Auteur: Maarten Hartsuijker is redactielid van iB-Magazine en helpt als ethisch hacker organisaties met het pentesten en scannen van applicaties en netwerken. Maarten is te bereiken via: m.hartsuijker@classity.nl.

Kraak de kluis

Ocean's Eleven, King of Thieves, The Italian Job… In Hollywood grossieren ze in films waarin good guys of bad guys een meesterplan bedenken voor het kraken van de kluis. Tegenwoordig niet zelden samen met een meesterhacker die de laatste technologische beveiligingen moet zien te kraken. Zelf maken we op internet ook steeds vaker gebruik van een kluis: de wachtwoordkluis. En soms blijkt het kraken van die kluis kinderlijk eenvoudig.

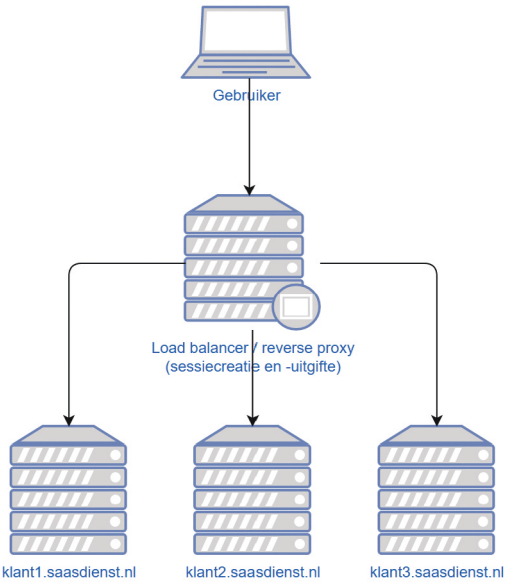
Een goede wachtwoordkluis heeft natuurlijk end-to-end-beveiliging die borgt dat wachtwoorden vóórdat ze naar de opslaglocatie worden verzonden, versleuteld worden met een encryptiesleutel die is afgeleid van een door de eigenaar verzonden hoofdwachtwoord. Deze aanpak borgt dat iemand die de kluis kraakt alleen versleutelde wachtwoorden kan stelen. En niet de wachtwoorden zelf. Dit klinkt logisch (en veel wachtwoordkluizen hanteren deze aanpak), maar sommige kluizen hebben deze extra beveiliging niet. Toegang tot een gebruikersaccount betekent in dat geval toegang tot de wachtwoorden in de kluis.

Security by design: persoonlijk encryptiesleutel

Inmiddels alweer een paar jaar geleden mocht ik voor een klant een Identity en Access Management-oplossing onderzoeken. De tool zou de gebruikers van de organisatie gaan helpen om via één plek alle applicaties toegankelijk te maken. Gebruikers hoefden per applicatie slechts één keer hun wachtwoord in te geven, dat vervolgens de kluis in ging om automatisch te worden ingevuld zodra er op een applicatie werd ingelogd. Het principe werkte prachtig en was al bij vele Nederlandse organisaties succesvol in gebruik. De eerste security by design-bevinding werd direct zichtbaar: gebruikers hoefden alleen op een webportaal in te loggen om hun wachtwoorden te gebruiken. Er werd geen hoofdwachtwoord gevraagd. Hierdoor kon elke beheerder na een wachtwoord reset van het kluisaccount een medewerkersidentiteit overnemen. De mogelijkheid impliceert ook dat de wachtwoorden in de kluis niet volledig end-to-end beschermd zijn. Een inbraak in deze cloud zou daardoor eenvoudig tot een grote supply chain aanval kunnen leiden. Waaronder op de organisatie van mijn klant, die daarom besloot de applicatie onafhankelijk te laten toetsen zodat hij de beveiligingsrobuustheid beter kon inschatten. Zoals van dit product verwacht mag worden was de basisbeveiliging op orde. Firewalls, reverse proxies, beveiligingsupdates, wachtwoordpolicy, datavalidatie, MFA-mogelijkheden… De usual suspects leverden niet direct iets op. Maar in het sessiemanagement was iets vreemds aan de hand.

Opzet van de dienst

Klanten van de dienst werden voorzien van hun eigen virtuele bedrijfskluis: klant.saasdienst.nl. Elke kluis had zijn eigen accountbeheer, kluisopslag, et cetera. Voor de klantomgevingen stond een centrale toegangsvoorziening (load balancer). Iedereen die succesvol inlogde kreeg van de toegangsvoorziening een sessiecookie. Op het klantsysteem kon aan de hand van het sessiecookie worden gecontroleerd wie de gebruiker was. Tot zover leek er weinig mis. De sessiecookies waren door hun willekeurigheid niet te voorspellen. En zonder sessiecookie kon de toegangsvoorziening niet worden gepasseerd en was er geen toegang tot de klantkluis mogelijk. En toch bleek deze sessiecookie de sleutel tot het overnemen van de hele SAAS-dienst en alle kluizen.



Figuur 1 - Diagram kraak de kluis.

Geen tenant controle

De architectuurkeuze om het sessiemanagement te ontkoppelen van de klantomgevingen zorgde ervoor dat sessiecookies binnen alle klantomgevingen bruikbaar waren. Want hoewel de klantapplicaties aan de hand van het cookie controleerde welke gebruiker er aan het cookie was gekoppeld, werd er niet vastgesteld voor welke klantomgeving het cookie was uitgegeven. Een beheerder hoefde daardoor enkel een account aan te maken met een gebruikersnaam gelijk aan een gebruiker van een andere omgeving om (in combinatie met een kleine aanpassing in de applicatieverzoeken) toegang te krijgen tot dat account in die omgeving. Daarbij werd hij overigens geholpen door het feit dat de naam van de key-user in alle klantomgevingen gelijk was. De kluis kon in dit geval dus eigenlijk 'gekraakt' worden doordat de sessiesleutel onbedoeld een loper was geworden. Een loper die het mogelijk maakte om alle kluizen te openen waarin zich de sleutels van vele bedrijfsapplicaties van alle klanten van deze IAM-dienst bevonden. En een loper waarvan de applicatiearchitect vol ongeloof eerst de werking wilde zien voordat hij écht overtuigd was dat hij bestond. Twee uur later was de loper overigens onbruikbaar. Met de oplostijd was niets mis.

Ignorance is bliss

Net als in Hollywood hebben ook in dit geval de eigenaren van de kluis vermoedelijk nooit geweten hoe eenvoudig hun kroonjuwelen in die kluis voor onbevoegden toegankelijk waren. Maar het ontwerp van de kluis had iedereen eigenlijk vrij eenvoudig kunnen valideren. Want een wachtwoordkluis in de cloud zonder een van een persoonlijk masterwachtwoord afgeleide encryptiesleutel, hoor je eigenlijk vandaag de dag niet meer in gebruik te nemen.