**Intro**

Wie weet er precies wat er gebeurd als je je paswoord invult als je inlogt op je email?

Ik kan je al zeggen Niemand weet precies wat er gebeurd als je je wachtwoord op je email.

Ik ben Alex Jonkman en ik doe mijn presentatie over Wachtwoord encryptie. En het opslaan van wachtwoorden.

Deze presentatie zal zo’n 5 minuten duren waarna er nog tijd is om vragen te stellen.

**Inhoudsopgave**

De meeste weten het nut van een wachtwoord wel ongeveer, maar eerst ga ik uitleggen wat het nut is van een wachtwoord.

Daarna over het opslaan van een wachtwoorden door gebruik te maken van basis Algoritmes.

**Het Wachtwoord**

Het wachtwoord verschilt heel erg met andere data wat opgeslagen wordt. Meeste data wordt opgeslagen om later terug te kijken zoals het bekijken van je email en het bekijken van je bankzaken wordt een wachtwoord voor iets heel anders gebruikt. Voor dat soort zaken is het heel belangrijk dat je het data wat opgeslagen wordt precies weer terug krijgt zoals het was.

Je moet natuurlijk niet hebben dat je e mailberichten ineens totaal andere berichten zijn. Of dat je ineens toch minder geld hebt dan dat de bank website je toonde.

Een wachtwoord daarentegen hoef je niet te bekijken want die weet je als het goed is want anders kun je niet inloggen op het systeem waar je op in wilt loggen.

Het wachtwoord dient dus alleen voor het kijken of je echt degene bent die je zegt dat je bent. Dit is ook gelijk de reden waarom het wachtwoord zo belangrijk is.

Als je op een website je e mail invult om in te loggen zeg je dus eigenlijk tegen het systeem “Ik ben deze persoon laat mij erin”. Waarna het systeem zegt “bewijs jij maar dat je diegene ook bent”. En dat kun je dan bewijzen via het wachtwoord.

Het probleem van wachtwoorden is dus dat het systeem ook moet kunnen nagaan of het wachtwoord wel klopt.

**Wachtwoord opslag**

In zeer oude systeem werd het wachtwoord zogeheten “flat text” opgeslagen in de database. Dit houd in dat het wachtwoord wordt opgeslagen zoals hij ook ingetyp is tijdens het registreren. Als jouw wachtwoord 123 is wordt hij dan ook opgeslagen als 123.

Dit houd in dat het systeem jouw wachtwoord weet en dus makkelijk kan zien of jij ook degene bent die je zegt dat je bent. Het probleem hieraan is dat iedereen die toegang tot de database heeft gelijk je wachtwoord kan zien waarna hij kan inloggen op jouw account.

Hiervoor moest is uitgevonden worden. Zoals ik aan het begin al verteld had is het opslaan van data en van een wachtwoord verschillend omdat je niet het wachtwoord precies of terug te halen zoals hij is ingetypt. Het enige wat een wachtwoord voor dient is het verifiëren dat je bent wie je zegt dat je bent. Er is dus een mogelijkheid om je wachtwoord te verstoppen en het systeem hoeft niet te weten waar het wachtwoord is verstopt maar alleen hoe hij het verstopt. Dit verstoppen is het basis van hashing. Een makkelijk voorbeeld van hashing is bijvoorbeeld een 1 opslaan als A een 2 als B en een 3 als C enzovoort. Hierdoor als we een account maken op het systeem met als wachtwoord 123 wordt hij dus opgeslagen als ABC. Hierdoor is het niet gelijk duidelijk voor mensen met toegang naar de database dat het wachtwoord 123 is omdat er ABC is de database staat. Terwijl het systeem wel weet dat als jij 123 invult bij het wachtwoord veld dat dat toch het goede wachtwoord is. Al is 123 naar ABC wel heel makkelijk om ook weer terug te rekenen zijn de basis encryptie methoden zoals md4, md5, sha1, sha256 dat niet.

**Het probleem**

In de basis encryptie methoden worden wachtwoorden altijd op dezelfde manier encrypt zodat als je je wachtwoord intypt dat hij het opnieuw kan encrypten en kan vergelijken met het wachtwoord in de database. Dit houd in dat 123 altijd ABC is op elk systeem en dat als je weet dat ABC 123 is dat je nog steeds kan inloggen op het systeem.

Omdat de wachtwoorden hetzelfde zijn op elk systeem is het mogelijk om een gigantische database met alle wachtwoorden op te slaan waarna je alle wachtwoorden kan opzoeken in de database want ABC is altijd 123 en DEF is altijd 456. Deze databases noemen we rainbow tables.

**De oplossing**

Nu de oplossing hiervoor. Oke, wachtwoorden zijn er dus voor om te kijken of jij bent wie je zegt dat je bent op het systeem. Maar dan zonder dat het mogelijk is om het op te slaan in een andere database en terug te zoeken. Hiervoor zijn enkele trucjes bedacht.

Eentje die vaak gebruikt wordt is het combineren met het wachtwoord met een andere gegeven die vast staat. Dit wordt een “Salt” genoemd in de encryptie wereld. Een salt is meestal gewoon een rij characters. Omdat deze salt erbij toegevoegd wordt komen de hashes in de database nooit overeen met de hashes in andere databases. Hierdoor is 123 niet altijd ABC maar wellicht ook soms QWE of iets totaal anders willekeurig.

Wat op het heden vaak gedaan wordt is dat er een unieke salt wordt opgeslagen voor elke gebruiker van het systeem. Soms wordt dit erbij opgeslagen dat je niet alleen in de database het wachtwoord ziet staan maar ook de salt. Vaker wordt er gekozen om een andere gebruiker gegeven te gebruiken die al vast staat tijdens het registreren zoals de tijd van registratie of een niet aan te passen email adres.

Er zit een groot verschil tussen een unieke salt per gebruiker en een vaste salt. Als het een vast gegeven is hoeft er maar een grote database te komen die specifiek jouw salt bevat terwijl er als je een unieke salt hebt een hele database per salt moet komen alleen om een account te kunnen hacken. Dit maakt het dus niet mogelijk om in een redelijke tijd een hele database aan gebruikersaccounts kwijt te raken aan hackers.

**Samenvatting**

Om het samen te vatten geen een gebruiker weet wat er met zijn wachtwoord gebeurd. En geen een bedrijf gaat zeggen hoe hun ermee omgaan. De goede bedrijfen niet omdat ze hun encryptie geheim willen houden en de slechte bedrijfen neit omdat ze niet kunnen omgaan met wachtwoorden. We moeten maar vertrouwen dat er goed gebuik wordt gemaakt van encryptie.

Het lieft encryptie die een unieke salt gebuikt voor jouw en alleen jouw account. Want dat is om dit moment het vijligste wat we hebben. Bedankt voor uw aandacht.