Labo chapter 2

Bavo ’s Heeren

Yoeri Odekerken

Ruben Jamart

Brent Put

Wout Knoors

INDEX

**[1 Terminologie en vakjargon 2](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit" \l "heading=h.rhpshyi98s1s)**

**[2 Identificeer het algoritme 3](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit" \l "heading=h.4px759cjow1t)**

**[3 Encrypties & conversies 4](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit" \l "heading=h.gcksf1vfbwid)**

[**4 From DES to AES 5**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.lo8ze6pmqcbh)

[**5 RSA - DH 6**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.y0t50rdiou)

[**6 Steganography 7**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.nam9z1buiapp)

[**7 THM → steganography 8**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.f9du4mqs2cw4)

[**8 Kraken van de encryptie 9**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.4jf7gkb6jaok)

**[9 THM → crypto 101 10](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit" \l "heading=h.aw08x8hjqdzq)**

**[10 Crypto challenges (extra) 11](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit" \l "heading=h.32yv8ie99vdg)**

[**11 VPN 12**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.49cqcmdodmqs)

[**12 Hashes! 13**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.ii23ka3mqhhm)

[**13 Collisions and hashes 14**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.cehtthcjxdks)

[**14 THM → ‘Crack the hash” 15**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.7hdpjkte071w)

[**15 Hashing challenge 16**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.3ibuzrdohe6d)

[15.1 Easy to crack  
15.2 Rainbow table  
15.3 Herken de hash  
15.4 Hashcat  
15.5 Advanced cracking 16](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.ged25rblqw35)

[**16 Beveiligde files 17**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.iapm5jf50o03)

[**17 SSL Certificates 18**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.z86x3ziyj5xb)

[**18 SSL and github 19**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.dwch1crd5n0i)

[**19 SSL and SSH 20**](https://docs.google.com/document/d/1brOsmISHYsAJJFvL8qE_ZjmAnFMbak-38qju1BnxmcE/edit#heading=h.pr2w6qgj7f1t)

 TIJDSBESTEDING

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Bavo ‘s Heeren | Wout Knoors | Yoeri             Odekerken | Ruben Jamart | Bren Put | TOTAAL |
| Deelopdracht 1 | 30min | 30min | 30min | 30min | 30min | 2h30 |
| Deelopdracht 2 |  |  | 2h30 | 2h |  | 4h30 |
| Deelopdracht 3 | 10h |  | 6h |  |  | 16h |
| Deelopdracht 4 | 5h | 3h |  |  |  | 8h |
| Deelopdracht 5 |  |  | 5h | 3h |  | 8h |
| Deelopdracht 6 |  |  | 5h | 5h |  | 10h |
| Deelopdracht 7 | 1h30 | 1h | 1h30 | 1h | 1h | 6h |
| Deelopdracht 8 |  | 3h |  | 5h |  | 8h |
| Deelopdracht 9 | 1h | 1h | 1h | 1h | 1h | 5h |
| Deelopdracht 10 | 4h | 4h | 4h | 4h | 4h | 20h |
| Deelopdracht 11 | 5h30 |  | 5h30 |  |  | 11h |
| Deelopdracht 12 | 3h |  |  |  | 4h | 7h |
| Deelopdracht 13 | 2h30 | 2h30 |  |  |  | 5h |
| Deelopdracht 14 | 3h | 3h | 3h | 3h | 3h | 15h |
| Deelopdracht 15 | 6h | 3h | 4h |  |  | 13h |
| Deelopdracht 16 |  | 4h |  |  |  | 4h |
| Deelopdracht 17 |  |  |  | 3h |  | 3h |
| Deelopdracht 18 |  |  | 5h | 5h |  | 10h |
| Deelopdracht 19 | 1h |  | 8h | 2h |  | 11h |
| Gezamenlijke taken \* | 15min | 15min | 15min | 15min |  | 1h |
| **Totaal   Totaal** | 44h45 | 26h15 | 53h15 | 36 | 16h | 176h15 |

# 1 Terminologie en vakjargon

Opdracht gemaakt door: Yoeri Odekerken, Ruben Jamart, Bavo ‘s Heeren, Bren Put

|  |
| --- |
| Checklist   * ~~Ingevulde woordenlijst~~ |

# Lijst:

* **Encryption** : een methode dat een message of een file encodet zodat het niet zomaar door iedereen gelezen kan worden.
* **Decryption** : het omzetten van encrypted data in zijn originele vorm.
* **Cryptography**: het ontwerpen en gebruiken van codes.
* **Algorithm**: een procedure of formule voor een probleem op te lossen gebaseerd op het uitvoeren van een sequence van specifieke acties.
* **Key**: een value gebruikt bij een computer samen met een complex algorithm voorn het encrypten en decrypten van messages.
* **Steganography**: Verstopt data in een ander soort file. bv. als een afbeelding, een audiobestand of een andere tekst.
* **Security through obscurity**: beveiligingsmaatregelen geheim houden zodat onbevoegde mensen niet weten hoe ze het moeten ontgrendelen.
* **Kerckhoff’s principle**: het cryptosysteem moet goed beveiligd zijn, zelfs als alles publiekelijk bekend is buiten de key.
* **PK**I: systeem waarbij  uitgiften en beheer van digitale certificaten worden gerealiseerd.
* **Brute force**: gebruik van rekenkracht om probleem op te lossen met behulp van computers zonder een algoritme. dit is om berekeningen te versnellen.
* **end-to-end encryption**: versleutelen van bestanden/berichten zodanig dat enkel de ontvanger en verzender deze kunnen lezen
* **the Enigma machine**: een codeermachine. hiermee worden berichten gecodeerd in andere lettercombinaties dan het origineel die dan vervolgens terug vertaald kunnen worden door een identieke machine.
* **Entropy**: Het maakt gebruik van willekeurige dingen die door een systeem wordt verzameld voor gebruik in een algoritme  die willekeurige gegevens vereisen.
* **Confusion**: elke cijfer (bit) van de cijfertekst moet afhankelijk zijn van de verschillende delen van de sleutel waardoor 2 verbindingen worden verdoezeld (verwarring veroorzaken)
* **Diffusion** => 1 bit in de message veranderd , stukken van de encrypted messages zijn veranderd.
* **Session key** =>  Een tijdelijke symmetric encryption key voor een bepaalde tijd of sessie.
* **Ephemeral key** => Een key die voor elke uitvoering van een key exchange wordt genereert .
* **Block cipher** => Een algoritme van een symmetric cryptographic  om plaintext om te zetten naar een blok van ciphertext.
* **Cipher** =>  Een algoritme voor het uitvoeren van encryption of decryption.
* **Ciphertext** => Encrypted tekst omgevormd vanuit plaintext via een encryption algorithm.
* **Keyspace**:  Een verzameling van alle mogelijke verschillende sleutels van een bepaald cryptosysteem.
* **Hash**: Is een verzameling gerelateerde cryptografische hashfuncties
* **Monoalphabetic** ciphers: Is een substitutiecodering waarbij voor een sleutel het coderingsalfabet voor elk gewoon alfabet wordt vastgelegd tijdens coderingsproces.
* **Plain text**: Ongeformatteerde tekst.
* **Polyalphabetic ciphers**: is een cijfer gebaseerd op substitutie, met behulp van meerdere substitutie-alfabetten.
* **Stream cipher**: Een manier om te vercijferen in cryptografie waarbij klare tekst in cijfertekst wordt omgezet per pit of per woord.
* **Symmetric key**: 1 key voor zowel versleuteling als ontsleuteling
* **Asymmetric key**: een key om te versleutelen en een andere om de ontsleutelen
* **Cryptanalysis**: zwakheden zoeken in cryptografische algoritmen om te tekst te ontcijferen zonder de sleutel te kennen
* **Frequency analysis**: analyseren van de verdeling en aantal letters/cijfers die in een tekst voorkomen
* **Obfuscation**: verwarren van broncode om deze er anders uit te laten zien, de functionaliteit wordt behouden
* **RTO**: hersteltijd toestemming. streven om te voldoen aan de hersteltijd na een computercrash.
* **RPO**: Recovery point objective. Het maximale aantal gegevens dat verloren kan worden na een ramp, fout om gelijksoortige event voor het impact heeft op een bedrijf.

# 2 Identificeer het algoritme

Opdracht gemaakt door: Ruben Jamart, Yoeri Odekerken

**Algoritme 1 = Rotation**     
Key = Geen van toepassing  
Principewerking: Vervang de oude key door een nieuwe cryptographic key te genereren. In het voorbeeld worden de letters vervangen in de plaintext door de letter die 13 letters na de originele letter komt in het alfabet. (alfabet kan ook terug naar het begin gaan bijvoorbeeld z wordt m)

**Algoritme 2 = Base64**Key = Niet van toepassing  
Principewerking: Base64-codering kan worden toegepast op een willekeurige reeks bytes. Het algoritme voor de codering verwerkt de data in groepjes van drie bytes. Die bevatten 3 × 8 = 24 willekeurige bits. Die worden gehergroepeerd in vier bytes van 6 bits (4 × 6 is ook 24). Elke byte kan dan dus maar zes bits bevatten, de overige worden nul. Dat maakt 64 combinaties mogelijk. De 64 wordt vol gemaakt met het =-teken en de binaire nul. Daarmee kan ook een mouw worden gepast aan data worden omgezet naar leesbare tekens: hoofdletters, kleine letters en cijfers = 2 × 26 + 10 = 62 tekens, plus de slash en het plusteken.

**Algoritme 3 = OTP**Key = PXL  
Principewerking: Vervang de letters het nummer van de letter van het alfabet beginnent met A=0. Om het bericht te ontcijferen trekken we de sleutel af van de cijfertekst, ook hier weer modulo 10, dus zonder eventuele negatieve overdracht (vb. 25-59=76) en zetten vervolgens de getallen weer om in letters.

**Algoritme 4 = Substitution**Key = Niet van toepassing  
Principewerking: Letters gaan 8 letters naar achter in het alfabet (zelfde werking als bij rotation maar dan met 8)

**Algoritme 5 = Vigenère**Key = SecurityEssentials  
Principewerking: Men kiest eerst een geheime sleutel in de vorm van een geheim sleutelwoord, bijvoorbeeld ZODIAK. Dit schrijft men onder de klare tekst. Vervolgens zoekt men de klare letter in het verticale alfabet op en de letter van het sleutelwoord in het horizontale alfabet. De kruising van beiden is de resulterende codeletter.

**Algoritme 6 = HEX**Key =  Niet van toepassing  
Principewerking:  Talstelsel waarbij met tien cijfers wordt gewerkt, maar met zestien cijfers. De cijfers 0 t/m 9 worden daarom uitgebreid met 'A' (=10) t/m 'F' (=15), ook wel 'a' t/m 'f'.

**Algoritme 7 = XOR**Key = fedcba987654321  
Principewerking: De XOR-operator is zeer gebruikelijk als onderdeel in complexere cijfers. Op zichzelf, met behulp van een constant herhalende sleutel, kan een eenvoudige XOR-codering triviaal worden verbroken met behulp van frequentieanalyse. Als de inhoud van een bericht kan worden geraden of op een andere manier bekend is, kan de sleutel worden onthuld. De belangrijkste verdienste is dat het eenvoudig te implementeren is en dat de XOR-bewerking rekenkundig niet duur is. Een eenvoudig herhalend XOR-cijfer (d.w.z. dezelfde sleutel gebruiken voor xor-bewerkingen op alle gegevens) wordt daarom soms gebruikt om informatie te verbergen in gevallen waarin geen bepaalde beveiliging vereist is. De XOR-codering wordt vaak gebruikt in computermalware om reverse engineering te bemoeilijken.

|  |
| --- |
| Checklist   * ~~1.Type algoritme; Key; principewerking~~ * ~~2.Type algoritme; Key; principewerking~~ * ~~3.Type algoritme; Key; principewerking~~ * ~~4.Type algoritme; Key; principewerking~~ * ~~5.Type algoritme; Key; principewerking~~ * ~~6.Type algoritme; Key; principewerking~~ * ~~7.Type algoritme; Key; principewerking~~ |

# 3 Encrypties & conversies

Opdracht gemaakt door: Bavo ‘s Heeren & Yoeri Odekerken

Beantwoord de vragen voor de verschillende algoritmes.

Houd je aan de nummering zoals in de opgave.

Encrypteer volgend citaat via [CyberChef](https://gchq.github.io/CyberChef/) met de gevraagde algoritmes en beantwoord de vragen.

|  |
| --- |
| “One single vulnerability is all an attacker needs”  -Team 1TINh4- |

Je verandert de “auteur” met jouw team (Team 1TINH4).

Gevraagde algoritmes

1. Base64
2. ROT47
3. Hex
4. ROT13
5. AES \*  
   Key => Team-<Nummer> bijvoorbeeld: Team-01.   
   Verder vul je de key aan met voornamen van je groep als de key te kort is. Kopieer en plak deze dan achter elkaar tot hij lang genoeg is..
6. Bcrypt (# rounds = teamnummer)
7. Enigma (default settings, Plugboard= TE AM)
8. XOR => key = Team-<Nummer>
9. DES
10. ECC
11. MORSE
12. MD5

Base64

* Output : 4oCcT25lIHNpbmdsZSB2dWxuZXJhYmlsaXR5IGlzIGFsbCBhbiBhdHRhY2tlciBuZWVkc+KAnSAKLVRlYW0gMVRJTmg0LQ==
* Omschrijving: Lange output afwisselend met hoofd en kleine letters en cijfers, weinig tekens.
* Kenmerken: 64 tekenset, opvulling door = < .
* Toepassing: Om data op te slaan en te versturen over media wat ontworpen is om om te gaan met ASCII.
* Werking: binaire code converten naar ASCII-tekens (64 delig).

ROT47

* Output : “~?6 D:?8=6 GF=?6C23:=:EJ :D 2== 2? 2EE24<6C ?665D”

\%62> `%x}9c\

* Omschrijving: Niet zo lang. Afwisselend van tekens, cijfers en hoofdletters. Aanhalingsteken blijven gewoon hetzelfde en blijven ook op dezelfde plek staan.
* Kenmerken: Karakterset van 96 karakters, terugkerend patroon.
* Toepassing: Voor het voorkomen van spoilers, puzzle oplossingen en expliciete content.
* Werking: Zelfde werking als ROT13 maar dan met een karakterset van 97.

Hex

* Output : e2 80 9c 4f 6e 65 20 73 69 6e 67 6c 65 20 76 75 6c 6e 65 72 61 62 69 6c 69 74 79 20 69 73 20 61 6c 6c 20 61 6e 20 61 74 74 61 63 6b 65 72 20 6e 65 65 64 73 e2 80 9d 20 0a 2d 54 65 61 6d 20 31 54 49 4e 68 34 2d
* Omschrijving: Iedere groepje bestaat uit 2 karakters ofwel 2 cijfers ofwel een letter en een cijfer
* Kenmerken: 0-9 + A-F. Vaak per groepjes van 1byte (2HEX) of 1 woord(2byte = 4HEX).
* Toepassing: Kleurencode en mac-adressen
* Werking: zestiendelig talstelsels = binair per 4 bit

ROT13

* Output : “Bar fvatyr ihyarenovyvgl vf nyy na nggnpxre arrqf” -Grnz 1GVAu4-
* Omschrijving: Altijd kleine letters. De lengte hangt af van de tekst die je hebt, maar het is meestal een lange geëncrypteerde tekst.
* Kenmerken: Enkel 26 tekens van het alfabet zijn gewisseld.
* Toepassing: Voor het voorkomen van spoilers, puzzle oplossingen en expliciete content.
* Werking: Iedere letter van het alfabet vervangen door de letter die 13 plaatsen verder staan.

AES

* Output : f51e983a76dee6be167f46c948b293461c9d907f9cca8ae3847152a31ebad7163b0a8914f558b85a9c55a4132860f5394025cbef182856bb844b3dbf8e518a2d0302a33e358cd6de2d790b6a18f80246
* Omschrijving:  Een reeks wat bestaat uit letters van A-F en nummers van 0-9.
* Kenmerken: de plaintext wordt geëncrypteerd met de nummer uit de IV.
* Toepassing: Draadloze security, processor security, file encryption en SSL/TLS.
* Werking:3 varianten (128/192/256 bit). Block cipher; blokken van 4x4 (matrix), verschillende bewerkingen met de matrix waaronder een key toevoegen, substitution, permutation over zowel rijen als kolommen.

Brcypt

* Output : $2a$04$ENIXDtuvoHtZrBi6zbQX6eK3eJHtyDfvmST9PVBr9DqV1dYGtn/5K
* Omschrijving: Een zeer korte tekst afwisselend van grote en kleine letters. Het heeft ook cijfers maar zeer weinig tekens. Het bestaat vooral uit hoofdletters.
* Kenmerken: Lijkt op Base64 maar is een stuk korter en tussen $ staat de info over de hash.
* Toepassing: Encrypten en securely opslaan van passwords.
* Werking:  Het resultaat van het 64 keer versleutelen van de tekst “OrpheanBeholderScryDoubt” met behulp van Blowfish.

Enigma

* Output : EQHND TZGHA GREMG ZLEAC YGVPY YAJDR UYUPA KAMXT IMAAD NNPHA
* Omschrijving: Geëncrypteerde tekst met alleen maar hoofdletters.
* Kenmerken: De ciphertext bestaat uit hoofdletters en bestaat alleen maar uit letters uit het alfabet
* Toepassing: Om top-secret  berichten binnen het duitse leger te beveiligen in WW2.
* Werking: Er worden van de 5 codewielen 3 uitgekozen volgens een afgesproken positie. Daarna wordt de geschreven tekst geëncrypteerd naar ciphertext.

XOR

* Output : ..vK.aÊw.j.h.$.q.j.v.f.h.p.$.wÊe.hÊe.$.p.e.o.vÊj.a.w..w$à)¾a.iÊ5¾M¤lÞ)Ê
* Omschrijving: Een tekst met veel tekens/symbolen  erin en letters.
* Kenmerken:veel punten en hoofdletter E met een ^. er staat ook een raar teken in en een paar kleine letters.
* Toepassing: Error checking en fault tolerance
* Werking: we zetten de teken om in ASCII-waarde en daarna in binaire waarde. Deze gaan encrypten met een binaire key. Dit doen we met elke teken.

DES

* Output : HoF9ddIj0QJXVXhgaei049Tefa0mLwC4yRPvQrvw+hzd63jfI+b0ec/sKhxJ4o3d1Hwjeo32TNqBx8lQRbXW6nI00KlqpJee
* Omschrijving:alles wordt verdeeld in datablokken van 64-bits dier in 16- stappen gedefinieerde manieren door elkaar geschoven worden. De helft van de bits wordt in de meeste stappen via een xor-functie gemanipuleerd, waarbij zowel per stap de sleutel wisselt, als ook de groep van bits.

* Kenmerken: bestaat vooral uit grote letters en cijfers, zeer weinig tekens en een paar kleine letters.
* Toepassing: Financiële transacties van de U.S. government wat te maken heeft met electronic fund transfer.
* Werking: Plaintext opsplitsen in blokjes van 64 bit en dan steeds encrypten met een key.

ECC Output :

* Omschrijving:
* Kenmerken:
* Toepassing: Focus op pairs van public en private keys voor het decrypten van webverkeer.
* Werking:

MORSE

Output : **--- -. .**

**... .. -. --. .-.. .**

**...- ..- .-.. -. . .-. .- -... .. .-.. .. - -.--**

**.. ...**

**.- .-.. .-..**

**.- -.**

**.- - - .- -.-. -.- . .-.**

**-. . . -.. ...**

**-....- - . .- --**

**.---- - .. -. .... ....- -....-**

* Omschrijving: Is afwisselend van strepen en stippen om een letter voor te stellen. Het is een lange geëncrypteerde tekst.
* Kenmerken: Tekst bestaat uit streepjes en punten.
* Toepassing: In de scheepvaart wordt dit het meeste gebruikt en in het leger.
* Werking: Er wordt voor een letter uit het alfabet een teken gemaakt met punten en strepen.

MD5

* Output : c1068c1a1c7a057c9741e8662b1154cf
* Omschrijving: Afwisselend van cijfers en kleine letter is altijd maar 32 karakters lang.
* Kenmerken:  32-lange karakterreeks.
* Toepassing: Om files integrity te checken na een transfer.
* Werking: Een reeks van elke lengte te nemen en deze te coderen in een 128-bits fingerprint. Het coderen van dezelfde string met behulp van het MD5-algoritme resulteert altijd in dezelfde 128-bits hash-uitvoer.

* conclusie
  + iedere cipher is anders en verschilt van elkaar.

sommige zijn groter en zien er moeilijker te kraken uit. je ziet ook dat de meeste al goed gekend zijn door de mensen en tegenwoordig makkelijk te kraken zijn dus niet super veilig zijn voor het beveiligen van je bestanden.

|  |
| --- |
| Checklist   * ~~1. BASE64: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~2. ROT47: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~3. Hex: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~4. ROT13: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~5. AES: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~6. Bcrypt: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~7. Enigma: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~8. XOR: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~9. DES: Output + omschrijving + kenmerken + toepassing + werking~~ * 10. ECC: Output + omschrijving + kenmerken + toepassing + werking * 11. MORSE: ~~Output + omschrijving + kenmerken + toepassing + werking~~ * ~~12. MD5: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~13. Enigma: Output + omschrijving + kenmerken + toepassing + werking~~ * ~~CONCLUSIE~~ |

# 4 From DES to AES

Opdracht gemaakt door: Wout Knoors & Bavo ‘s Heeren

Beantwoord de vragen voor de verschillende algoritmes.

Houd je aan de nummering zoals in de opgave.

1. DES
   1. Geef een omschrijving van DES.

**DES is een methode om gegevens met symmetrische cryptografie te versleutelen. DES werkt in de basisvorm met een sleutellengte van 64 bits: 56 bits en 8 controlebits, maar er is gebleken dat DES in de oorspronkelijke vorm niet meer betrouwbaar en veilig was. nu word alles verdeeld in datablokken van 64-bits deze worden opgedeeld in 2 helften van elk 32-bits  die in 16- stappen gedefinieerde manieren door elkaar geschoven worden. De helft van de bits wordt in de meeste stappen via een xor-functie gemanipuleerd, waarbij zowel per stap de sleutel wisselt, als ook de groep van bits.**

1. Zitter er zwakheden in DES?  zoja, welke.

**ja DES maakt gebruik van een kleine sleutelgrootte waardoor dit niet super veilig is.**

**nog een zwakheid is dat bij dezelfde sleutel er bij dezelfde invoer altijd dezelfde uitvoer verschijnt. aangezien er blokken in van 64 bits gewerkt wordt  kan men dus de blokken van 64 bits overschrijven met een andere blok van 64 bits.**

1. Wat is 3DES?  Is 3DES op de dag van vandaag secure?

**3DES gebruikt het DES algoritme 3x om data te versleutelen. het is meer beveiligd wanneer de 3 sleutels niet gelijk zijn. 3DES bestaat uit 168-bits encryptie met 3 sleutels. het coderen kan op 2 manieren met 2 sleutels van 56 bits waarbij de 1e en 3e bewerking met dezelfde sleutel wordt uitgevoerd. of met 3 onafhankelijke sleutels van 56-bits.**

**3DES wordt enkel als secure beschouwd als de 3 sleutels apart worden gebruikt en als ze verschillend zijn.**

1. AES
   1. Geef een omschrijving van AES.

**AES gebruikt 128-bits platte tekst en 128-bits geheime sleutel die samen een 128-bits blok vormen.**

**dit wordt weergegeven als een 4 x 4 vierkante matrix.**

**Deze stap wordt gevolgd door de 10 rondes. Waaronder 9 rondes met de volgende fase**

* **Subbytes: Het gebruikt een S-box waarmee het byte voor bytesubstitutie van het hele blok (matrix) uitvoert.**
* **Rijen verschuiven: Rijen van de matrix zijn verschoven.**
* **Mix kolommen: Kolommen van de matrix worden van rechts naar links geschud.**
* **Voeg ronde sleutels toe: Hier, de Xor van het huidige blok en de uitgebreide sleutel wordt uitgevoerd.**

**En de laatste 10e ronde omvat alleen de fasen Subbytes, Rijen verschuiven en Ronde sleutels toevoegen en biedt 16 bytes (128-bits) cijfertekst**

1. Zitter er zwakheden in AES?  zoja, welke.

**geen gevonden zwakheden.**

1. DES & AES
   1. Formuleer in eigen woorden het verschil tussen DES en AES.

**de blok in DES word verdeeld in 2 helften voordat het verder wordt verwerkt terwijl  bij AES de hele blok wordt  verwerkt.**

**de sleutelgrootte van DES is relatief kleiner dan die van AES hierdoor is DES ook minder veilig dan AES. AES is ook veel sneller dan DES.**

1. Waar wordt DES, 3DES en AES de dag van vandaag gebruikt?

**3DES en AEs wordt vooral gebruikt door grote bedrijven voor het versleutelen van hun bestanden zodat deze veilig zijn.**

**DES wordt vaker gebruikt door de gewone mens omdat deze niet meer zo veilig is en iedereen deze kan kraken.**

|  |
| --- |
| Checklist   * ~~Omschrijving (3)DES~~ * ~~Omschrijvig AES~~ * ~~DES & AES~~ |

# 5 RSA - DH

Opdracht gemaakt door: Yoeri Odekerken & Ruben Jamart

Beantwoord de vragen.  Houd je aan de nummering zoals in de opgave

Componenten van RSA

Bereken een private en public key zoals in het lesvoorbeeld (p & q = 3 & 11) maar met andere getallen.    
Je mag de waarden van p & q zelf bepalen. Je mag hiervoor ook online hulpmiddelen raadplegen (tutorials, andere voorbeelden, calculators,...).   Beschrijf hoe je tot de resultaat bent gekomen.

1. Kan je de private key ‘terugrekenen’ vanuit de public key in jouw voorbeeld?
2. Wat is er verschillend met realistische waardes voor p&q t.o.v jouw voorbeeld?
3. Wat is het verschil van een realistische public en private key t.o.v jouw voorbeeld?
4. Aan welke voorwaarden moeten de parameters van RSA voldoen?
5. Wat kan je hieruit concluderen m.b.t het gebruik van RSA?

RSA praktisch

Wat maakt RSA secure?  Wat wiskundig fact?

Geeft een voorbeeld van public en private key vanuit realistische waarde.

Wat is het effect van quantum computing op prime factorization?

Diffie hellman

Wat is het verschil tussen RSA en Diffie Hellman?

Op welk wiskundig feit is Diffe Hellmann gebaseerd?

Tip: <https://muirlandoracle.co.uk/2020/01/29/rsa-encryption/>

Omschrijving RSA:  Een asymmetrische encryption die gebruikt maakt van het product van 2 priemgetallen met een gelijke lengte tussen de 100 en 200 getallen. Het wordt gebruikt voor het stabiliseren van een veilige verbinding.

* p & q = 5 & 17

n = 85

φ(n) = 64

public key e => 19

private key d => 27

* bewerking d: d x e = 1 mod(φ(n))

          d x 19 = 1 mod(φ(64))

                 19 x d = 1 mod 40

          513d = 27 mod 40

Deelvragen a - e:

a) Nee, je kunt  het niet berekenen. Omdat de private key en public key een wiskundig verband hebben.

b) Realistische waardes voor p&q zijn extreem groot waardoor de bewerkingen ook moeilijker worden.

c)  Realistische keys zijn numerieke keys die gegenereerd zijn , die worden  geconvert in string keys.

d) P & q moeten priemgetallen zijn.

e) Dat het in de realiteit veel meer werk achter zit om de public & private key te gebruiken dan in voorbeelden.

RSA Praktisch: Alleen de andere bevoegde persoon heeft de decryption key(exponent d, modulus n) om te decrypten. Om de private key te maken wordt de formule( d x e = 1 mod(φ(n))) gebruikt. Quantum computers zijn is staat om subroutines van een algoritme te runnen voor factoring. Dit gaat veel sneller dan ieder ander computer.

Diffie hellman : Het verschil met RSA is dat DH een key exchange protocol is, waarbij ze 1 gegeven (p) allebei weten, en bij RSA is er één gezamenlijke private en public key, waarvan de bevoegde personen deze alle 2 weten. Diffie Hellman is gebaseerd op een wiskundig feit genaamd g^ab(mod p).

|  |
| --- |
| Checklist   * ~~Omschrijving RSA + 5x deelvragen a - e~~ * ~~RSA Praktisch~~ * ~~Diffie Hellman~~ |

# 6 Steganography

Opdracht gemaakt door: Yoeri Odekerken & Ruben Jamart

Steganografie kan op verschillende manieren toegepast worden. Via [deze link](https://drive.google.com/file/d/1961bIKQPgKYKu__n9UTiI8KIU4YRsg8k/view?usp=sharing) krijg je bestanden met daarin een verborgen bericht. Kan jij de berichten vinden?

Verklaar hoe je het bericht hebt gevonden, hoe de methode werkt en waar je deze kan toepassen.

Opgelet: sommige technieken en tools die je kan gebruiken worden verduidelijkt in de opdracht THM→ Steganography.

|  |
| --- |
| Checklist   * figuur1.jpeg → Flag + verklaring * figuur2.jpeg → Flag + verklaring * BSOD.dub → Flag + verklaring |

# Figuur 1:A billboard outside of a building Description automatically generated with medium confidence

# 7 THM → steganography

Opdracht gemaakt door:: Alle teamleden

Checklist:

* ~~vervollediging van de room~~
* ~~Werking toegelicht.~~
* ~~Algemene conclusie over de technologieën en opgedane kennis van de THM-room.~~

# Graphical user interface, table Description automatically generated

hier was het makkelijk als je wat voorkennis van linux had je kon de dingen ook makkelijk opzoeken (maar je kent het normaal van de lessen linux). heb hier gebruik gemaakt van de manpages om op te zoeken zo kwam ik altijd tot een antwoord.

# Graphical user interface Description automatically generated

de bit vragen heb ik moeten opzoeken en heb zo zo kunnen oplossen de andere vragen kon je vinden in de bestanden die je gekregen hebt met behulp van het zsteg commando in linux.

Graphical user interface, application

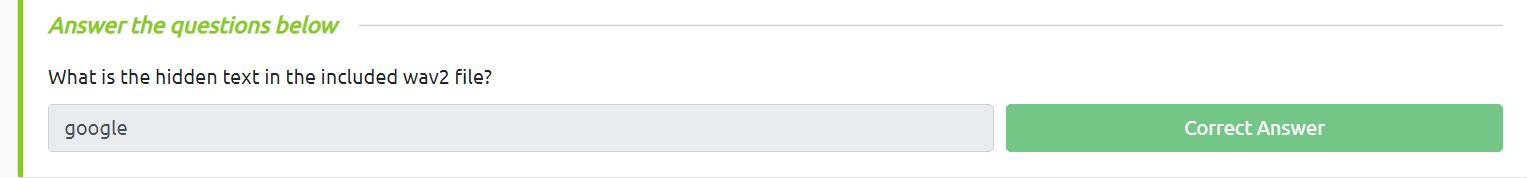
Description automatically generated

heb gebruik gemaakt van het exiftool in linux. zo ben ik tot een antwoord kunnen komen.

Table

Description automatically generated with low confidence

deze heb ik kunnen oplossen door wat info op te zoeken van stegoveritas. dit was zeer handig en zo heb ik deze kunnen oplossen heb de laatste vraag kunnen oplossen door het simpele linux commando cat.



dit was weer opzoeken in de mee gegeven files in het begin dit duurde even met het geluidsbestand maar na er even mee gespeeld te hebben en na wat opzoekwerk heb ik het antwoord kunnen vinden.

Background pattern

Description automatically generated with medium confidence

dit heb ik opgelost aan de hand van de files deze duurde wat langer door de geluidsfiles maar met wat opzoekwerk ben ik uiteindelijk tot een antwoord gekomen.

# Conclusie:

Je kan ook geheime boodschappen verstoppen in files. Het kan soms even duren voor deze gevonden zijn maar na lang ermee te “spelen” kan je de antwoorden wel vinden. Ik vind dit wel een leuke manier om geheime boodschappen te verzenden. Het is een speciale technologie (vind ik).  Over het 2e deel heb ik niet zoveel opgedaan omdat dit linux commando’s waren (deze moesten al gekend zijn voor het examen).

# 8 Kraken van de encryptie

Opdracht gemaakt door:: Wout Knoors & Ruben Jamart

Ontcijfer onderstaand tekst m.b.v een frequentieanalyse.  De zin is geschreven in het nederlands! Je vindt de [letterfrequentie van het nederlands online terug](https://onzetaal.nl/taaladvies/letterfrequentie-in-het-nederlands).

Bij voorkeur maak je gebruik van een zelfgeschreven script voor de code te kraken. Online tools kan je gebruiken om je oplossing te controleren, maar je moet de frequentie analyse zelf ook opzetten!

Cipher tekst:

|  |
| --- |
| Qci cu ssg jhhnzssyq jag ssg istui qcs hgyssuzaan cu wsfaati fsi zsxoyv jag uozuicioics.  Qs ysiisnu cg qsms istui mcrg jsnjagwsg qhhn agqsns ysiisnu. Qhhn wsznoct is fatsg jag ssg dnsbosgicsiazsy tag rs qsms istui isnow yssuzaan fatsg.  Mh tag rs qs fssui jhhnthfsgqs ysiisnu cg qsms istui jsnjagwsg qhhn qs fssui jhhnthfsgqs ysiisn jhywsgu qs dnsbosgicsiazsy. Qs fssui jhhnthfsgqs ysiisn cg hgms iaay cu qs ‘s’.  Rs may mcsg qai rs ugsy ssg yssuzans istui ocithfi.  Ssg zcrthfsgqs jnaaw jhhn qsms hvqnaexi cu, xhs tag rs qsms jhnf jag sgenpvics uisntsn fatsg?  Qs dyaw qcs zcr qsms hvqnaexi xhhni cu {Avvsyfhsu}. |

a = a; b = z; c = e; d = q; e = s; f = d; g = w; h = x; i = c; j = r; k = t; l = y; m = f; n = g; o = h; p = v; q = b; r = n; s = u; t = i; u = o; v = j; y = p; z = m

|  |
| --- |
| Checklist:   * Script/code voor het decoderen.      * ~~Antwoord op de vraag.~~ * ~~FLAG = {\*\*\*\*\*\*\*\*}~~   FLAG = {appelmoes} |

# Uitkomst: dit is een voorbeeld van een tekst die onleesbaar is gemaakt met behulp van substitutie. de letters in deze tekst zijn vervangen door andere letters. door gebruik te maken van een frequentietabel kan je deze tekst terug leesbaar maken. zo kan je de meest voorkomende letters in deze tekst vervangen door de meest voorkomende letter volgens de frequentietabel. de meest voorkomende letter in onze taal is de ‘e’. je zal zien dat je snel een leesbare tekst uitkomt. Een bijkomende vraag voor deze opdracht is, hoe kan je deze vorm van encryptie sterker maken? de flag die bij deze opdracht hoort is {Appelmoes}.

# Link naar bestand: https://mega.nz/file/sgJxRZRD#PVsj5sN1YFg5epvL3SZa1f-SER7lL1099o52KdAnnjY

# 9 THM → crypto 101

Opdracht gemaakt door: Alle teamleden

Checklist:

* ~~vervollediging van de room~~
* ~~Werking toegelicht.~~
* ~~Algemene conclusie over de technologieën en opgedane kennis van de THM-room.~~

# A picture containing background pattern Description automatically generated

Eerste vraag was gewoon op een knopje drukken, vraag 2 heb ik informatie over SSH keys opgezocht en bleek dat er alleen passphrases gebruikt worden.

Background pattern

Description automatically generated with medium confidence

Vraag 1 heb ik moeten opzoeken en heb ik direct gevonden, vraag 2 duurde wat langer omdat ik geen antwoord kon vinden. Uiteindelijk is dat wel gelukt. Vraag 3 stond in de tekst uitgelegd.

Background pattern

Description automatically generated with medium confidence

Voor deze vragen te vinden heb ik in pycharm een python bestand aangemaakt om alles in te geven. Zo krijg je direct het antwoord.

Table

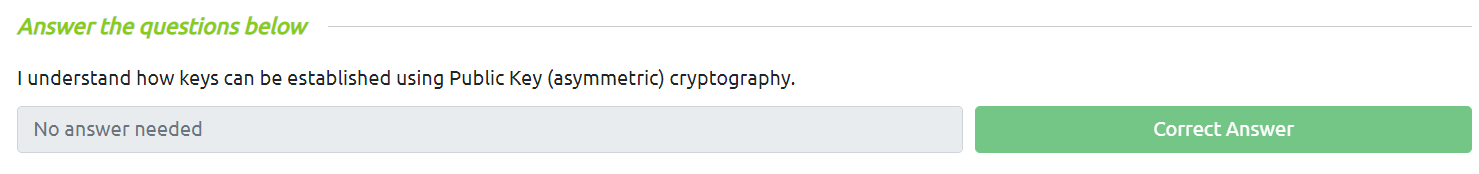
Description automatically generated with low confidence

Vraag 1 heb ik nee tegen gezegd omdat op het internet stond dat ze regelmatig aangevallen worden. Vraag 2 stond bij dezelfde website uitgelegd. Vraag 3 is redelijk duidelijk, het heet een “public key” dus die is publiek.

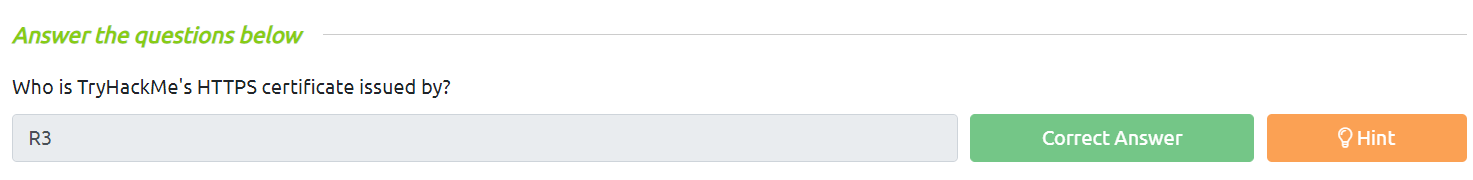
A picture containing text

Description automatically generated

Vraag 1: n is het product van p en q: 4391 \* 6659 = 29239669. Vraag 2 is gewoon op de knop drukken.



Gewoon op de knop drukken.

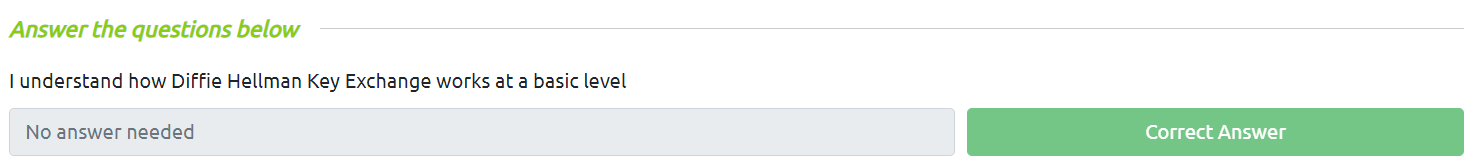


Deze vraag heeft even geduurd… Ik heb de certificate moeten vinden van TryHackMe en daarvan de common name.

Graphical user interface

Description automatically generated with medium confidence

Vraag 1 & 2 gewoon knop drukken. Vraag 3 stond uitgelegd in te tekst. Vraag 4 moet je iets downloaden en in linux vanalles doen: Eerst unzip je het bestand en moet je een john command gebruiken om het te cracken.

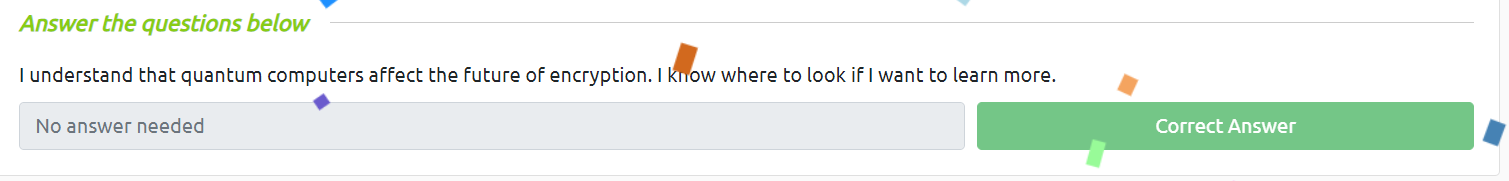


Gewoon op de knop drukken.

Graphical user interface, text

Description automatically generated with medium confidence

Eerst moest je een bestand downloaden, dan druk je op de knop. Bij vraag 2 moet je weer linux gebruiken. Eerst moet je het bestand unzippen, dan importeer je de key in GPG en kan je het message bestand decrypten.



Gewoon op de knop drukken.

# 10 Crypto challenges (extra)

Opdracht gemaakt door: Iedereen

Extra opdracht:

* Vervolledig één of meerdere crypto challenges op:
* - Hack the box: Je kan de challenges vinden via [deze link](https://app.hackthebox.com/challenges) onder het item “crypto”. Voorzie in het verslag de nodige uitleg hoe je de challenge aangepakt hebt & wat het resultaat is.

* - Tryhackme: Je kan de challenges vinden via [deze link](https://tryhackme.com/hacktivities?tab=search) en vervolgens zoek je als term naar “crypto”. Voorzie in het verslag de nodige uitleg hoe je de challenge aangepakt hebt & wat het resultaat is.

* Maak zelf een crypto

challenge voor één van de andere teamleden.

* Hack the box:
  + *BabyEncryption:*

We passen de code aan zodat we de code kunnen ontcijferenA screenshot of a computer

Description automatically generated with medium confidence

We voeren de code van de bovenste screenshot uit en we hebben de message ontcijferd.

Message: “Look at you reversing equations congrats”.

* Tryhackme:

CTF Collection Vol.1

Task 2:

Bij deze opdracht moesten we de onderstaande code decoderen. Ik heb                             gemaakt van Cyberchef met het algoritme Base64.Graphical user interface, text, application, chat or text message

Description automatically generated

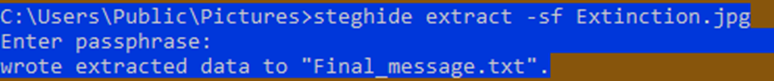
Task 3:

Graphical user interface, text, application, chat or text message

Description automatically generated

De taak was om de flag uit een image te halen. M.b.v. Notepad heb ik de flag kunnen vinden.

Task 4:



Graphical user interface, text, application, Word

Description automatically generated

Bij de volgende opdracht heb ik steghide moeten gebruiken voor de flag  van de image te vinden.Graphical user interface, text, application, chat or text message

Description automatically generated

Task 5:

Hier moest de block tekst highlighted worden om de flag te zien.Graphical user interface, text, application, chat or text message

Description automatically generated

Task 6:

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

Vanuit een qr-code werd the flag weergeven.

Task 7:

A picture containing table

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Via Cyberchef heb ik de flag kunnen vinden van het bestand.

Task 8:

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

.Via cyberchef a.d.h.v. het Base58 algoritme de flag gevonden.

Task 9:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

.Via cyberchef a.d.h.v. het ROT13 algoritme de flag gevonden.

Task 10:



Graphical user interface, text, application, chat or text message

Description automatically generated

Ik heb gekeken in de html-code voor de flag.

Task 11:

Graphical user interface, text, application

Description automatically generated

Task 12:

Logo

Description automatically generated with low confidence

Graphical user interface, text, application, chat or text message

Description automatically generated

De flag kun je vinden op het reddit account van Tryhackme.

Task 13:

A picture containing text

Description automatically generated

Via brainfuck decoder de flag opgehaald.

Task 14:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Eerst zorgen we dat  S1 in raw text staat. Daarna gebruiken we S2 als key om onze flag te krijgen.

Task 15:

A picture containing rectangle

Description automatically generated

Task 16:

Graphical user interface

Description automatically generated

Rectangle

Description automatically generated with medium confidence

Hiervoor moest ik een photoshop programma gebruiken en de helderheid van de foto omlaag te zetten om zodanig de flag te zien.

Task 17:

Graphical user interface, text, application, chat or text message

Description automatically generated

Hiervoor moest ik een luisterfragment openen en horen wat de flag was.

Task 18:

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Met behulp van deze website heb ik de juiste flag kunnen vinden.

Task 19:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Geëncrypteerde code decrypten met en zodanig de flag krijgen.

Task 20:

Graphical user interface, text, application

Description automatically generated

Eerst moeten we decimaal naar hexadecimaal gaan.

Graphical user interface, text, application, chat or text message

Description automatically generated

Nadat we het hexadecimale getal hebben berekend, kunnen we het omzetten naar een flag.

Task 21:

Graphical user interface, text

Description automatically generated

Hiervoor heb ik in Wireshark moeten werken met het bestand wat bij de task zat. Na een paar stappen te doen, heb ik uiteindelijk de flag gevonden.

Checklist:

* ~~Werking toegelicht van elke (extra) challenge.~~

# 11 VPN

gedaan door: Yoeri Odekerken & Bavo ‘s Heeren

Part 1 Onderzoek \*

Gebruik een VPN (<https://protonvpn.com/>) en onderzoek onderstaande items.

Doe onderstaande items met en zonder VPN en verklaar het verschil.

1. Toon aan wat je IP-address (whatsmyipaddres.com)

***De IP-address verschillen van elkaar.***

1. Traceroute PXL.be

***De traceroute verschilt, met VPN is het een veel langere route dan zonder VPN.***

1. Connecteer met [htww.pandora.com/tps://w](https://www.pandora.com/)   Tracht met VPN wel verbinding te maken.

***Zonder VPN is het niet mogelijk om hier op te serven, met VPN wel.***

1. Doe een snelheidstest

***Zonder VPN is de internetsnelheid beter dan met VPN.***

1. …..

Part 2 Vragen m.b.t VPN \*

1. VPN = V…..P…..N…..

Virtual Private Network

1. Wat is VPN?  Beantwoord de vraag aan de hand van de cybersecurity cube.

**data in transit: want dankzij een vpn heb je veel privacy. het zorgt ervoor dat je moeilijker te traceren bent. het blokkeert de uitgaande data om opgevangen te worden door een 3e persoon hierdoor is je uitgaande data veiliger.**

1. Waar of wanneer een VPN gebruiken? Geef ook aan waarom een VPN een oplossing biedt en hoe het conceptueel werkt.

**een vpn werkt als volgt: een vpn versleutelt je netwerk verkeer door het om te leiden naar een ander land bv je zit in belgië en je gebruikt een vpn en zo proberen je te traceren vinden ze je bv in japan. hierdoor surf je veiliger en je hebt ook voordelen bv dingen die enkel in het andere land te verkrijgen zijn kan je gebruik van maken.**

1. Waar kan je VPN toepassen in je privé / studentenleven?

**als je op netflix films wilt kijken die enkel in het buitenland beschikbaar zijn kan je ze wel zien als je een vpn gebruikt.**

1. Waar kan je VPN toepassen in je schoolomgeving?

**de school kan door gebruik van een vpn bepaalde sites blokkeren voor de leerlingen.**

1. Waar kan je VPN toepassen in je werkomgeving?

**je kan van thuis uit veilig op het LAN netwerk van een bedrijf  werken.**

1. Schets VPN in het OSI model.  (Binnen welk laag/lagen wordt VPN gebruikt en verklaar.)

**laag 3 (netwerklaag) :  met laag 3 bereikt de gegevensoverdracht het internet. In de netwerklaag worden de eindapparaten logisch geadresseerd. Hieraan wordt in laag 3 een eenduidig IP-adres toegewezen. Aan een datapakket.**

**laag 4 (transportlaag) : de transportlaag fungeert als schakel tussen de toepassing georiënteerde en de transport georiënteerde lagen. Op dit niveau van het OSI-model wordt de logische end-to-end verbinding, het overdrachts kanaal, tussen de communicerende systemen gerealiseerd.**

1. Er bestaan twee types VPN remote access VPNs & Site-to-site VPNs.   Wat is het verschil tussen deze twee en waar worden ze toegepast?

**In Remote Access VPN: individuele gebruikers zijn  verbonden met het privénetwerk en het is  in staat om op afstand toegang te krijgen tot de diensten en bronnen van dat privénetwerk.(wordt gebruikt in bedrijven voor mensen die van thuis uit werken.) Site-to-Site vpn: ook Router-to-router genoemd hier wordt gebruik gemaakt van Ipsec beveiliging. gebruikt om een ​​gecodeerde tunnel te creëren van het ene klant netwerk naar de externe site van de klant.(worden gebruikt om externe kantoren te verbinden met het hoofdkantoor je kan ook meerdere huizen met elkaar verbinden).**

*\*Opmerking: De items uit het onderzoek zijn voornamelijk toepassingen op ‘geolocatie en IP-adres masking’.   Beperk de antwoorden op vragen niet tot enkel deze toepassing van VPN.*

Part 3 Conclusie

* Omschrijf kort en krachtig wat een VPN is en waarvoor het gebruikt wordt.

**een vpn beveiligd je netwerk verkeer door je adres om te leiden naar een ander land bv je zit in belgië en als ze je traceren komen ze uit in China. je kan het vergelijken met een tunnel je begint in belgië en je eindigt ergens anders. Een vpn wordt gebruikt voor veiliger te surfen, buitenlandse film te kijken, moeilijker getraceerd te worden of dingen opzoeken die enkel in een bepaald land zijn. het wordt ook gebruikt in het bedrijfsleven voor veiliger te werken op het netwerk van dat bedrijf.**

|  |
| --- |
| Checklist:   * Part 1 * Part 2 * Part 3 |

# 12 Hashes!

Door Bren & bavo

* Kies een bestand en een wachtwoord.

bestand: pxl.txt

wachtwoord: pxl123

* Geef van het bestand en het wachtwoord de hash volgens de volgende algoritmes;   **Whirlpool,SHA2,SHA3, MD4, MD5, HMAC, bcrypt, scrypt.**

Whirlpool:

bestand:**A4A81F94591204CC5F18058732F98909F16151A91D31BE015A148619992CCAB57B494FECFFC47D37E3CB2C105DFDDF1201C3227F0F9A0AC05708610F50822049**

wachtwoord:**6F50D54676C8E1430CFB46C0F73F6883BC170A6E0FF863C97F8B3AD6A6008AE73222DE81A078584F5161E49BE51D1B069721CAEC6D7DA29886D09C3C6D03D448**

SHA256: bestand:**4214d092c66e3f8da00becb7fb62e575876befac92596b046f62573104032190**

wachtwoord:**a5ebd4c270e26ccd706ec3ace34b9ba2ad9500f51bb4c036fb1c8f8814a701b8**

SHA3-256:

bestand:**5440411184c07b9dfda828867746838069eb13ad4b8c6d38528fae25c9b69659**

wachtwoord:**df2a9203b0f63e26220c027a9585418d8f83b1f5f913feac928ee6bdf38bd08e**

MD4:

bestand:**acab7705fd1905be55e1e2cc7e12f737**

wachtwoord:**b798406290a4a6b00b63850f2859d3b2**

MD5:

bestand:**6401fcdc1365399d3002d240fc4c4638**

wachtwoord:**3d0db0f8b043bcba5a2cdbb63dbc3dec**

HMAC-SHA256:

**secret key: security**

bestand:**3c1f09997772bd4b3c0d35027c26a1f745da844c23902583231fce883e9f3e05**

wachtwoord:**94933542e8d8bfed01ed1cb63531e9f21841b67229495aa26b31f3591927a959**

bcrypt:

bestand:**$2a$10$nUw.Jyks6sAmX7w9JS0gO.sRoxYRTaBIwmVDyx953bid01Nws9Liy**

wachtwoord:**$2a$10$SZRwbexJH/zBXXBUSVW5buNq7uH0l8yDa/.KFSTSR1dcFN3NW.5/i**

scrypt:

bestand:**250e306960d1c7f24a2ff934a7815d2471859b2c77d57ef9019b79e1d81e7bcb85e6faf5830ecb07d1e8e2ad1b3ad2f44dd36660cd9492dc49467950921a2b74**

wachtwoord:**10b70ff50e1bf10375059110367558f4555b967226cae212abf81a46d5704305f36a3d1bc9743801af0c5b24ce95aa02cd8146c1c9c9552b1529df450d97870e**

* Geef in het kort een  conclusie per algoritme (wat zie je aan de hash, hoe  kan je de hash herkennen, wat zijn secure hashes en welke niet, kan je één van de not secure hashes omkeren of beïnvloeden.).

Whirlpool:

je ziet vooral hoofdletters en cijfers. is een zeer lange hash.

is secure

SHA2:

je ziet vooral kleine letters en cijfers. niet al te lange hash.

SHA3:

bestaat vooral uit cijfers en een paar kleine letters. geen super lange hash.

MD4:

lijk vooral op MD5

niet secure

heeft 32 karakters.

MD5:

herkennen: 32 karakters

niet meer secure tegenwoordig:omkeerbaar.

HMAC:

je hebt een security key nodig het bestaat vooral uit cijfers en paar kleine letters. is niet super lang.

bcrypt:

niet zo lange cipher die vooral uit kleine en hoofdletters bestaat.

scrypt:

lange cipher die vooral uit cijfers bestaat

* Vervolledig de tabel en geef een algemene conclusie!  Over het gebruik van hashing algoritmes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Year** | **Secure?** | **Output size** | **Rounds** |
| Whirlpool | 2003 | Yes | 512 bits | 10 |
| SHA2 | 2001 | Yes | 224/256/384/512 bits | 64/80 |
| SHA3 | 2016 | Yes | 224/256/384/512 bits | 24 |
| MD4 | 1990 | No | 128 bits | 3 |
| MD5 | 1992 | No | 512 bits | 4 |
| HMAC | 1996 | Yes | variabel | variabel |
| bcrypt | 1999 | Yes | 192 bits | variabel |
| scrypt | 2009 | Yes | variabel | variabel |

**Het zijn allemaal goede opties en kunnen zowat door elkaar gebruikt worden maar hier zijn een paar adviezen:**

**Het gebruik van MD4 en MD5 is afgeraden omdat ze niet meer veilig zijn, gebruik deze alleen nog voor integriteits-checks.**

**scrypt is het moeilijkst te kraken maar neemt ook het meeste resources in gebruik.**

* Pas salting toe de hashes van de paswoorden.
  + Wat verandert er aan de hash?

**Salting is het toevoegen van een willekeurige string tot een paswoord voordat je de het hashed.**

* Wat heeft dit als gevolg voor het kraken van de hash?

**de hash wordt extra beveiligd dankzij het salting. de salting zorgt ervoor dat de hash niet te kraken is met een vooraf berekende tabel.**

* Wat is het verschil met peper?

**Pepper is hetzelfde in concept maar wordt toegevoegd nadat het paswoord al gehashed is.**

|  |
| --- |
| Checklist:   * Ingevulde tabel * Conclusie per algoritme * Algemene conclusie * Info en omschrijving salting |

# 13 Collisions and hashes

gemaakt door: Bavo & Wout Knoors

Wat is er bijzonder aan de volgende twee figuren? → [Download](https://drive.google.com/file/d/19KT5FqrfS2xkZOvXQaT1AYxHmYuguE3s/view?usp=sharing)

Wat is er bijzonder aan de volgende twee PDF documenten?  → [Download](https://drive.google.com/file/d/19EEDbkkS2-AByfT8fcK27HUZeQoe-Np9/view?usp=sharing)

1. Wat valt je op aan de figuren? Zoek een verklaring voor je bevindingen.

het zijn beide foto’s van iets dat gecrashed is (iets dat is misgelopen). dit komt omdat er niet goed gecommuniceerd is of er is op het moment zelf iets misgelopen of iets niet goed berekend waardoor dit fout loopt.

1. Wat valt je op aan de pdf-documenten.

het zijn beide hetzelfde enkel de achtergrondkleur is bij 1 bestand rood en bij het ander bestand blauw.

1. Wat is een collision en wat zijn de gevolgen hiervan?

2 bestande of wachtwoorden met dezelfde hash. bv ( je upload je examen en iemand heeft dezelfde hash als jou maar er staat een ander bestand in dan kan het zijn dat ze het verkeerde bestand nakijken en kan je in de problemen geraken.)

1. Wat is de oplossing om collisions tegen te gaan?

een grotere of meer secure hash gebruiken.

1. Hoe zou je zelf 2 bestanden genereren met dezelfde hash?

eventueel een klein programmaatje maken via python of eventueel via java. Deze kan dan de hashes genereren.

Checklist:

* ~~6 vragen Beantwoord~~

# 14 THM → ‘Crack the hash” Door: Iedereen

# Checklist:

* Room level 1 completed
* Room level 2 completed
* werkwijze verklaard. (Niet enkel printscreens van de oplossing)

# Graphical user interface Description automatically generated

1. Deze tekst is een MD5 en kun je converteren naar het woord “easy”. Dit hebben we gedaan met een online converter.
2. Deze tekst is een SHA1 en kun je converteren naar het woord “password123”. Dit hebben we gedaan met een online converter.
3. Deze tekst is een SHA2-256 en kun je converteren naar het woord “letmain”. Dit hebben we gedaan met een online converter.
4. Deze tekst is een Bcrypt en kun je converteren naar het woord “Bleh”. Hiervoor gebruikten we het commando: “hashcat -m 3200 tryhackme.txt rockyou.txt”
5. Deze tekst is een MD4 en kun je converteren naar het woord “Eternity22”. Dit hebben we gedaan met een online converter.

A picture containing graphical user interface

Description automatically generated

* Deze tekst is een SHA2-256 en kun je converteren naar het woord “paule”. Dit hebben we gedaan met een online converter.
* Deze tekst is NTLM en hiervoor moet je het commando “hashcat -m 1000 tryhackme.txt rockyou.txt” om het te converteren naar: “n63umy8lkf4i”.
* Deze tekst is SHA512crypt en hiervoor gebruikten we een online converter. Het paswoord is: “waka99”
* Voor de laatste is de tekst HMAC-SHA1 en gebruikten we het commando: “echo e5d8870e5bdd26602cab8dbe07a942c8669e56d6:tryhackme> tryhackme.txt”.

# 15 Hashing challenge

Door: Wout, Yoeri, Bavo

## 15.1 Easy to crack

Hashes 1-5

1. 5f4dcc3b5aa765d61d8327deb882cf99
2. 81dc9bdb52d04dc20036dbd8313ed055
3. d8578edf8458ce06fbc5bb76a58c5ca4
4. 098f6bcd4621d373cade4e832627b4f6
5. 03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4 (sha 256)

Vragen step 1

1. Welke algoritme is gebruikt voor de hash en wat is het wachtwoord?

1e  **MD5  =>  password**

2e   **MD5 => 1234**

3e   **MD5  => qwerty**

4e   **MD5  => test**

5e  **SHA 256  =>  1234**

1. Welke methode(s) heb je gebruikt om de hash te kraken?

1e  **een online generator die de md5 omzet naar gewone tekst.**

2e **een online generator ik gebruikte de site hash toolkit (zeer handig je geeft de hash in en hij zegt wat voor hash het is wat de boodschap is en ook nog andere hashes)**

3e  **een online generator hash toolkit**

4e   **een online generator hash toolkit**

5e  **ik heb hierbij ook een online generator gebruikt om deze te achterhalen (hash toolkit).**

1. Waarom zijn deze hashes makkelijk te kraken?

omdat je tegenwoordig online generators hebt, deze kraken de hashes automatisch voor je je hoeft zo zelf niet te doen en je weet meteen wat er staat.

1. Als we de eerste hash (5f4dcc3b5aa765d61d8327deb882cf99) een karakter wijzigen (naar 6f4dcc3b5aa765d61d8327deb882cf99).
   1. Is de output ongeveer hetzelfde?

neen als de hash veranderd zal de output niet het zelfde zijn .

1. Is deze dan nog steeds makkelijk te kraken?

neen omdat deze hash niet bestaat.

15.2 Rainbow table

Gegeven: [md5\_woordenlijst](https://drive.google.com/file/d/196fA0l6LPsuEoc_gvR_Wvs7mF1RnMm7g/view?usp=sharing) +

* **d8ac13f95359d2a45256d312676193b3**
* **54b1c483a93051f1719ad0572b3d414e**
* **65f0480470eec8d12d8629b5dcad2f72**
* **1ea63d4a03928f95d7028a4dd4fa9426**
* **701f33b8d1366cde9cb3822256a62c01**
* **a2a551a6458a8de22446cc76d639a9e9**
* **b95351311be5bb96359e6568afda299c**

Gevraagd: Kraak de hashes door gebruik te maken van de woordenlijst en een (eigen) script.  (PS of Python of bash of bat).   
Bijvoorbeeld door gebruik te maken van een .bat file:   
 FIND “hashvalue” md5\_woordenlijst.txt >> antwoord.txt     
 ;>> voegt content toe aan een file.

Uitbreiding: Vervolledig het script zodat je niet één hash kan opzoeken maar al de onderstaande hashes kan opzoeken met 1 klik.

## 15.3 Herken de hash

[Meer info over verschillende hashes](https://hashcat.net/wiki/doku.php?id=example_hashes)

* e91e6348157868de9dd8b25c81aebfb9   => **MD5**
* 5d2d3ceb7abe552344276d47d36a8175b7aeb250a9bf0bf00e850cd23ecf2e43  =>**SHA256**
* fd193fcf2511e9ba3a879e3a691edc375f3d1b0dcef1d59e50ef6df9fb09c873bccf4732bb63c402346d91d7447f7a56db393b9c6aae92406a9a89ee6a79fd4e  =>**Whirlpool**
* $2a$10$If3huPOIxrpjWySZ9eoToeCUr8yLQHuGMVhjH2fZ8aO1LgsrnUt6i  =>**bcrypt**

## 15.4 Hashcat

Installeer hashcat en bekijk de help pagina (--help)

General:

-m = hashtype

-a = attackmode (straight; combination;togglecase;

-o = output

**Opdracht HASHCAT**

Step1:  Hashes die voorkomen in rockyou.txt

* cfc723a39d137f8428d8eeec9cebd384
* 204d7916206f2797d71ab26f523d931f
* e75bbc3294ceffd90fd6febd13769b61
* 6707efa899fa779af0f165cfe23b8569

Step2: ander type van hash die voorkomen in rockyou.txt of [wordlistNL](https://github.com/OpenTaal/opentaal-wordlist)

* 1e5fcfc4e60fcad03f92260202359caf0a7b8d8d0955b7b0020f158150b5305f

Step 3:  Rule based hashes die voorkomen in [wordlistNL](https://github.com/OpenTaal/opentaal-wordlist)

* 7f86d1368c2aa6e9bfe21654cf75351e
* 40267b17d04ff2344344635be1ecceee
* aabf01f3220280000e830d7f14c84cee
* 30f16456b85fab3a016e07dc2f1a2fa2
* 4951f0eed8d251980ef8497ed9ce2bf7

15.5 Advanced cracking

We hebben een hash van een wachtwoord van onze goede vriend Ernest.

→ Genereer zelf een woordenlijst met de woorden vanuit de omschrijving.

→ Gebruik rules om karakters om te zetten en cijfers toe te voegen.

* Hash => “het wachtwoord is vrij in te gegeven”
  + 93ac53bbc9470bd1f934b0c208f69090
  + e81c602454bbe574f7802456e2297495
  + 3f2460dea519c0ff07835805fd084ac9
  + 4e6df07e9c56569e614163994c82ce27
* Hash =>  met  password policy: min 8 tekens, 1 hoofdletter, 1 getal & 1 speciaal karakter’   (Tip:  a=@, E=3, i=1, s=5, t={, o=0, 8=&, …)
  + a616c2b8838d7e09d74704df47029356
  + 9c63e2e15ed5ff7e19d0039bdc77f1df
  + 5e88214fba455805992b5c3fdff4d39f
* Hash => ‘Good password??!!” (mogelijks niet te kraken… Dit is een lang paswoord, met verschillende tekens, hoofdletters en cijfers, maar het heeft nog steeds verband met Ernest.)
  + 3a524555cba2dcf356580fbe975c4e4d

## 15.6 Hashes en wachtwoorden

Wat kan je concluderen over het kraken van hashes.

1. Is een hashfunctie nog steeds een one-way-function? + verklaar
2. Bestaan er hashes die niet te kraken zijn. + verklaar
3. Hoe ga je hiermee om m.b.t. paswoorden opslaan?  + verklaar
4. Waaraan moet een wachtwoord voldoen, reken houden met deze oefening?
5. Past de eerder omschreven password policy nog steeds?
6. Wat kan je doen om rainbow table attacks tegen te gaan.  + verklaar

Checklist:

* 15.1: 5\*wachtwoord gegeven + antwoord op 4 vragen.
* 15.2: 7 wachtwoorden + script & werking + conclusie.
* 15.3: 3\*type van de hash
* 15.4: 4+1+5 x wachtwoord + korte verklaring werkwijze in Hashcat
* 15.5: 4 x hash “wachtwoord vrijgegeven” + 3x hash met policy + conclusie
* 15.6: antwoord op de zes vragen.

# 16 Beveiligde files

Door:Wout Knoors

Gegeven is deze link naar [een beveiligd word document](https://drive.google.com/file/d/16Wq8oyH7o3yuX2nEmEWYBQecYksRLBTG/view?usp=sharing). Download het bestand. Kan jij het wachtwoord kraken?

Geef in het verslag:

Inhoud:

**PXL{Cr4ck\_Th3\_P4ssw0rd}**

Afgelegde weg:

**Ik gebruikte een online tool namelijk** [**https://www.lostmypass.com/**](https://www.lostmypass.com/)**.**

**De tool heeft verschillende aanvallen ingebouwd afhankelijk van hoe moeilijk het is om het  wachtwoord te kraken. In dit geval was het een heel makkelijk wachtwoord namelijk “password123” dus was de eerste aanval zijnde een dictionary attack de enige die nodig was.**

**indien het meer secure was had het ook een brute force attack toegepast.**

conclusie:

**Dit was een heel insecure wachtwoord wat ik ook via brute force had kunnen  vinden  als ik even had geprobeerd, maar er zijn ook een aantal online tools die dit heel makkelijk en in seconden kunnen oplossen.**

Checklist:

* ~~Inhoud van het bestand~~
* ~~Werkwijze~~
* ~~Conclusie~~

# 17 SSL Certificates

Door Ruben Jamart

SSL certificaten kunnen op verschillende manieren ingesteld worden. Test de volgende sites met [ssllabs](https://www.ssllabs.com/ssltest/index.html) en vergelijk de resultaten. Indien er verbeteringen mogelijk zijn lijst die dan ook op voor elke site.

1. [www.ikea.com/](https://www.ikea.com/)
2. [www.pxl.be](http://www.pxl.be)
3. [www.howest.be](http://www.howest.be)
4. [www.njtf.cn](http://www.njtf.cn)
5. [www.xjufe.edu.cn](http://www.xjufe.edu.cn)  \*

\* Deze site zal op SSLlabs een melding geven dat er iets verkeerd is, gebruik de optie om deze te negeren en alsnog een volledig rapport te genereren.

Gevraagd:

1. Screenshot per website in bijlage.
2. Bespreking in eigen woorden over de score, met name WAAROM iets wel/niet goed is en op welke technologie dit terugslaat.
3. Als er bepaalde concepten zijn waarom het misgaat die je niet kent, verwachten we daar een uitleg van.
4. Bij de minpunten, geef ook een concrete oplossing

(screenshots zie bijlagen onderaan)

**Bij Ikea**: Hier is er sprake over een uitstekende beveiliging!

**Bij PXl**: Uitwisseling van de encryption key is niet zo goed. Dit komt omdat Diffie Hellman de keys heeft uitgewisseld.

**Bij Howest**: Er worden verouderde protocollen gebruikt zoals TLS 1.0 en 1.1!

**Bij NJTF**: De servers gebruiken SSL 2 & 3 die helaas niet meer veilig zijn + Diffie Hellman heeft de encryption keys weer uitgewisseld.

**Bij XJUFE**: Certificaat van de servers is onbetrouwbaar.

Besluit: Het is noodzakelijk dat het certificaat van een betrouwbare bron komt en alle protocollen geupdate worden.

Checklist:

* ~~Bespreking van de score → Waarom wordt deze score gegeven → wat kan er beter.~~
* ~~screenshots in bijlage.~~

# 18 SSL and github

Door Ruben Jamart & Yoeri Odekerken

<https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/>

Maak een **private** repository op GitHub met als naam “SecEss\_<groepsnummer>”. Je hebt al gezien dat je via het git clone commando een repository kan clonen naar je systeem. Dit kan je ook doen via SSH. Volg de documentatie via [deze link](https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh) om een nieuw key-pair aan te maken (tip: gebruik geen passphrase).

Maak screenshots van het genereren van de keys, het toevoegen van de keys aan GitHub & de output van het Git clone commando via SSH.

Bespreek in het verslag de werkwijze & het nut van de SSH keys. Voorzie een duidelijke conclusie!

Checklist:

* ~~Github communicatie via SSH is ok. (Toon aan)~~
* ~~Werking toegelicht.~~

# <https://github.com/TheHooba/SecEss-4.git>

**Waarom is communicatie via SSH goed?**

SSH biedt veel veiligheid voor het uitdelen van gegevens en de communicatie tussen verschillende systemen. Daarnaast is een groot voordeel van SSH dat authenticatie met een publiek en geheim sleutelpaar mogelijk is. Github is een handige platform hiervoor die gratis voor iedereen is en waar je praktisch alles mee kan.

**Toelichting werking:**

Ten eerste moet je op je github een nieuwe repository maken.

Graphical user interface, application

Description automatically generated

Die noem je SecEss\_4. Zet die ook op private.

A screenshot of a computer

Description automatically generated with medium confidence

Daarna moet je checken of er een SSH key al gereed is. Dit doe je door het het te klonen naar je eigen PC of laptop.

Maak een nieuwe folder aan ergens op je PC en noem die “SecEss\_4”. Daar open je Git Bash door rechter muisklik te doen in een lege plek en daarna “Git Bash Here” aan te klikken. Dit opent Git Bash. Daarna vul je volgende code in:   
“git clone <https://github.com/TheHooba/SecEss_4.git>”.  
Als je volgende melding krijgt mag je die negeren.

Text

Description automatically generated  
Als je nu in de folder gaat kijken en onder view “Hidden Items” aan vinkt hoor je normaal een .git folder te zien. Dat betekent dat je succesvol gekloond hebt. Nu kan je checken of je al een SSH key hebt.

Ga terug naar Git Bash en vul de volgende code in: “ls -al ~/.ssh”



Als je deze melding krijgt heb je nog geen SSH key. Dus die moeten we aanmaken.

Open Git Bash weer en typ: “ssh-keygen -t ed25519 -C " [je-email@email.com](mailto:je-email@email.com)"

Text

Description automatically generated

Als je dit krijgt moet je op Enter drukken.

Text

Description automatically generated

Vul hier je wachtwoord in. (je kiest deze zelf). Je gaat het niet kunnen zien maar hij typt het wel in. Deze ga je 2 keer moeten ingeven. Onthoud deze goed.Text

Description automatically generated

Als je dit scherm krijgt is het gelukt. Je hebt een SSH key gegenereerd.

Daarna moet je de SSH key toevoegen aan je Github account.

Kopieer de SSH key door in je Git Bash het volgende te typen: clip < ~/.ssh/id\_ed25519.pub”.  
Ga naar je browser en ga naar Github.com. Ga daarna naar instellingen.

Onder “Access” ga je naar “SSH and GPG keys”. Klik op “New SSH key”.  
Geef een titel naar keuze en doe dan rechtermuisklik in het vakje voor Key en selecteer plakken. Je mag daarna op “Add SSH key” drukken.

**Als je de SSH connectie wilt testen mag je het volgende doen:**

Ga terug naar Git Bash en typ: “ssh -T git@github.com”.

Vul daarna het wachtwoord in dat je eerder hebt gekozen.

Text

Description automatically generated

Als je deze melding krijgt is alles gelukt.

# 19 SSL and SSH

Door: Ruben Jamart & Yoeri Odekerken

* Vertrek vanuit 2 virtuele linux machines. Op VM1 voeg je een user toe met als  username VM1Groep<Groepsnummer>, VM2 voeg je de user met als username VM2Groep<Groepsnummer> toe en installeer je het software pakket openssh-server (sudo apt-get install openssh-server). Via het [ssh commando](https://www.ssh.com/ssh/command/) kan je verbinding maken van VM1 naar VM2 a.d.h.v. de username/password combinatie van de user op VM2.

* Probeer opnieuw verbinding te maken met het ssh commando van VM1 naar VM2 maar gebruik deze keer een private/public keypair. Op het internet kan je verschillende guides vinden hoe je dit kan doen. De private key staat op VM1, de public key op VM2. Als alles goed gaat, kan je verbinding maken van VM1 naar VM2 zonder gebruik van wachtwoorden.

1. Geef het bewijs dat je via ssh kan inloggen van VM1 naar VM2 a.d.h.v. username & password.
2. Geef screenshots van het genereren/kopiëren van de keys en het bewijs dat je de private key vanop VM1 gebruikt om op VM2 in te loggen zonder wachtwoord.

Bespreek in het verslag de werkwijze & het nut van de SSH keys. Voorzie een duidelijke conclusie!

1. Nota: Andere dag gedaan => ander ip-adres dan de volgende screenshots.
2. Text

   Description automatically generated

key genereren/kopiëren

Text

Description automatically generated

Text

Description automatically generated

Private key gebruiken

Graphical user interface, application

Description automatically generated

Inloggen zonder wachtwoord:

Text

Description automatically generated

Werking:

Ik heb op VM1 een private & public key gegenereerd. Ik heb de public key gekopieerd naar vm2 waardoor ik op vm1 kan inloggen op vm2. Deze keys kunnen handig zijn om de hele inlog proces makkelijker te maken, daarnaast is nog eens extra veilig doordat je een public & private key gebruikt.

Checklist:

* ~~communicatie via SSH is ok. (Toon aan)~~
* ~~Werking toegelicht.~~

BIJLAGEN

oef 17

ikea

Graphical user interface, application

Description automatically generated

PXL

Graphical user interface

Description automatically generated

Howest

Graphical user interface, application

Description automatically generated

njtf

Graphical user interface, application

Description automatically generated

xjufe

Graphical user interface

Description automatically generated