren te hebben als je alternatieven overweegt. Het is mijn advies om het moeilijker te maken voor kwaadwillende actoren en gemakkelijker voor je publiek!