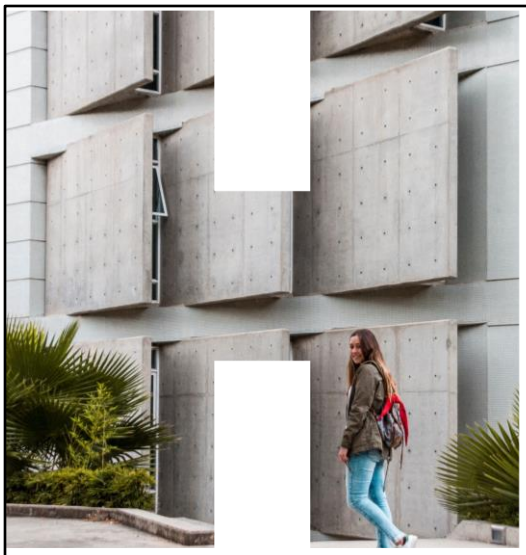

HO
GENT



Cybersecurity

5. Het verzekeren van integriteit

HO GENT

5. Het verzekeren van integriteit

5.1 Soorten integriteitscontroles
5.2 Digitale handtekeningen
5.3 Certificaten

HO
GENT

5.1 Soorten integriteitscontroles

HO GENT

Hashing algoritmes

- Vormt een reeks van bits om naar een reeks van een vast aantal bits (alles in IT kan voorgesteld worden in bits)
- Het is een wiskundige eenrichtingsfunctie. Het is in de ene richting makkelijk te berekenen, maar onmogelijk te berekenen in de andere richting
  - Vergelijkbaar met het vermalen van koffiebonen. Het vermalen van bonen tot gruis is gemakkelijk, maar het terug samenplakken van gruis tot bonen is onmogelijk

HO GENT

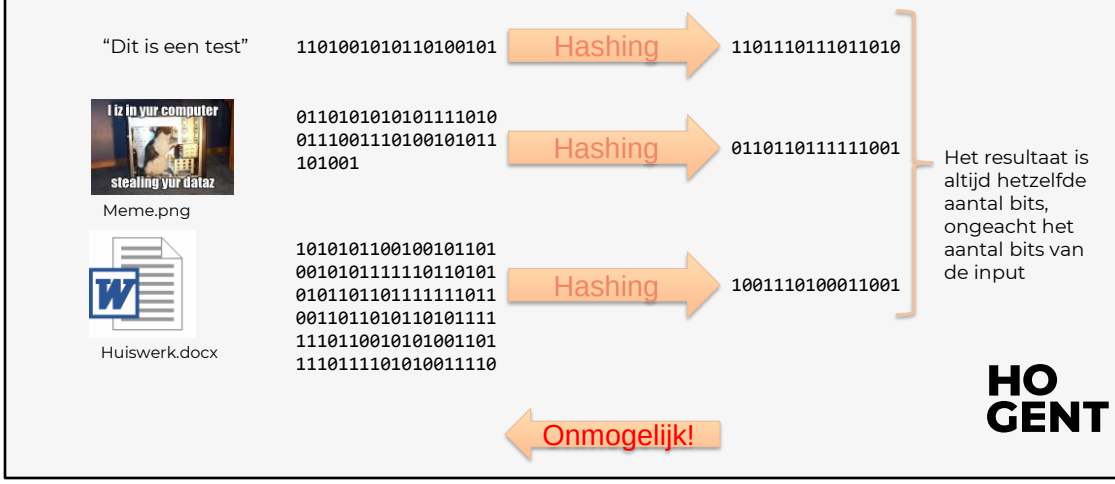
Hashing algoritmes

- Een cryptografische functie heeft de volgende eigenschappen:
  - De input kan uit om het even welk aantal bits bestaan
  - De output heeft steeds hetzelfde aantal bits (ongeacht het aantal bits van de input)
  - De hashfunctie is een eenrichtingsfunctie en is onmogelijk om te keren
  - Twee verschillende inputwaarden zullen steeds een verschillende outputwaarde geven

HO GENT

5.1 Soorten integriteitscontroles

Hashing algoritmes



HO GENT

De hier getoonde binaire waarden zijn verzonnen voor educatief inzicht. Effectieve waarden worden berekend en getoond in de praktijkopdrachten.

Hashing algoritmes

- Er bestaan tegenwoordig een hele reeks hashing algoritmes. De meest populaire zijn MD5 en SHA:
  - Message Digest 5 Algorithm (MD5) ontwikkeld door Ron Rivest. Het geeft een 128-bits output terug
  - Secure Hash Algorithm ontwikkeld door het US National Institute of Standards and Technology (NIST). Hiervan bestaan verschillende varianten afhankelijk van het gewenst aantal bits van de output:
    - SHA-224 (224 bits)
    - SHA-256 (256 bits)
    - SHA-384 (384 bits)
    - SHA-512 (512 bits)
  - Probeer het zelf op http://www.fileformat.info/tool/hash.htm

HO GENT

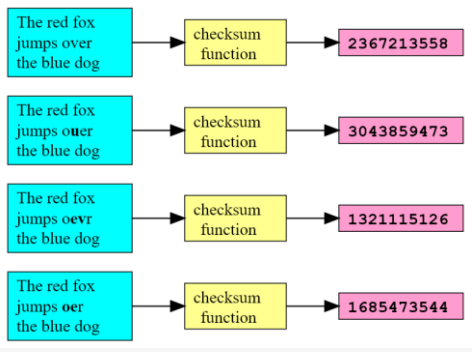
Toepassingen

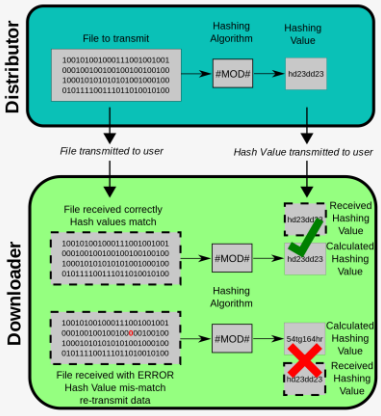
- Controleren op fouten in data
- Het veilig bewaren en controleren van paswoorden in databanken
- Identificeren van data aan de hand van een kleinere waarde (fingerprint)
- Gebruikt bij efficiënte opslag van data in hashtabellen

HO GENT

5.1 Soorten integriteitscontroles

Controleren op fouten





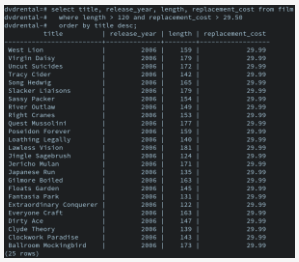
HO GENT

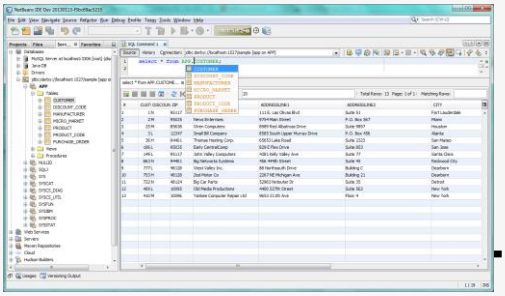
Oorsprong afbeeldingen:

* https://en.wikipedia.org/wiki/Checksum#/media/File:Checksum.svg
* https://en.wikipedia.org/wiki/MD5#/media/File:CPT-Hashing-File-Transmission.svg

Veilig bewaren van paswoorden

- Gebruikersnaam en paswoord worden vaak opgeslagen in databanken
- Databanken zijn een efficiënte manier om data op te slaan, analyseren en op te vragen





Veilig bewaren van paswoorden

- Door de groei van data collectie en de vaak vertrouwelijke aard van data, is het belangrijk voor cybersecurity personeel om het steeds groeiend aantal databanken te beschermen
- Databanken zijn een gewild doelwit van cybercriminelen. Een gelekte hoeveelheid gegevens uit een databank noemt een Data Breach (datalek)
- Velen hergebruiken dezelfde username (vaak hun e-mailadres) en paswoord. Als deze combinatie wordt gevonden in een data breach, wordt deze door cybercriminelen getest op andere websites. Dit is een grote oorzaak van hacks
- Het gebruik van 1 paswoord dat je telkens lichtjes aanpast per website heeft geen zin. De software van cybercriminelen houdt hier rekening mee
- Gebruik een paswoordmanager en gebruik een uniek paswoord per website!
- Test jezelf op https://haveibeenpwned.com/

HO GENT

Veilig bewaren van paswoorden

Adult Friend Finder

Date: October 2016
Impact: 412.2 million accounts
Details: This breach was particularly sensitive for account holders because of the services the site offered. The FriendFinder Network, which included casual hookup and adult content websites like Adult Friend Finder, Penthouse.com, Cams.com, iCams.com and Stripshow.com, was breached in mid-October 2016. The stolen data spanned 20 years on six databases and included names, email addresses and passwords.

The weak SHA-1 hashing algorithm protected most of those passwords. An estimated 99% of them had been cracked by the time LeakedSource.com published its analysis of the data set on November 14, 2016.

More than 540 million records about Facebook users were publicly exposed on Amazon's cloud computing service, according to a cybersecurity research firm. A report out Wednesday by UpGuard said two third-party Facebook app developers posted the records in plain sight, causing yet another major data breach for the world's biggest social network.

According to UpGuard, a Mexico-based media company called Cultura Colectiva was responsible for the biggest leak. It exposed 146 gigabytes of Facebook user data, including account names, IDs and details about comments and reactions to posts. It's unclear how many individual users had data exposed.

Separately, an app called At the Pool exposed databases that appeared to include data about user IDs, friends, photos and location check ins, as well as unprotected Facebook passwords for 22,000 users. The app – which was meant to help people meet up for offline activities – shut down in 2014.

HO GENT

https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html
https://www.cbsnews.com/news/millions-facebook-user-records-exposed-amazon-cloud-server/

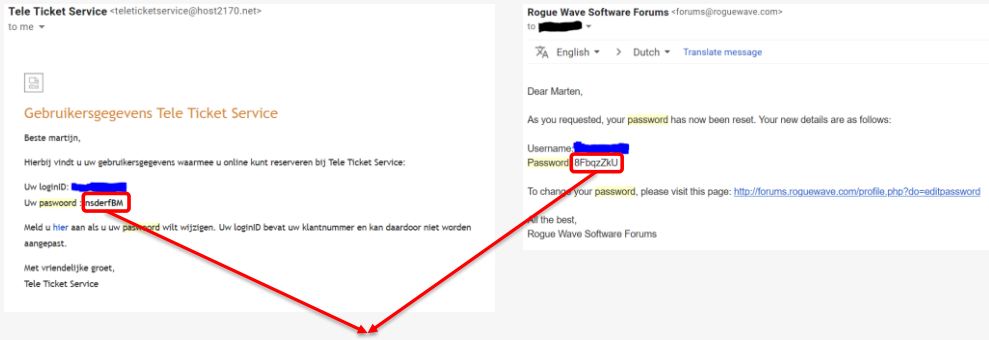
Veilig bewaren van paswoorden

- **Poging 1**: we slaan de gebruikersnaam en paswoord gewoon op in de databank. Bij het inloggen moeten we gewoon het volgende controleren:
  - Naam == naam in databank?
  - Paswoord == paswoord in databank?
  - Wordt ook “plaintext” opslag genoemd
  - Eenvoudig
  - Als iemand in de databank geraakt, kan hij meteen alle paswoorden uitlezen → GEVAARLIJK!

Gebruikersnaam	Paswoord
tom@gmail.com	Dit is een St3rk Paswoord!
hanne@outlook.com	zwakpaswoord
ruth@gmail.com	{4w^nv9y]]V{}-53hvxSQ.-f_ZeerSterk

HO GENT

Veilig bewaren van paswoorden



Plaintext == GEVAARLIJK!

HO GENT

Veilig bewaren van paswoorden

- **Poging 2**: we slaan de gebruikersnaam en de hashwaarde van het paswoord gewoon op in de databank. Bij het inloggen moeten we gewoon het volgende controleren:
  - Naam == naam in databank?
  - Hash(paswoord) == hashwaarde in databank?
  - De organisatie heeft de paswoorden zelf dus NIET in zijn databank staan en kan ze dus zelf NIET zien
  - Als iemand in de databank geraakt, kan hij de paswoorden NIET uitlezen → VEILIG!

	Gebruikersnaam	Paswoord
2	tom@gmail.com	c341c6a9c6246f9a0063487ecf68f8981f6e2bfc
3	hanne@outlook.com	9ffab04eed6079cf9a576d2a6d2e9a49320db79c
4	ruth@gmail.com	927a06fd5707fb4bbbaa1ad262cc2ea490dc5e1b

HO GENT

Salting

- Salting is een extra maatregel om hashing veiliger te maken
- Wanneer 2 gebruikers hetzelfde paswoord gebruiken, zal voor beiden dezelfde hashwaarde worden opgeslagen. Hierdoor weten aanvallers dat ze door 1 paswoord te kraken 2 vliegen in 1 klap slaan
- Een salt is een random reeks bits dat wordt toegevoegd aan het paswoord. Zo zullen beide gebruikers een verschillende hash krijgen, ook al gebruiken ze hetzelfde paswoord

	Salt		Hash Value
Hash ("password" +	QxLUF1bIAdeQX)	=	b3bad1e5324f057753a4b8d7cef293e4
Hash ("password" +	R9PeIC7sxQXb8)	=	713c7beb54841a26a7c81eb06d6cf066

HO GENT

Salting

	Gebruikersnaam	Paswoord
2	elise@gmail.com	13!Kabouters0pP4d
3	hanne@outlook.com	13!Kabouters0pP4d

Plaintext → GEVAARLIJK!

	Gebruikersnaam	Paswoord
2	elise@gmail.com	934db60e4fc67993bd89cda5ec0793bfbeba6938
3	hanne@outlook.com	934db60e4fc67993bd89cda5ec0793bfbeba6938

Hash waardes → matig VEILIG!

Maar gelijke paswoorden geven gelijke hashwaarden. Cybercriminelen kunnen dit misbruiken

	Gebruikersnaam	Paswoord	Salt
2	elise@gmail.com	cca528d733db517b74273c998a1e6836e262f421	9665d054e5c
3	hanne@outlook.com	468418c2415084f1c48be5e950606cb9ba2f9bc5	fea2aa6744b

Hash waardes + salt waardes → extra VEILIG!

Bv. hash(13!Kabouters0pP4d9665d054e5c) == cca528d733db517b74273c998a1e6836e262f421
hash(13!Kabouters0pP4dfea2aa6744b) == 468418c2415084f1c48be5e950606cb9ba2f9bc5

HO GENT

Rainbow tables

- Vanuit een hashwaarde de oorspronkelijke input berekenen is zo goed als onmogelijk
- Een manier is voor alle mogelijke inputwaarden de hashfunctie uit te rekenen (bruteforcing) en de resultaten opslaan in een lijst. Een dergelijke lijst noemt een Rainbow table
- Gebruik dus een aparte salt bij elk opgeslagen paswoord, dit maakt de Rainbowtable nutteloos

Password	MD5 Hash
123456	e10adc3949ba59abbe56e057f20f883e
password	5f4dcc3b5aa765d61d8327deb882cf99
12345	827ccb0eea8a706c4c34a16891f84e7b
12345678	25d55ad283aa400af464c76d713c07ad
qwerty	d8578edf8458ce06fbc5bb76a58c5ca4
123456789	25f9e794323b453885f5181f1b624d0b
1234	81dc9bdb52d04dc20036dbd8313ed055
baseball	276f8db0b86edaa7fc805516c852c889
dragon	8621ffdbc5698829397d97767ac13db3
football	37b4e2d82900d5e94b8da524fbeb33c0

HO GENT

Collisions

- Hashing algoritmes moeten telkens een unieke output hebben voor verschillende inputs
- Wanneer je voor 2 verschillende input waarden dezelfde output waarde krijgt, spreekt men van een “collision”
- Een hashing algoritme verliest zijn nut als collisions bewust veroorzaakt kunnen worden

```
Hash(qwerty)     == 548d4efa8
Hash(ietsAnders) == 548d4efa8
                    COLLISION!
```

HO GENT

https://csrc.nist.gov/publications/detail/sp/800-107/rev-1/final

Collisions

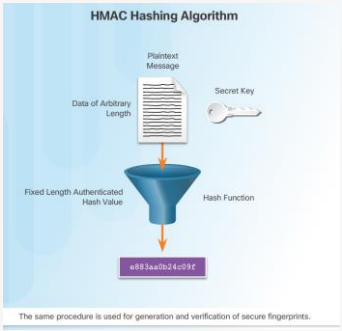
- Hashing algoritmes worden zo onderverdeeld in zwakke en sterke hashing algoritmes
- MD5 en SHA-1 zijn niet meer bruikbaar voor cybersecurity doeleinden, gebruik deze dus hiervoor NOOIT!
- SHA-2 en SHA-3 kunnen wel nog gebruikt worden voor cybersecurity doeleinden
- https://csrc.nist.gov/projects/hash-functions
- Praktisch voorbeeld op https://www.mscs.dal.ca/~selinger/md5collision

HO GENT

https://csrc.nist.gov/publications/detail/sp/800-107/rev-1/final
https://www.mscs.dal.ca/~selinger/md5collision/
https://speakerdeck.com/ange/kill-md5
https://sha-mbles.github.io/

5.1 Soorten integriteitscontroles

HMAC



- HMAC versterkt het hashing algoritme door het toevoegen van een extra sleutel
- Hierdoor wordt controleren van integriteit aangevuld met authenticatie
- De checksum is het resultaat van het HMAC algoritme en wordt meegestuurd met het bericht
- De sterkte van HMAC hangt af van het gebruikte hashing algoritme (verschillende hashing algoritmes zijn mogelijk)

HO GENT

Wie graag meer wil weten, kan eens snuisteren in de volgende RFC's (Request for Comments):

* https://tools.ietf.org/html/rfc2104
* https://tools.ietf.org/html/rfc6234

Vertragende hashing algoritmes

- Om te vermijden dat er heel veel pogingen worden gedaan om een hash te kraken of een collision te vinden wordt er vaak gebruik gemaakt van vertragende hashing algoritmes
- Een aanvaller met meer rekenkracht moet zo toch nog steeds lang wachten om de hashwaarde te kraken
- Bekende voorbeelden:
  - PBKDF2
  - bcrypt
  - Argon2

HO GENT

5.2 Digitale handtekeningen

HO GENT

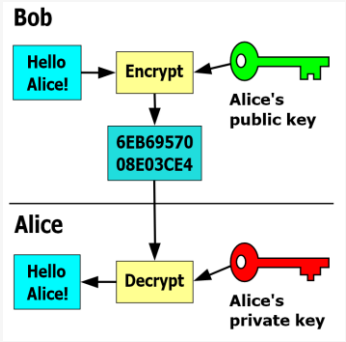
Handtekeningen en de wet

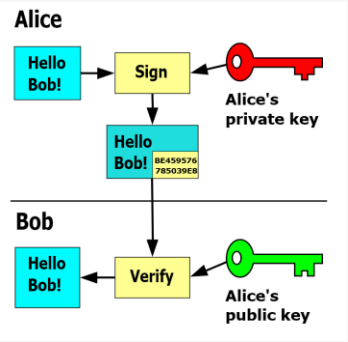
- Digitale handtekeningen vullen de zelfde rol als gewone handtekeningen, maar dan voor elektronische bestanden
- Met een digitale handtekening kunnen 2 zaken worden gecontroleerd:
  - Het bestand is niet veranderd nadat de handtekening is gegenereerd
  - Het bestand is daadwerkelijk afkomstig van de persoon die de handtekening heeft gegenereerd en niet van iemand anders
- In veel landen heeft een digitale handtekening evenveel juridisch gewicht als een traditionele analoge handtekening

HO GENT

Hoe werkt dit?

- Asymmetrische cryptografie genereert een publieke en private sleutel per persoon. Deze sleutels zijn wiskundig aan elkaar gelinkt





HO GENT

In latere vakken zie je in meer detail hoe dit werkt.

5.2 Digitale handtekeningen

Hoe werkt dit?

- Test het op https://sela.io/pgp-en/
- Normaal gebruik je hiervoor software op jouw eigen computer (GPG). Genereer nooit jouw sleutels via een website voor echt gebruik!
- Voorbeeld publieke en private sleutel:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: OpenPGP v2.0.8

793hZyatESpYcgQplSrDXBBc9MZGRE9fvZE8Qvsc+r9/kpf7BLJ/
YU5xKVFkFuaSTx2k6Mwd2F6Mo9OoNcAdeS9vALjvtyD8T/EGoPRq
SQ1rbQfIi814JW7unXgtyfu6/si2fsYa/4Qv0ds1ZyPHvVxh2e79
5T0yRTGz9GFaisapG+XoaYVrbfuauVtITKgQHXb8Pl6ekE6SQ+gS
c3kK+Mah/HbN/OmQxLESVg4d2QpUSP/klq+NtHq08cNQgW4HZoOw
SDkTEVEpfDDCIvuYcHF5gnITpFGku58BUQLSdi8rVSe/VNbaHupR
KTXT9XrMRAjfAyWrZ85RLE7UvFKkZHGTxnpZIv/9JShfd6+OlBZ4
W3b8XGBDcqskWbnXi6Ia3VIuYQNhrO8HWeh/TgTpWWqibG4DumW8
S/aGKkaXI+iR/tP7B66DB8q2zEJoXw==
=bGdt
-----END PGP PUBLIC KEY BLOCK-----
```

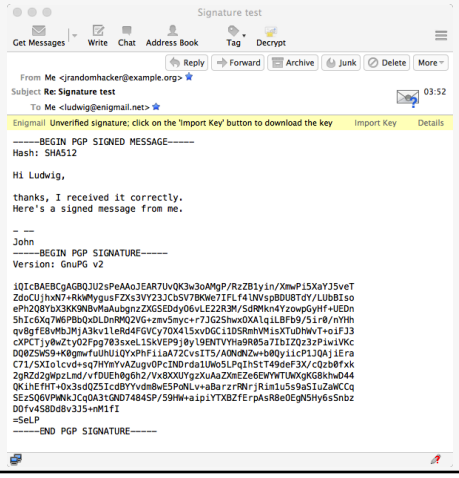
```
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: OpenPGP v2.0.8

xcaGBF9fh/EBEADk+w2xYivAYznxkZM3Rv0U0gyqj9gDbVmPloQ9f
3F/0B9Tmtwht3Z8JmwPJ++1GUUnT03mMlGYKtblVYPX1uAQItIaBN
Cyk/wZ4+T16g3rSaN6H8tzu4BLbqu7LmSFNZGKqE18PiePdTe5y/D
793hZyatESpYcgQplSrDXBBc9MZGRE9fvZE8Qvsc+r9/kpf7BLJ/B
YU5xKVFkFuaSTx2k6Mwd2F6Mo9OoNcAdeS9vALjvtyD8T/EGoPRq6
zEQI3wMlq2fOUSxO1LxSpGRxk8Z6WSL//SUoX3evjpQWeJZmbkooI
Q3KrJFm514uiGt1SLmEDYazvB1nof04E6VlqomxuA7plvP8Kbznp3
lyPokf7T+weugwfKtsxCaF8=
=UmmP
-----END PGP PRIVATE KEY BLOCK-----
```

HO GENT

De sleutels hier zijn niet geldig, ze dienen gewoon ter illustratie zodat je je kan voorstellen hoe een sleutel er uit ziet. De inhoud van zo een sleutel wordt opgeslagen in een tekst- of binair bestand.
In latere vakken zie je in meer detail hoe dit werkt.

Hoe werkt dit?

- Voorbeeld handtekening:



HO GENT

De handtekening hier is niet geldig, ze dient gewoon ter illustratie zodat je je kan voorstellen hoe een handtekening er uit ziet.
In latere vakken zie je in meer detail hoe dit werkt.

5.3 Certificaten

HO
GENT

De basis

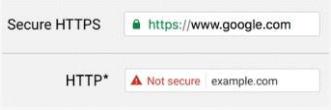
- Een digitaal certificaat is gelijkaardig aan een elektronisch paspoort
- Het laat toe gebruikers, toestellen en organisaties toe om veilig informatie te delen over het internet
- Een certificaat authenticeert en verifieert dat de verzender van bericht daadwerkelijk die persoon is
  - Het koppelt een persoon aan een publieke sleutel
- Certificaten worden uitgedeeld door betrouwbare organisaties
- Certificaten bieden ook vertrouwelijkheid aan door encryptie

HO GENT

5.3 Certificaten

Gebruik bij websites

- HTTPS versleutelt data tussen webserver en client
  - Je kan zien of encryptie aan staat aan het "slot" icoontje in de webbrowser



- HTTPS bewijst niet dat de website geen frauduleuze website is (bv. phishing)!
  - Root CA's zijn geen politie
  - Initiatieven als Let's Encrypt laten iedereen toe om SSL certificaten te genereren voor domeinnamen (g00gle.com, microsofl.com, ...)
  - "slot" icoontje in de webbrowser toont enkel aan of er encryptie is of niet

HO GENT

Je kan zelf het certificaat van de website die je bezoekt bekijken in jouw webbrowser. Een tutorial voor firefox vind je op https://support.mozilla.org/en-US/kb/secure-website-certificate .
Meer informatie over Phishing en HTTPS:
* https://www.wired.com/story/phishing-schemes-use-encrypted-sites-to-seem-legit/
* https://info.phishlabs.com/blog/more-than-half-of-phishing-sites-use-https

5.3 Certificaten

Gebruik bij websites

- Elke browser heeft een lijst van vertrouwde organisaties (Root Certificate Authorities)
- Organisaties bewijzen aan een Root CA dat ze het internetdomein bezitten (bv. example.com)

HO GENT

Een versimpelde uitleg over HTTPS en certificaten vind je op https://robertheaton.com/2014/03/27/how-does-https-actually-work/ .
Je kan de lijst van Root CA's voor Firefox vinden op https://ccadb-public.secure.force.com/mozilla/CACertificatesInFirefoxReport . Chrome gebruikt de lijst van Root CA's van het besturingssysteem; voor Windows kan je deze vinden op https://ccadb-public.secure.force.com/microsoft/IncludedCACertificateReportForMSFT .

Opstellen van een certificaat

- Een digitaal certificaat volgt een gestandardiseerd formaat zodat software dit kan lezen en verstaan, ongeacht de verzender
- De X.509 standaard is op dit moment de standaard voor het opstellen en beheren van digitale certificaten
- Een Public Key Infrastructure (PKI) is de verzameling van beleid, functies en procedures om certificaten te bewaren, beheren, verspreiden, gebruiken en ongeldig te verklaren

HO GENT

Voorbeeld van een certificaat

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            72:14:11:d3:d7:e0:fd:02:aa:b0:4e:90:09:d4:db:31
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=Texas, L=Houston, O=SSL Corp, CN=SSL.com EV SSL
Intermediate CA RSA R3
        Validity
            Not Before: Apr 18 22:15:06 2019 GMT
            Not After : Apr 17 22:15:06 2021 GMT
        Subject: C=US, ST=Texas, L=Houston, O=SSL
Corp/serialNumber=NV20081614243,
CN=www.ssl.com/postalCode=77098/businessCategory=Private
Organization/street=3100 Richmond Ave/jurisdictionST=Nevada/jurisdictionC=US

...
```

HO GENT

Het certificaat hier is niet geldig, ze dient gewoon ter illustratie zodat je je kan voorstellen hoe een certificaat er uit ziet.
In latere vakken zie je in meer detail hoe dit werkt.


HO
GENT