Veilig Programmeren

STRIDE, Encryption, Brute Force, XSS (Cross Site Scripting), input control (regular ex pressies), error catching, logging, password rules, and more
Gegeven aan OITAOO8B Periode 1 en 2 van leerjaar 2019/20

- STRIDE
- Encryptie Algoritme en Sleutel
- ASCII
- Brute Force
- Login
- Objecten - extra uitleg
- Encryptie - Theorieles
- Case: Stuxnet Worm 2010
- Case: SecureHomes
- Risiso's en Maatregelen
- Case: Efteling, vrijkaartjes winnen
- Case FinTech BV.
- Nikto test
- Create Safe Code - SQL Injection
- PEN test 1 - CentOS - SSH
- ChatGPT

STRIDE

Safety First?

Niet altijd 'safety first'. Er moet een goede balans zijn tussen veilig en usability.

Daarnaast geldt dat als je het te ingewikkeld maakt voor gebruikers dat het dan onveiliger kan worden. Bijvoorbeeld als je een wachtwoord niet meer kan onthouden dan moet je het wel opschrijven.

Uit een Berceley University Study blijkt dat:

- 70% van de gebruikers vergeet hun wachtwoord
- 90% van de gebruikers die het wachtwoor dniet meer weten lopen weg van de site
- 40% van de gebruikers schrijft hun wachtwoorden op
- Wachtwoorden worden door een gemiddlede gebruiker 7-9 keer opnieuw gebruikt op andere sites

> Advies: gebruik een goede wachtwoord manager en gebruik voor elke site een uniek wachtwoord.

Liefst geen bekende/bestaande woorden. We zullen later nog zien waarom niet.

STRIDE

Bij ontwerp van veilige software moet je aan de volgende zaken denken:

- **S**poofing
- **T**ampering
- **R**epudiation
- **I**nformation Disclosure
- **D**enial of Service

- **E**levation of Privilige

> Let op, STRIDE moet je kennen voor je cijfer voor it keuzevak.

Spoofing

Spoofing is je voordoen als iemand anders. Dat kan als je iemand zijn password op een of andere manier achterhaald. Dat kan op vele manieren: keyloaggers, phissing, camera's plaatsen, social engineering, brute force attack, ...

Tampering

Tampering is het veranderen van data. Dat kan bijvoorbeeld door SQL injection (data in de database veranderen), maar ook door achterdeurtjes te gebruiken om op een systeem binnen te komen en daarna gegevens te veranderen. Door spoofing toe te passen (je voordoen als iemand anders) kun je ook gegevens aanpassen.

Repudiation

"*I did not do that*!", is iets doen en dan zeggen dat jij het niet gedaan hebt. Bijvoorbeed geld opnemen en dan tegen de bank zeggen dat jij het niet hebt gedaan. De bank heeft dan bijvoorbeeld video-opnamen om te bewijzen dat jij het wel bent geweest.

Non repudiation betekent dat je een systeem zo maakt dat iemand niet of heel moeilijk kan ontkennen dat hij iets heeft gedaan. Log files kunnen daarbij heel handig zijn.

Information Discluse

Geheime of gevoelige informatie blootleggen. Denk aan Wiki leaks of aan het Facebooks Cambridge-Analytica schandaal.

Denial of Service

De 'bekende' DDOS aanvallen. Dit betekent zoveel verkeer genereren dat een webserver 'omvalt'. Dit wordt meestal met BOTs uitgevoerd. Dit is een virus dat op vele (miljoenen) computers is verspreid en dat op commando verkeer naar dezelfde site stuurt.

Jouw computer zou deel uit kunnen maken van een BOT netwerk, daar merk je over het algemeen weinig van.

Elevation of Privildge

Jouw rechten verhogen. Dus je bent een gewone gebruiker en je weet jezelf de rechten van de administrator of in een Unix-omgeving van root eigen te maken.

Encryptie Algoritme en Sleutel

Veel threads (=bedreigingen) die zijn beschreven in het STRIDE model kunnen worden tgengehouden of worden verminderd door encryptie teo te passen.

We gaan ons wat meer verdiepen in encryptie, we leren:

- Wat (symetrische) encryptie is;
- hoe werkt een brute force attack werkt;
- en hoe je een veilig wachtwoord maakt.

In deze les leren we wat een encryptie algorime is en wat een encryptie-sleutel is.

Algoritme

Encryptie werd al toegepast ver voordat de eerste computer er was. De meest eenvoudige encryptie werkt per karakter. Elk karakter wordt omgezet naar een ander karakter. Een a wordt bijvoorbeeld een b, een b een c , een c en d en ga zo maar door. Het algoritme in dit geval is: 'neem telkens het volgende karakter in het alfabet', en om d eboodschap terug te vertalen (te ontcijferen) zou het algrotme zijn: 'neem telkens het vorige karakter uit het alfabet'.

Nu kan je dat met een stap grootte van 1 doen (neem het voglende karakter), maar je kunt het ook met bijvoorbeeld 3 doen, dus a wordt dan d en b wordt e, c wordt f en ga zo maar door. Je kunt het ook met 12 doen, of met 23.

Dit verschuiven van de letters met een stapgrootte x, noemen we het algoritme. Dit is de *manier* waarop we gaan encrypten.

Sleutel

Dit getal, zeg maar de stap grootte, noemen we de sleutel. In principe is het algorime altijd bekend maar is de sleutel geheim.

> Vraag: hoeveel verschillende sluetels denk je dat er zijn bij het hierboven beschreven algoritme?

Het encryptie algoritme en de sleutel samen zorgen voor de encryptie.

Voordat we het algoritme gaan maken dat een boodschap kan coderen en decoderen volgens het hierboven beschreven algoritme moeten we even een stukje theorie behandelen. Dit doen we in het volgende hoofdstuk.

ASCII

We gaan leren wat ASCII is en hoe we met ASCII waarden in PHP kunnen werken.

Karakter

Wat is een karakter. Dat is één letter, cijfer of teken op de computer. Dus alle tekens die je ziet zijn een karakter, ook een spatie.

ASCII tabel

Alle karakters hebben een numerieke waarde. Dat komt omdat een computer alleen maar getallen (in nullen en enen) kan onthouden. Dus alles in een computer wordt omgezet in nummers. In de ASCII tabel kun je vinden welk nummer bij welk karakter hoort. In onderstaande tabel (http://www.asciitable.com/) zie je welk nummer bij welk teken hoort.

De waarde onder Dec is de decimale waarde en dat is de waarde waar wij het eenvoudigst mee kunnen werken. Kijk in de tabel waar het getal 9 staat. Zie je dat de ASCII waarde 57 is? En het uitoepteken heeft een waarde van 33.

Ascii Table

Je ziet dat alle letters netjes op alfabetische volgeorde onder elkaar staan. De B is 1 groter dan de de A. Dus als we willen schuiven met letters dan hoeven we alleen maar de ASCII waarde van een letter te nemen en er N (1,2,3 of meer) bij op te tellen.

PHP

Je moet twee fiuncties in PHP kennen:

`ord()`

Hiermee maken we van een karakter een cijfer, dus ord('Z') geeft het getal 9.

`chr()`

Hiermee zetten we een nummer om in een karakter, dus chr(89) geeft een letter 'Y'.

We kunnen nu dus een karakter omzetten in een nummer, er iets bij op tellen en het nieuwe getal weer omzetten in een karakter.

> Vraag: wat doen we als we bijvoorbeeld 1 willen optellen bij de letter 'Z'

Opdracht Coderen

Als je het antwoord op deze vraag weet dan ben weet je alles om een encryptie algortime te maken dat letters verschuifd met N. Het enige dat je nog moet weten is dat je spaties ongemoeid laat. Spaties worden dus niet encrypt.

Dus maak een algortime dan een string en een nummer als input krijgt en dat de string encrypt door de afzondelijke letters x posities op te schuiven. Bijvoorbeeld, stel je object heet Crypt en je method heet encrypt, dan:

```
Crypt->encrypt('ABC ABC', 1) geeft als return value: 'BCD BCD'
```

O aj nog een laatste puntje. Laten we alleen hoofdletters nemen om één en ander neit te ingewikkeld te maken. Dus alles wat we gaan encrypten zetten we eerst in upper case met de PHP functie `strtoupper($string)`

Succes!

Brute Force

Als het goed is hebben we nu een encryptie algoritme. Het algoritme verschuifd de letters x posities. Als je voorbij de Z komt dan begin het weer bij de A. Dus als je de X 5 posities verschuifd dan reulteert dat in een C. Tel maar na.

Dit betekent ook dat als je 26 posities vershuifd je weer terug komt bij dezelfde letter. Of als je 27 cijfers verschuifd is dat hetzelfde als 1 letter vershuiven,

Stel we hebben de volgende tekst:

```
HTP HPPE HLE STPC DELLE RPDNSCPGPY
```

Je weet deze boodschap is gemaakt met het 'verschuif-algoritme'. Je weet alleen de sleutel niet. Hoeveel sleutels zijn er ook alweer? Slecht 25, want bij 26 kom je weer uit op originele boodschap en 27 is hetzelfde als 1.

Dus maak nu een loop waarin je alle sleutels probeert. Decodeer de bovenstaande boodschap door alle mogelijke sleutels te proberen.

> Wat was de originele boodschap?

> Welke sleutel is gebruikt?

Je ziet dat je door alle sleutels te proberen heel snel de orginele boodschap vindt. Dit proberen van alle sleutels heet brute force, brute kracht. Gewoon alles proberen. Bij 25 sleutels lukt dat vrij makkelijk maar als we de sleutels groter maken dan wordt brute force steeds moeilijker.

Symmetrische encryptie

In dit voorbeeld gebruikte we hetzelfde algoritme en eigenlijk ook dezelfde sleutel om een boodschap te encytpen (te versleutelen) en om een boodschap te de-crypten (te ontcijferen). Eigenlijk was het niet helemaal dezelfde sleutel want als we met +1 hadden encrypt dan hadden we -1 of 25 nodig om de decrypten. Maar eigenlijk horen 1 en -1 bij elkaar. Het is immers dezelde sleutel met een ander teken. Als we bij encrypten en decrypten dezelde sleutel gebruiken dan hebben we het over een symmetrische encryptie. Zodra je de sleutel weet dan kan je de code kraken. Dit is anders bij asymmetrische encryptie.

Het voordeel van symmetrische encryptie is dat het lekker snel is om te encrypten en te decrypten. Het nadeel is echter dat als je de sleutel eenmaal weet, dat je dan alle boodschappen kunt ontcijferen.

Asymmetrische encryptie

Bij asymmetrische encryptie is dit anders. Het is werkt wat ingewikkelder en is daardoor wat trager, maar als je de sleutel onderschept, kun je de boodschappen nog niet ontcijferen.

Sleutelgrootte

Omdat we in ons algoritme maar 26 letters hadden, hadden we een keuze uit 1..25 voor de sleutel. Als we hoofd en kleine elttes samen nemen dan zouden we wel wat mer sleutels hebben, maar het schiet nog niet op.

Het schiet wel op als we twee karakters als één teken zien. We gaan dus eignelijk twee karakters omzetten in een nummer. We gebruiken dan geen ASCII tabel meer maar maken onze eigen tabel. AA=0, AB=1, AC=2,.....BA=27, BB= 28, etc. Op deze menier hebben we 676 mogelijke combinaties. Di is 26x26. Als we een groepje van drie karakters nemen dan hebben we al 17576 mogelijke combinaties en dit nummer groeit snel. Als we een groepje van 8 karakters nemen dan hebben we 208 miljard mogelijkehden voor een sleutel, Als we alle sleutels zouden wilen proberen en we kunnen er 1000 per seconden testen dan zouden we ruim in het selchtse geval 6.5 jaar bezig zijn om de sleutel te vinden.

> Vraag: waarom 'in het slechtse geval' zal het 6.5 jaar duren, kan het ook minder lang duren?

Toch is deze methode van letters verschuiven zelfs als we het groepjes doen, niet heel goed.

Ten eerste moet de sleutel heel goed geheim gehouden worden en dat terwijl je hem in eerste instantie wel moet delen. Je zult de sleutel dus een keer moeten opsturen.

Ten tweede kun je met behulp van statistieken voorspellen. Je weet bijvoorbeeld dat de letter e veel vaker voorkomt dan de q. Met die gegevens kun je veel gerichter op zoek gaan naar de sleutel. Zeker als je veel verdleutelde data hebt.

Later gaan we nog eens kijknen naar andere vormen van encryptie. Nu gaan we eerst eens terug nar ons STRIDE model.

Hieronder is een lijst van gebruikersnamen waarmee brute force via SSH op een Linux server is geprobeerd binnen te komen.

```
/var/log/auth.log.1:Dec  5 02:49:52 vps789715 sshd[446357]: Invalid user julia from
45.155.204.39 port 39582
/var/log/auth.log.1:Dec  5 02:49:56 vps789715 sshd[446359]: Invalid user kermit from
45.155.204.39 port 5758
/var/log/auth.log.1:Dec  5 02:50:00 vps789715 sshd[446361]: Invalid user kernel from
45.155.204.39 port 22205
```

```
/var/log/auth.log.1:Dec  5 22:13:50 vps789715 sshd[452901]: Invalid user dietpi from
91.223.67.146 port 7236
/var/log/auth.log.1:Dec  5 22:13:55 vps789715 sshd[452903]: Invalid user pi from 91.223.67.146
port 44171
/var/log/auth.log.1:Dec  5 22:13:59 vps789715 sshd[452905]: Invalid user openhabian from
91.223.67.146 port 24953
/var/log/auth.log.1:Dec  6 02:07:53 vps789715 sshd[453971]: Invalid user admin from
45.155.204.39 port 30624
/var/log/auth.log.1:Dec  6 02:07:56 vps789715 sshd[453973]: Invalid user akiwifi from
45.155.204.39 port 38556
/var/log/auth.log.1:Dec  6 02:07:59 vps789715 sshd[453975]: Invalid user config from
45.155.204.39 port 43557
/var/log/auth.log.1:Dec  6 21:36:39 vps789715 sshd[460616]: Invalid user carlos from
91.223.67.146 port 14752
/var/log/auth.log.1:Dec  6 21:36:42 vps789715 sshd[460618]: Invalid user admin from
91.223.67.146 port 22747
/var/log/auth.log.1:Dec  6 21:36:45 vps789715 sshd[460621]: Invalid user informix from
91.223.67.146 port 28668
/var/log/auth.log.1:Dec  7 01:14:09 vps789715 sshd[462043]: Invalid user admin from
45.155.204.39 port 6832
/var/log/auth.log.1:Dec  7 01:14:12 vps789715 sshd[462045]: Invalid user admin from
45.155.204.39 port 16685
/var/log/auth.log.1:Dec  7 01:14:14 vps789715 sshd[462047]: Invalid user miguel from
45.155.204.39 port 22317
/var/log/auth.log.1:Dec  7 22:24:53 vps789715 sshd[469657]: Invalid user abella from
91.223.67.146 port 41510
/var/log/auth.log.1:Dec  7 22:24:56 vps789715 sshd[469659]: Invalid user antonio from
91.223.67.146 port 30818
/var/log/auth.log.1:Dec  8 01:39:51 vps789715 sshd[470718]: Invalid user admin from
45.155.204.39 port 53722
/var/log/auth.log.1:Dec  8 01:39:54 vps789715 sshd[470720]: Invalid user xml from
45.155.204.39 port 16602
/var/log/auth.log.1:Dec  8 01:39:58 vps789715 sshd[470722]: Invalid user gns3 from
45.155.204.39 port 36374
/var/log/auth.log.1:Dec  8 22:38:56 vps789715 sshd[480723]: Invalid user linktechs from
91.223.67.146 port 39741
/var/log/auth.log.1:Dec  8 22:38:59 vps789715 sshd[480725]: Invalid user localadmin from
91.223.67.146 port 59244
/var/log/auth.log.1:Dec  8 22:39:02 vps789715 sshd[480792]: Invalid user login from
91.223.67.146 port 28668
```

```
/var/log/auth.log.1:Dec  8 22:39:05 vps789715 sshd[480794]: Invalid user login from
91.223.67.146 port 37396
/var/log/auth.log.1:Dec  8 22:39:09 vps789715 sshd[480796]: Invalid user mac from
91.223.67.146 port 39889
/var/log/auth.log.1:Dec  9 06:40:56 vps789715 sshd[493250]: Invalid user netgear from
45.155.204.39 port 30211
/var/log/auth.log.1:Dec  9 06:40:59 vps789715 sshd[493252]: Invalid user admin from
45.155.204.39 port 33234
/var/log/auth.log.1:Dec  9 06:41:06 vps789715 sshd[493256]: Invalid user lena from
45.155.204.39 port 39205
/var/log/auth.log.1:Dec  9 06:41:10 vps789715 sshd[493258]: Invalid user linaro from
45.155.204.39 port 41967
/var/log/auth.log.1:Dec  9 23:02:03 vps789715 sshd[506238]: Invalid user mira from
91.223.67.146 port 14146
/var/log/auth.log.1:Dec  9 23:02:07 vps789715 sshd[506240]: Invalid user mother from
91.223.67.146 port 42884
/var/log/auth.log.1:Dec  9 23:02:10 vps789715 sshd[506242]: Invalid user music from
91.223.67.146 port 10022
/var/log/auth.log.1:Dec  9 23:02:17 vps789715 sshd[506246]: Invalid user natalia from
91.223.67.146 port 38994
/var/log/auth.log.1:Dec 10 06:44:23 vps789715 sshd[508843]: Invalid user mailto from
45.155.204.39 port 22873
/var/log/auth.log.1:Dec 10 06:44:27 vps789715 sshd[508845]: Invalid user manager from
45.155.204.39 port 25270
/var/log/auth.log.1:Dec 10 06:44:29 vps789715 sshd[508847]: Invalid user media from
45.155.204.39 port 26638
/var/log/auth.log.1:Dec 10 06:44:34 vps789715 sshd[508849]: Invalid user michael from
45.155.204.39 port 29436
/var/log/auth.log.1:Dec 10 06:44:36 vps789715 sshd[508851]: Invalid user michelle from
45.155.204.39 port 30938
/var/log/auth.log.1:Dec 10 23:17:07 vps789715 sshd[515371]: Invalid user optiproerp from
91.223.67.146 port 20045
/var/log/auth.log.1:Dec 10 23:17:11 vps789715 sshd[515373]: Invalid user pablo from
91.223.67.146 port 21729
/var/log/auth.log.1:Dec 10 23:17:14 vps789715 sshd[515375]: Invalid user patrol from
91.223.67.146 port 15725
/var/log/auth.log.1:Dec 10 23:17:18 vps789715 sshd[515377]: Invalid user pc1 from
91.223.67.146 port 1109
/var/log/auth.log.1:Dec 10 23:17:23 vps789715 sshd[515379]: Invalid user pedro from
91.223.67.146 port 42140
```

```
/var/log/auth.log.1:Dec 11 06:37:48 vps789715 sshd[518052]: Invalid user nelson from
45.155.204.39 port 15185
/var/log/auth.log.1:Dec 11 06:37:50 vps789715 sshd[518054]: Invalid user netgear from
45.155.204.39 port 17289
/var/log/auth.log.1:Dec 11 06:37:54 vps789715 sshd[518056]: Invalid user nico from
45.155.204.39 port 21331
/var/log/auth.log.1:Dec 11 06:37:57 vps789715 sshd[518058]: Invalid user nsa from
45.155.204.39 port 25242
/var/log/auth.log.1:Dec 11 06:38:00 vps789715 sshd[518060]: Invalid user operator from
45.155.204.39 port 29352
/var/log/auth.log.1:Dec 11 16:44:43 vps789715 sshd[521894]: Invalid user u2078299 from
62.194.183.33 port 54137
/var/log/auth.log.1:Dec 11 16:47:07 vps789715 sshd[521898]: Invalid user u2078299 from
62.194.183.33 port 54143
```

Login

We gaan een veilige en goede login maken

We gaan leren hoe in PHP een gebruiker wordt geauthoriseerd via een login en user id.

We kennen drie delen:

1. Login form
2. De control page (php)
3. De database connectie waarmee het user_id en password worden gecontroleerd.

Maak de files

```
login.html
login.php
db.php
```

We beginnen met de db.php. We maken nog geen connectie met de databse maar defeiniëren voorlopig een assiociative array met user-id en met het niet-encrypted wachtwoord. We gaan dit later uiteraard aanpassen, maar om één en ander te testen is dit nu beter en sneller.

```
// vul het array aan met je eigen login
$logins = [ 'Damon' => 'geheim12', 'Su-Yen' => 'secret123', 'Bob' => 'grotegroep3',
□□□'Omar' => 'Laptop12', 'JariF' => 'verweg-42' ];
```

Maak nu een object login met de private property username en één public functie:

```
Login->authenticate($username, $password);
```

Weet je nog hoe he een object maakt? Maak eerst de class, definiëer de property username en maak de methods/functies. Instantieer de class door er een object van te maken.

Als je niet meer weet hoe dat precies zat met classes en objects, dan kun je bijvoorbeeld kijken naar:

https://www.youtube.com/watch?v=pFz180MbCM8

https://www.roc.ovh/books/veilig-programmeren/page/objecten---extra-uitleg

>

> Vraag 1: waarom staat Login hierboven met een hoofdletter?

Deze functie (of eigenlijk method) returned Trueof False; true als de combinatie username en password klopt en anders False.

Maak nu het form en post de login en userid naar login.php. Check eerst of de variabelen uit het form bekend zijn in login.php door deze af te drukken. Als dat het geval is, check dan of de login en userid combinatie klopt.

Nu is het zaak dat de browser onthoud dat de gebruiker is aangelogd. Hoe doen we dat ook alweer?

Inderdaad met een cookie.

```
setcookie('login', 'True', 1200);
```

> Vraag 2: waar staat de 1200 voor?

> Vraag 3: werkt dit?

> Vraag 4: waarom is dit niet veilig?

Session variabele

We gaan een sessie opbouwen. Een sessie is een stukje geheugen op de webserver waarin gegevens van jou worden bijgehouden. We starten een sessie op met:

```
session_start();
```

Nu kunnen we de status van de authenticatie (of we succesvol zijn aangelogd) in een sessie variabele op de server vastleggen.

```
$_SESSION['user'] = $username;
$_SESSION['login'] = 'True';
```

> Vraag 5: Leg uit waarom het bijhouden van of iemand is aangelogd via een sessie variable wel veilig is (in tegenstelling tot het vastleggen van de inlog status via een cookie).

Deze variabelen worden nu op de server opgeslagen en elke keer als we een pagina vragen worden deze variabele mee gestuurd naar de browser zodat je ze kunt uitlezen.

```
echo $_SESSION['user'];
```

Drukt de user name af.

Maak nu een nieuwe pagina, waarin je laat zien of er iemand is aangelogd en als dat zo is wie dat is. Gebruik hiervoor de sessie variabelen.

De output van de pagina ziet er dus als volgt uit:

> `Er ie niemand aangelogd`
>
> of
>
> `JariF is aangelogd`

Bekijk in je browser welke cookies er worden gebruikt, herken je de geheime sessie key?

> Vraag 6: waarom is de deze sessie key zo lang?

> Stuur de file login.php via teams op (opdracht op tijd klaar is 1 punt).

Opdracht

Verander nu het array zodanig dat alle wachtwoorden encypted worden opgeslagen (in het array). Gebruik hiervoor de php functie `crypt()`. Zoek op hoe deze werkt. Je kunt de functie gebuiken zonder 'salt'. Dit gaan we later behandelen.

Maak alle nodige veranderingen zodat je de gebruikers kan authenticeren aan de hand van de encrypted passwords.

Objecten - extra uitleg

Deze pagina is een copy van:
http://www.sitemasters.be/tutorials/1/1/607/PHP/OOP_(Object_Oriented_Programming)

In deze tutorial hoop ik uitleg te geven over hoe met een begin kan maken met het object georiënteerd programmeren (OOP). Ik ga ervan uit dat je al wat basiskennis hebt van PHP en dat je gebruik maakt van PHP 5.3 of hoger.

Objecten zijn “toewijzingen” van classes. Een class is een soort van 'blauwdruk', waarin je de logica van het object begint. Dit klinkt lastig, maar ik zal het uitleggen met een voorbeeld. Ik zal het voorbeeld nemen van een user-module. In deze module gaan we een aantal dingen stoppen. We gaan de mogelijkheid geven om meerdere eigenschappen op te geven en willen de mogelijkheid geven voor een user om iets te zeggen.

In OOP zou dit er zo uitzien:

```
<?php
Class User {
    public $name;
    public $age;

    public function __construct($name,$age) {
        $this->name = $name;
        $this->age = $age;
    }

    public function say($message) {
        echo $this->name . ': ' . $message . '<br />';
    }

    public function sayAge() {
        $this->say('Ik ben ' . $this->age . ' jaar.');
    }
}



$piet = new User('Piet',23);
```

```
$klaas = new User('Klaas',26);
$bert = new User('Bert',29);

$piet->say('Bert, alles goed?');
$bert->say('Jawel man!');
$klaas->say('Hoe oud ben jij Piet?');
$piet->sayAge();
?>
```

Er wordt een Class 'User' aangemaakt. In deze class geef je allerlei mogelijkheden (in dit geval om een naam/leeftijd op te geven, om iets te willekeurigs te zeggen en om je leeftijd te zeggen.

Onder de class maak ik drie objecten aan ($piet, $klaas en $bert). Op het moment dat ik zeg 'new User', weet PHP dat ik een class bedoel en gaat hij op zoek naar de class. Omdat ik achter newUser ('Piet',21) heb gezet gaat hij op zoek naar de __construct method (die is de standaard 'init' functie) en hij verwacht 2 gegevens (naam en leeftijd). Hierdoor komt de class eigenlijk tot leven en worden zijn alle variabelen/functies tot mijn beschikking bij Piet. Daarna doe ik hetzelfde voor de andere personen en uiteindelijk heb ik dus drie gebruikers.

Nu kan ik via het object ($piet, $berg en $klaas) de methodes aanroepen en ook de variabelen. Als ik $piet->say('iets'); opgeef betekent dit dat ik de 'say' method aanroep van het object piet. Piet gaat hierdoor praten en je ziet zijn naam staan, omdat het object de naam heeft gekregen bij het initialiseren (__construct).

Ik zei ook dat je eventueel de variabelen direct aan zou kunnen roepen. Dit kan door het object aan te roepen en de variabelenaam erachter. Als ik de leeftijd van piet wil weten, kan ik deze zo ophalen:

`$piet->age;`

Bij de methode 'sayAge' zien we iets anders wat heel belangrijk is, en dat is een speciale variabele '$this'. In deze variabele zitten alle variabelen en methodes van het object. Dus als je in het object $piet zit en je gebruikt $this, zit daar alles in van het object piet. Dus je zou kunnen doen $this->say(), $this->name, $this->age. Dit geeft je dus de mogelijkheid om vanuit één method, andere methods aan te roepen die binnen deze class zitten.

Voor ingevulde variabelen/argumenten
Het is de mogelijkheid (net als bij normale functies) om voor ingevulde variabelen op te geven bij de methods. Deze hoef je dan niet verplicht mee te geven als je de method aanroept.

```
<?php

public function setGender($gender = 'M') {
```

```
    $this->gender = $gender;
}


?>
```

Als ik in dit voorbeeld $piet->setGender(); zou opgeven, dan wordt zijn gender (geslacht) op 'M' gezet.

Encryptie - Theorieles

Wat is Encryptie?

Encryption (of encryptie of versleutelen) is het versleutelen van gegevens zodat je zonder de juiste sleutel de gegevens niet kan lezen. Encryptie bestaat uit het algortime (de regels/procedure van de encryptie) en een of meer sleutels.

Voorbeeld van encrypted data.

```
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALhzGuSqLXPRAz1MJzhvTdHMTCTp
tkvusbSVmVRTL7YmW5ICIW570+8uzQX5+16KoXT/B5s5ZnNgTq9VvmrAbjuHw+bL
dQnB+SPRdg9vKTfJOWRLonyCzKS9T/wSVW9yTPuaol2++4HR6LF2P2ifwvgj1aq2
UYZXcu2TmgdDMA+JXyREV8kRRnXxUbaeq6+2Ypnl1+0qIGqAL3mWpZ5CC1/nWCIR
vMi0ASOwYPnjcu86FwIcf/24hoHFsdp2eKosI+IoV23JXGhhXeKKROeEz85kezpx
```

Decryption (of decryptie of ontsleutelen) is het weer leesbaar maken van encypted data.

Waarom Encryptie?

Encryptie is belangrijk omdat het kan helpen voorkomen dat informatie in verkeerde handen valt.

Stel je krijgt toegang tot de database van Facebook en je kunt alle logins en passwords lezen. Dat zou heel vervelend zijn, maar het is iets minder vervelend als de wachtwoorden zijn versleuteld en het zou nog minder erg zijn als alle gegevens zouden zijn versleuteld. Of stel je telefoon of laptop word gestolen. Vervelend, maar als je gegevens op de harddisk (of SSD) zijn versleuteld kan niemand jouw gegevens lezen.

Door encryptie worden bijna alle risico's op de threats uit het STRIDE model verminderd. Dat geld voor de threats;

> *spoofing, tampering, repudiation, confidentielity, en authorization,*

> Vraag 1: Hoe kan encryptie het risico op repudiation ('I did not send that email') helpen vermindeeren?

Encryptie is niet alleen iets van computers ; het is al 2000 jaar oud.

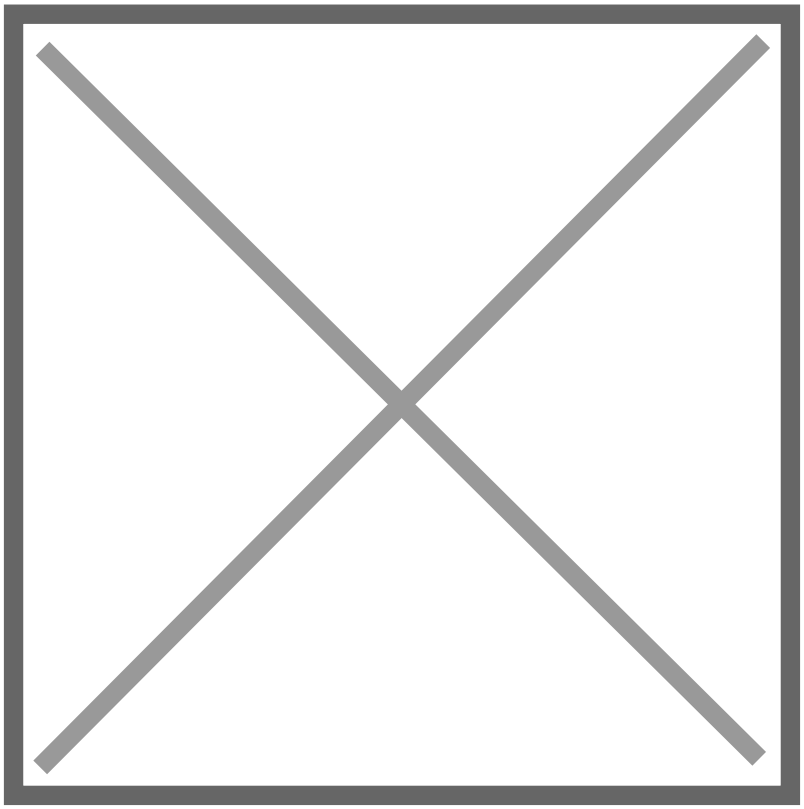
>

> Vaag 2: In welke situatie denk je dat 1000 of meer jaar geleden encryptie zou zijntoegepast? Als je niets kunt bedenken, denk eens terug aan "Game Of Thrones".

Symetrisch Encryptie

We maken onderscheid in symetrische encryptie en asymetrische encryptie.

Symetrische encryptie is eenvoudig en al heel oud. De meest eenvoudige vorm is het omzetten van elke letter in een ander letter. Je kunt letters verschuiven, maar je kunt het ook ingewikkelder maken. In de klas hebben we geoefend met Caesar encryptie.



Symetrisch encyptie is redelijk eenvoudig en daardoor ook realtief makkelijk en efficiënt op een computer te implementeren. Er is ook een groot nadeel:

> Bij symetrische encryptie moet je een sleutel uitwisselen en je moet daarbij voorkomen dat de sleutel in verkeerde handen valt.

PKI, Public Key Infrastructure

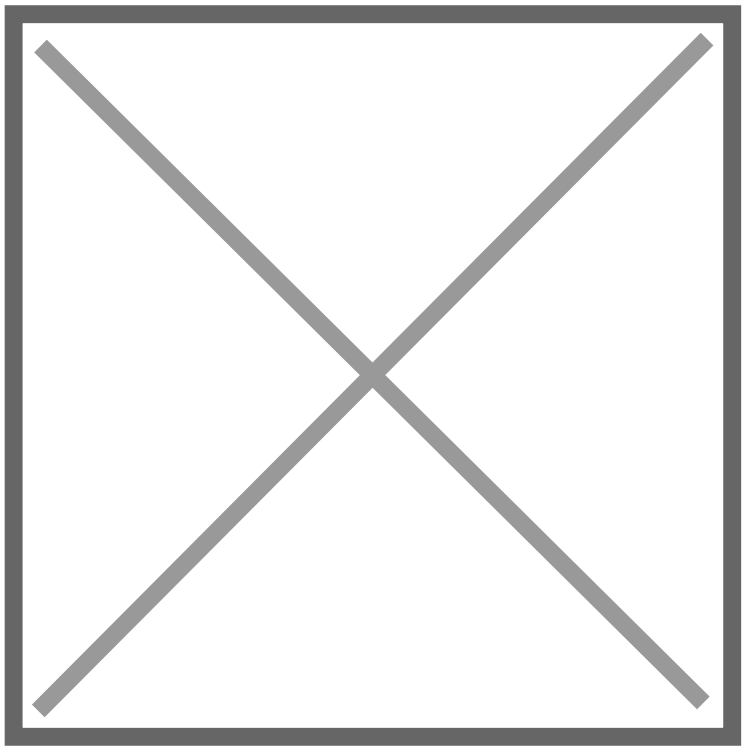
Het nadeel van het lastig uitwisselen van de key bij symetrische encryptie wordt opgelost bij PKI, *Public Key Infrastructure.*

PKI, is een systeem met twee sleutels: de public key en de private key (de publieke sleutel en de privé sleutel).

De public key kan je verspreiden, iedereen mag hem hebben, want je kunt er alleen een bericht mee versleutelen. Je kunt er geen bericht mee ontcijferen/lezen. Alleen met de bijbehorende private key kan het versleutelde bericht worden gelezen.

In de les hebben we dit zichbaar gemaakt met een (open) hangslot en de sleutel. Het open hangslot was de public key en die kon iedereen krijgen. Je kunt dan een boodschap sturen en alleen degene met de sluetel van het hangslot kan dan het bericht ontcijferen.

Of iets formeler (en let op: als we het over security en encryptie hebben dan hebben we het altijd over Alice en Bob).
Stel, Alice wil een bericht sturen aan Bob. Bob is in het bezit van een publieke sleutel en een privésleutel. Alice ontvangt dan de publieke sleutel van Bob. Hiermee versleutelt zij het bericht en daarna verstuurt ze het naar Bob. Bob ontsleutelt het bericht met zijn privésleutel en kan het dan lezen.



Dus je geeft idereen die dat wil je public sleutel en iedereen die dat wil kan een bericht versleutelen en jij bent de enige die met de private key het bericht kan ontsleutelen.

Of stel dat jij een wachtwoord wilt instellen op www.lekkergoedkoop.nl dan kun je met de public key van deze site je wachtwoord versleutelen en je weet dan zeker dat alleen de site www.lekkergoedkop.nl jouw versleutelde wachtwoord kan lezen. Maar is dat wel zo? Wat nu als je denkt dat je op de site www.lekkergoedkoop.nl zit maar via een slinkse manier (DNS Spoofing, Host file manipulatie, Phissing) op een andere site terecht ben gekomen. Je denk dat je je wachtwoord naar www.lekkergoedkoop.nl stuurt maar het is een hele ander site. Hoe weet je nu zeker dat je de

public key van de echte site hebt gekregen?

Certificaten

Een certificaat is digitaal ondertekend versleuteld document dat is uitgegeven door een Certified Authority. Je kunt het zien als een diploma dat is uitgegeven door een school. Niet iedere school mag zo maar een diploma uitreiken. Een digitaal certificaat mag ook alleen worden uitgegeven door bepaalde instanties en via bepaalde web sites.

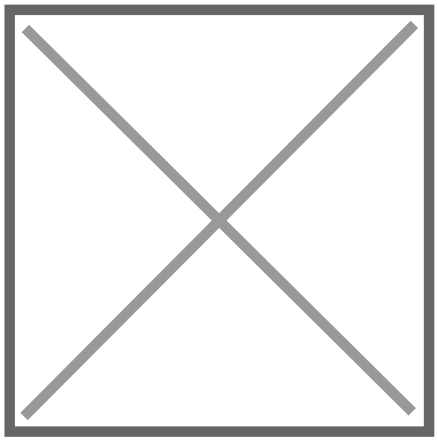
Als ik nu een site heb en ik wil dat de gebruiker een public key krijgt zodat wij veilig kunnen communiceren, dan kan ik deze public key via een certificaat uitdelen. Dit certificaat verklaart dan dat ik ben wie ik zeg dat ik ben en het certificaat bevat mijn public key.

Bij het aanvragen van een certificaat bij een CA, Certified Authority, wordt gecontroleerd wie jij bent en of jij een betrouwbare partij bent. Een certificaat heeft altijd een bepertke houdbaarheiddatum en zal dus na verloop van tijd moeten worden vernieuwd.

Dus nu kunnen we naar een website en kunnen we al het verkeer tussen mijn computer en de web server versleutelen via de private key die ik via een certificaat heb ontvangen. Alles opgelost of niet?

SSL

SSL staat voor **Secure Socket Layer** en het is standaard manier om verkeer tussen jouw browser en de webserver te versleutelen. Zodra er een SSL verbinding is opgezet tussen jouw browser en de webserver dan zie je een slotje of ander icoontje wat aangeeft dat je een veilige verbinding hebt. Ook veranderd het begin van de URL van HTTP in HTTPS.



Het opzetten van een veilige (=versleutelde) veriding kunnen we dat het beste doen via PKI waarbij de public key via een certificaat wordt opgevraagd. Maar PKI is complex en daardoor traag, als we al het verkeer telkens via PKI willen encrypten en decrypten dan gaat dat al snel voor vertraging zorgen. Symetrische encrypty is veel sneller maar daarbij was het probleem dat we niet

eenvoudig de sleutel konden verspreiden.

De oplossing is de combinatie van beide technieken.

We gebruiken PKI om een tijdelijke symetrische sleutel uit te wisselen die we daarna gebruiken voor al het SSL verkeer. Dus we gebruiken de 'trage' PKI methode om ervoor te zorgen dat we op een veilige manier de *symetrische* sleutel uitwisselen. Vanaf het moment dat de browser en de web server allebij dezelfde symetrische sleutel hebben kan het verkeer via symetrische encrypty veilig en snel worden opgezet. Zodra de web sessie eindigt (via uitloggen of via een time-out) wordt de symetrische key ook weggegooid en bij de volgende sessie wordt er een nieuwe symetrische sleutel uitgewisseld via PKI. Door telkens een neiuwe symetrische sleutel te gebruiken verlaag je de kans dat deze key uitlekt en door een hacker kan worden misbruikt.

In het kort: SSL en het Slotje geven aan dat de website is beveilgd met een SSL certificaat. SSL is alleen tegenwoordig eigenlijk TSL.

TSL

TLS staat voor **Transport Layer Security** en is een verbeterde en veiligere versie van SSL. Met de ontwikkeling van TLS zijn vele kwetsbaarheden die voorkomen in oude SSL protocollen verholpen en zijn nieuwe beveiligingsmechanismen toegevoegd. In de praktijk wordt eigenlijk allees TSL nog gebruikt maar we nomen het nof vaak SSL.

Vragen

1. In welke situatie denk je dat 1000 of meer jaar geleden encryptie zou zijn toegepast? Als je niets kunt bedenken, denk eens terug aan "Game Of Thrones".

2. Hoe kan encryptie het risico op repudiation ('I did not send that email') helpen verminderen?

3. Hoe kan encyptie het risico op tampering verminderen?

4. Noem een voorbeeld waarbij het niet erg is om een via een niet beveiligde verbinding (niet SSL) een web site te bezoeken.

5. Als je je wachtwoord op een site veranderd dan moet je je oude en nieuwe wachtwoord opgeven. Is het dan van belang om dit via een SSL verbiding te doen? Leg uit waarom!

6. Waar staat HTTPS voor *(alle* letters benoemen)?

7. Vul bij (a), (b), (c) en (d) (zie tabel hierboven) de voor- en nadelen van symetrische en asymtrische encryptie in:

	Voordeel	Nadeel
Symetrische encryptie	(a)	(b)
Asymetrische encryptie	(c)	(d)

Case: Stuxnet Worm 2010

Stuxnet Worm 2010: Iran's Nuclear Program Blocked

Video Youtube

Stuxnet is an extremely sophisticated computer worm that exploits multiple previously unknown Windows zero-day vulnerabilities to infect computers and spread. Its purpose was not just to infect PCs but to cause real-world physical effects. Specifically, it targets centrifuges used to produce the enriched uranium that powers nuclear weapons and reactors.

Stuxnet was first identified by the infosec community in 2010, but development on it probably began in 2005. Despite its unparalleled ability to spread and its widespread infection rate, Stuxnet does little or no harm to computers not involved in uranium enrichment. When it infects a computer, it checks to see if that computer is connected to specific models of programmable logic controllers (PLCs) manufactured by Siemens. PLCs are how computers interact with and control industrial machinery like uranium centrifuges. The worm then alters the PLCs' programming, resulting in the centrifuges being spun too quickly and for too long, damaging or destroying the delicate equipment in the process. While this is happening, the PLCs tell the controller computer that everything is working fine, making it difficult to detect or diagnose what's going wrong until it's too late.

...

In October 2011, Duqu came to light5. This is a descendent of Stuxnet. It used a zero-day exploit to install spyware that recorded keystrokes and other system information. It presages a resurgence of Stuxnet-like attacks but we have yet to see any version of Duqu built to cause cyber-sabotage.Various long term attacks against the petroleum industry, NGOs and the chemical industry6also came to light in 2011. And hactivism by Anonymous, LulzSec and others dominated security news in 2011

Opgaven

1. Wat is een *Windows zero-day vulnerabilitie*?
2. Wat is een computer worm?
3. Zoek op internet hoe deze worm (waarschijnlijk) het gebouw is binnengekomen.

4. Voor deze hack is er (waarschijnlijk) gebruik gemaakt van een zogenaamde rootkit, leg uit wat dat is.
5. Als je deze hacks leest aan welk€ OWASP Risks denk je dan?

Case: SecureHomes

Het bedrijf SecureHomes, gevestigd in de regio Rotterdam wil meer flexibiliteit voor zijn bewakers. Het bedrijf bewaakt woningen en bedrijven in en rond Rotterdam. Zodra er alarm gaat moet een medewerker die dienst heeft naar het object toe rijden om te gaan kijken of er wat aan de hand is. Meestal is het vals alarm en er een sprake van een storing of loopt er bijvoorbeeld een kat binnen.

Overdag zijn er altijd voldoende bewakers om rond te rijden maar met name ’s nachts kan het druk worden en is het belangrijk dat er snel wordt gereageerd op een alarm.

Gedurende de nacht zijn bewakers soms even niet beschikbaar: ze hebben pauze, gaan naar de WC of zitten te eten. Een bewaker mag in totaal gedurende zijn dienst zich 45 minuten afmelden en daarmee aangeven dat hij niet beschikbaar is. Als een bewaker zich afmeld dan kijkt het computersysteem of de afmelding mogelijk is. Het kan zijn dat er teveel medewerkers gelijktijdig pauze nemen en het verzoek kan dan worden afgewezen. Het systeem stelt een alternatieve tijd voor om de pauze op te nemen.

Het huidig ‘pauze-systeem’ werkt via internet en is traag en moet regelmatig worden gereset. SecureHomes wil een nieuw systeem laten ontwerpen.

Bedenk of je vragen hebt en stel deze tijdens de les aan de 'klant'.

Leg uit aan welke maatregelen jij denkt om dit nieuwe systeem veilig te maken. Noem de in jouw ogen drie belangrijkste maatregelen die jij voorstelt om dit systeem veilig te maken. Ga uit van de risico’s, waar liggen die, wat is belangrijk voor deze applicatie en met welke maatregelen zorg je ervoor dat de belangrijkste risico’s worden beperkt.

Jou verslag ziet er dus als volt uit:

Heel belangrijk

Risico 1: leg uit wat jij als belangrijkste risico ziet

Maatregel 1: leg uit wat jij hier tegen zou kunnen doen (1 of 2 maatregelen)

Belangrijk

Risico 2: leg uit wat jij als iets minder belangrijk risico ziet

Maatregel 2: leg uit wat jij hier tegen zou kunnen doen (1 of 2 maatregelen)

Aan te bevelen

Risico 3: leg uit wat jij als risico ziet

Maatregel 3: leg uit wat jij hier tegen zou kunnen doen (1 of 2 maatregelen)

Risiso's en Maatregelen

Risico's

1. De beveiligers werken via internet dat onveilig is en traag.
2. Te weining bewakers beschikbaar.
3. Servers gaan down
4. Internet gaat down of is erg traag
5. Pause is te lang
6. Hackers komen in het systeem
7. Ggevens worden gestolen door hacker.
8. Systeem raakt bespmet door virus
9. Devices van bewakers worden gestolen of verloren
10. Iedereen wil gelijktijding pause
11. Valse alarmmeldingen
12. Hackers zien roosters en gebruiken die om inbraak te plannen
13. Systeem wordt gehacked en alle pauses worden goedgekeurd
14. Logins worden gehacked en iedereen kan een bewaker op pause aanmelden
15. Geen wifi/4G brereik (bijv. in gebouw) en bewaker kan zich niet afmelden
16. Het hele systeem wordt gehacked en het hele systeem wordt verwijderd
17. Hackers uit het buitenland voeren een DDOS uit.
18. Bewakers krijgen phissing email waarmee ze hun password kunnen kwijtraken.
19. Bewaker is password vergeten

Maatregelen

1. LAN aanleggen en op die manier afschermen van internet
2. Leg SMS systeem aan als back-up (voor als servers niet beschikbaar zijn)
3. Samen werken met andere bedrijven om personeelstekort op te vangen

4. Eigen servers gaan gebruiken.
5. Pauses minder lang maken
6. Betere WiFi
7. Dataverkeer moet encypted zijn
8. Dataverkeer moet in 'geheim-taal' 04:00 uur is bijvoorbeeld 03:00 uur
9. Firewall installeren
10. Goede wachtwoorden gebruiken
11. anti virusbescherming
12. Lock out en wipe out voo rmobiele devices instellen
13. Geef aan als er teveel mensen met pause dreigen te gaan
14. Koop goede/snelle computers
15. Houd software up-to-date
16. Controleer of je web site geen scripts accepteerd via de invoervelden.
17.

Case: Efteling, vrijkaartjes winnen

De Efteling, een attricatiepark in het zuiden van Nederland wil graag een actie starten. Zij willen een website waar je mooie foto's van je bezoek van het atractiepark kunt oploaden. Je kunt je registreren met een code die je op je toegangskaart kan vinden. Als je dan bent geregsitreerd dan kun je per persoon maximaal 3 foto's opsturen. Elke dag worden de 10 mooiste foto's geselecteerd en elke winnaar krijgt dan via de site een nieuwe vrije toegangskaart via email toegestuurd.

Jij bent gevraagd om aan te geven wat alle risico's zijn. Maak een risico analyse waarbij je van elk risico aangeeft wat de kans is (1,2,3: laag, middel-groot, hoog) en de impact (1,2,3: laag, hoog, zeer hoog) per risico's zijn. Bepaal ook de maatregelen die je per risico kunt nemen, maak per maatregel ene inschatting van de kosten (1,2,3: groot, middel, laag).

Uiteindelijk moet je een tabel inleveren die er als volgt uit ziet:

Nr.	Risico	Kans	Impact	Maatregel	Kosten
1	Password gehacked	?	?	multi factor authenticatie	?
2					

Beschrijf eronder per nummer de details en vermeld of jij de maatregel aanbeveeld, dus bijvoorbeeld:

1. De kans op het hacken van password is niet erg groot want..... de maatregel die we kunnen nemen is MFA maar dit gaat ten kosten van de gebruikersvriendelijkhdeid. Wij raden het daarom niet aan.
2. ...

Maak hiervan een docuement en stuur dit op via Teams

Case FinTech BV.

Omschrijving

FinTech is een financieel handelsbedrijf en maakt ongeveer 100 miljoen winst per jaar (400 000 per dag). Het bedrijf heeft 150 traders en die hebben elk 2 pc’s met Windows 7. Verder heeft het bedrijg nog 300 Windows Servers, 1400 Unix Servers en 800 pc’s. Het bedrijf heeft 300 IT-ers en nog 50 andere medewekers in dienst.

- 100 miljoen winst per jaar / 500 medewerkers
- 150 traders met Windows 7 machine's
- 2000+ andere PC's

> Zet de **opties a,b en c** op volgorde van prioriteiten. Welke optie zou jij eerst uitvoeren en welke optie daarna?

Optie A

FinTech heeft oude Trader PC’s met Windows 7. Windows 7 heeft vulnerabilities en wordt niet meer door Microsoft gepatched. Windows 7 is dus kwetsbaar.
De Trader PC’s zijn niet verbonden met internet en zijn fysiek afgeschermd.

Het upgraden van alle PC’s naar Windows 10 kost 10 miljoen.

Optie B

FinTech BV heeft een paar 100 Office PC’s (Office/email) en die zijn verbonden met het internet. Deze pc’s worden gepatched en draaien op Windows 10. Via mail komen er soms phishing verzoeken binnen.
Via extra email software kan al het netwerkverkeer worden gecontroleerd.

Kosten zijn 40 000 eenmalig en 6 000 per jaar voor software onderhoud.

Optie C

FinTech BV verdient 100 miljoen per jaar.
Er zijn bepaalde uitzonderlijke (3-10 per jaar) dagen dan er 5 tot 10 miljoen per dag kan worden verdient. Als de stroom op deze dagen uit zou vallen dan is niet alleen deze winst niet verdient het bedrijf kan ook nog eens een extra verlies maken van 5 tot 10 miljoen maken. Dus als de stroom op deze uitzonderlijke dagen uitvalt dan kan de winst voor een heel jaar ‘verdampen’.
De kans op stroomuitval is echter zeer laag (1x per 10 jaar).

Kosten 10 miljoen eenmalig en 100 000 onderhoud per jaar.

Nikto test

We gaan testen of onze webserver veilig is met de Nikto test.

(Docs zie: https://hackertarget.com/nikto-tutorial)

Installatie

Indien je geen Ubunto onder Windows hebt, installeer dit dan via de Windows App store.

Open Ubuntu onder Windows en ga naar: /mnt/c/Users/<home>/ en download de nikto suite.

```
wget https://github.com/sullo/nikto/archive/master.zip
```

Unzip de file onder windows en ga daarna onder Ubuntu naar de juiste directory.

Ga naar: mnt/c/Users/<home>/master/nikto-master/program en type perl nikto.pl.

Als het goed is zie je een help file. Als dat zo is dan kunnen we de scan beginnen.

Start de Nikto test: https://cirt.net/nikto2-docs/usage.html#id2780332

Rapportage

Maak een kort verslag van de site die je hebt onderzocht en vertel in je eigen woorden wat je hebt gevonden. Denk daarbij aan de OWASP punten. Geef zo praktishce mogelijk advies aan de hand van de resultaten.

Create Safe Code - SQL Injection

We gaan een eenvoudige login maken en we gaan zien hoe je een eenvoudige login kunt omzeilen. Hiervoor gaan we SQL injection gebruiken. De opdracht is om deze login te maken en dan zodanig aan te passen dat de login veilig is en niet meer kan wordnen omzeild.

Maak een database met usernaem en wachtwoord en maak een eenvoudige login.

De database kan worden aangemaakt met script dat je hier kunt downloaden.

form.html

Maak een html pagina met een login form. Het form begint met:

`<form class="" action="login.php" method="post">` en heeft twee invoer velden een user en wachtwoord.

Maak het form verder af.

Database

De php code is hieronder toegevoegd. De login maakt gebruik van een een include. De code van de include waarin een verbinding met de database wordt gemaakt staat hieronder ook. let op dat je nog wel de database parameters moet invullen.

login.php

```
<?php


include "includes/db.php";


echo "User:".$_POST['user']."<br>";
echo "Wachtwoord:".$_POST['wachtwoord']."<br>";


$user=$_POST['user'];
$wachtwoord=$_POST['wachtwoord'];


$myConn = new DB;
```

```
$query = "SELECT * FROM login where username='$user' and wachtwoord='$wachtwoord'";
$result = $myConn->executeSQL($query);

if ( $result != 0 ) { // voor deze opdracht moet je deze regel niet veranderen
    echo "<br> Login as $user <br>";
} else {
    echo "<br> Invalid login! <br>";
}

?>
```

db.php

```
<?php

// Define DB Params
define("DB_HOST", "localhost");
define("DB_USER", "");
define("DB_PASS", "");
define("DB_NAME", "login");

class DB{
protected $dbh;
protected $stmt;
    protected $resultSet;

public function __construct(){
$this->dbh = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME, DB_USER, DB_PASS);
        $this->resultSet=[];
}

public function executeSQL($query){
        $this->stmt = $this->dbh->prepare($query);
$result = $this->stmt->execute();
        if (! $result) {
           die('<pre>Oops, Error execute query '.$query.'</pre><br><pre>'.'Result code:
'.$result.'</pre>');
       }
 $this->resultSet = $this->stmt->fetchAll(PDO::FETCH_ASSOC);
       return count($this->resultSet);
```

```
}

    public function getRow(){
if (count($this->resultSet) >0 ){
return array_shift($this->resultSet);
        } else {
return 0;
        }
    }



}

$DB = new DB;

?>
```

Test of de login werkt.

Voer dan als wachtwoord `' OR '1'='1` in. Kijk wat er gebeurt en en leg uit wat er gebeurt. Hoer heet deze techniek?

Opdracht 1 (P1W1)

Maak de code zodanig veilig dat het truuckje wat hier boven is beschreven niet meer werkt.

Opdracht 2 (P2W2)

1. Breidt de code uit zodat je via de browser een gebruiker en wachtwoord kan toevoegen in de database.
2. Pas de code die je zojuist hebt gemaakt aan zodat de wachtwoorden encypted worden opgeslagen.
3. Pas de code verder zodanig aan dat de controle op het invoeren van een juist wachtwoord plaatsvind op encrypted wachtwoorden.

Opdracht 3 (P2W2)

Voer in de login bij username het volgende in:

```
<script>alert('you are hacked!')</script>
```

DIt sctipt is verder niet gevaarlijk maar je kunt je voorstellen dat je op deze manier wel een script kunt 'injecteren' dat gevaarlijk is. Pas de code aan zodat je beveiligd bent tegen dit soort aanvallen.

Opdracht 4 (P2W2)

Kun je een algemene beschrijving maken van hoe je er voor zorgt dat je via een form (user input) de kans op hacken zo klein mogelijk maakt. Maak deze beschrijving in één of een paar regels.

--

PEN test 1 - CentOS - SSH

Een PEN test is een penetratietest. Een test om te kijken en te controleren of je ergens binnen kunt komen. Dat kan een file server zijn maar dat kan ook een web server zijn. We zelf een PEN test uitvoeren op onze eigen web site en we gaan de website daarna veiliger maken. In deze les gaan we een server inrichten als voorbereiding op de PEN test.

Het uitvoeren van de PEN deel van het examen Veilig programmeren.

Overzicht

Testen op XAMP heeft weinig zin, omdat dat een afgeschermde development omgeving is en een development omgeving is niet veilig (zie opgave 1).

We gaan dus een productieomgeving inrichten daarvoor gebruiken we VMWare. Dit lijkt heel erg op een echte productie machine. Als je nog van plan bent om zelf een productieserver in te richten, let dan goed op want de stappen die we gaan nemen zijn vrijwel hetzelfde bij het inrichten van een VM.

Via VMWare gaan we een CentOS (7) Linux machine inrichten. Er zijn meerdere Linux distributies, maar CentOS is gratis en wordt veel gebruikt als (web)server.

Nadat we een Linux server hebben, gaan we deze installeren en inrichten. Hiervoor zul je ene boel moeten uitzoeken op het internet. In deze les worden alleen de stappen beschreven; je moet zelf uitzoeken en overleggen hoe het allemaal precies werkt.

Als de server is ingericht met **MariaDB**, **PHP7.x** en een **Apache** webserver dan gaan we onze eigen web applicaties installeren. Het beste is als we een PHP voorbeeld en Laravel voorbeeld kunnen gebruiken. Dan kunnen we de verschillen zien.

Met een zogenaamde Nikto scan gaan we de test uitvoeren.

Installation

Install VMWare Workstation 15.

(https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html)

New VM

Download CentOS DVD: http://isoredirect.centos.org/centos/7/isos/x86_64/

(ik heb deze lijnk gekozen: http://centos.mirror.triple-it.nl/7.7.1908/isos/x86_64/CentOS-7-x86_64-DVD-1908.iso)

Install CentOS

Start VMWare PLayer op en start en install van de ISO die je hebt gedonwload.

- Extra VMWare tools hoef je neit te downloaden als daar om wordt gevraagd.
- Taal English (Ireland)
- System Installation Destionation aangeven (op nieuwe VMWare Schijf).
- Network & Host Name selecteren en Ethernet (rechtsboven) aan zetten!
- Begin Installation
- Root password instellen (niet vergeten; opschrijven!)
- Geen user aanmaken (tijdens installatie)
- Reboot

Done

Install Software

(based on tutorials from https://www.howtoforge.com)

Eerst moeten we een repository toevoegen (dat is een soort database met software):

```
yum -y install epel-release
```

Dan gaan we een editor installeren omdat de meeste de standaard vi editor te lastig vinden.

```
yum -y install nano
```

Dan gaan we MariaDB Installeren

```
yum -y install mariadb-server mariadb
```

Datbase opstarten en zorgen dat die bij een reboot weer automatisch wordt gerestart.

```
systemctl start mariadb.service
systemctl enable mariadb.service
```

Nu moeten we een root wachtwoord voor de SQL server instellen.

```
mysql_secure_installation
```

Install Apache Webserver.

```
yum -y install httpd
```

Start de webserver en zorg dat die bij een reboot weer automatisch wordt gerestart

```
systemctl start httpd.service
systemctl enable httpd.service
```

Zet een hostname voor je Appache Server

```
nano /etc/httpd/conf/httpd.conf
```

En plaats deze regel. De regel staat al in de file maar er staat een # voor waardoor de regel in commentaar staat. Haal dat # weg.

Zet firewall for http en https open.

```
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --reload
```

Check wat je ip address is (zoek zelf even uit hoe) en controleer of je browser de standaard Apache web pagina kan vinden op je nieuwe server.

Nu gaan we PHP installeren, eerst de juiste repo toevoegen, alle software updaten en de installer (YUM) updaten.

```
rpm -Uvh http://rpms.remirepo.net/enterprise/remi-release-7.rpm
yum -y install yum-utils
yum update
```

Install PHP 7.3

```
yum-config-manager --enable remi-php73
yum -y install php php-opcache
```

En restart de web server.

```
service httpd restart
```

De docuement root is `/var/www/html`

Zet daar een klein PHP scripje neer om te controleren of de PHP engine werkt.

Done!

Website maken

Een (niet-Laravel) web site kan je nu maken door de in de document root directory te maken en daar in een web site te plaatsen. Gebruik een van de websites die we eerder hadden gemaakt en cotnroleer of die werkt.

Voor het aanmaken van een Laravel website is het handig dat we grote heoveelheden files tussen onze Laptop en de VM kunnen kopieren, dat doen we later.

Installeer MobaXterm (of Putty)

We hebben tot nu toe rechtstreeks op de VM gewerkt Dit is alsof we in een datacentrum staan en rechstreeks op onze server werken. Zo werkt het bij een niet-VM natuurlijk niet. We moeten dan remote aanloggen en dat doen we met een SSH client. Download MobaXterm en maak via SSH een verbinding met he VM die in de achtergrond natuurlijk wel draait.

Omgeving veilig(er) maken

Er is een aantal zaken die we kunnen regelen om onze omgeving veiliger te maken:

- Als je via een SSH client aanlogt op je (VM) server dan moet je een password gebruiken. Enerzijds wil je je password heel lang maken en anderzijds moet je het telkens intypen en wil je het niet te lang maken. Om dit te voorkomen kun je SSH keys opzetten. Dit werkt via PPK (Public Key Infra Structuur). Weet je nog van dat hangslotje hoe dat werkt? We gaan in MobaXterm een SSH key pair maken. Je plaatst je public key op de server en houd je private key op je laptop. Het opzetten van SSH keys tussen een Windows machine en een LInux machien is redelijk complex en daarom gaan we dit in de les samen doen.
  Zonder deze stap kun je wel gewoon door met de rest van deze les.

- Eingenlijk zou je nooit met root (Super User - mag alles!) moeten aanloggen. Sterker: je kan je server zo instellen dat je nooit rechtstreeks met root kan inloggen, maar alleen via een ander account. Het is een goed gebruik om root niet of alleen in noodgevallen te genbruiken. We moeten dus een eigen user voro onszelf maken.

- Zoek nu op hoe je in de `/etc/sudoers` file je nieuwe gebruiker de rechten kan geven om zichzelf super user te maken. Op die manier kun jij ook alles. Als je iets niet kan dan typ je er sudo (Super User Do) voor en hoppa, je bent voor dat ene commando root.

- In de webserver directory gaan we de rechten nu zo instellen dat we met onze 'gewone' user files kunnen plaatsen en kunnen aanpassen. Natuurlijk moet je hier *max* vervangen in de user die je zelff hebt aangemaakt.

```
sudo chown max:apache /var/www/html/
```

- SSH gaat standaard over poort 22. Als je een domein hebt dan duurt het vaak niet langer dan een paar weken en dan zijn er 1000+ hackers die proberen een SSH verbinding op te zetten naar jowu server. In de VM omgeving heb je hier natuurlijk geen last van. Door SSH over een 'geheime' poort plaatst te laten vinden houd je een boel gelegenheids hackers buiten de deur. Zoek uit hoe dat werkt en stel dat in op je VM. Zorg er dan ook voor dat je SSH client ook via de andere alternatieve poort gaat (dit is opgave 6).

Opgaven

1. Zorg ervoor dat je webserver met PHP draait op je VM. Hierboven staat beschreven hoe je dat moet doen.
   Maak een schermafdruk waarop je URL te zien is en voer een PHP scripje uit dat jouw naam op het scherm zet (in de brwoser).

2. Noem een aantal zaken waar je aan zou moeten denken als je een web site vanuit jouw XAMPP development omgeving in productie zou willen zetten. Wat moet je veranderen/aanpassen (denk aan alles wat we tot nu gedaan hebben)?

3. Noem twee goede redenen waarom je zeker in een omgeving met meerdere system managers het root account nooit (rechtstreeks) wilt gebruiken.

4. Waar staat SSH voor?

5. Waarom heb je in een VM omgeving geen last van hackers die proberen via poort 22 een SSH verbinding op te zetten?

6. Zorg ervoor dat het SSH verkeer over een andere dan poort 22 plaats vind. Zoek zelf op internet uit hoe dat werkt?

7. Hoe zou jij als heacker proberen binnen te komen via SSH; wat zou je doen/proberen?

8. Stel iemand hacked jouw (VM) server en steelt jouw key uit de file `authorized_keys`. Wat kan de hacker nu? Kan die in alle servers komen waar jij via dit key/pair toegang tot hebt? Leg uit hoe en waarom.

9. Met het commando `sudo su` maak je jezelf root en hoef je niet telkens `sudo` in te typen. Waarom zou dit een slechte gwoonte zijn?

10. Beschrijf van de volgende Linux commando's wat ze doen en test ze uit. Geef telkens een voorbeeld.

Commando	Voorbeeld	Uitleg
cd		
mv		
rmdir		
rm		
mkdir		
ll (of ls-la)		
history	niet nodig	
exit	niet nodig	

--

ChatGPT

Knight Capital

Knight Capital was een succesvol bedrijf datzijn beld verdiende op de beurs door the handelen in financiële products zoals aandelen.

In 2011 had dit bedrijf bijna 1500 manin dienst en were er een winst van 115 miljoen US dollar gemaakt. Dat is 750 000 per werknemer. Ter vegelijking Albert Heijn maakte in 2011 2.2 miljard euro winst en had 413 000 mensen in dienst. Dat is 5300 per werknemer. En 2021 was een heel goed jaar voor AH.

In 2012 gebeurde er iets met 2012.

Opdracht 1

Zoek op het internet op wat er gebeurde met Knight Capital. Noem zo veel mogelijk feiten.

Opdracht 2

Zoek op wat de oorzaak was van dit vervelende voorval.

Chat GPT

In 2023 were ChatGPT 3.5 en 4.0 gelanceerd.

Opdracht 1

Wat kun je allemaal doen met ChatGPT?

Opdracht 2

Wat zijn de voordelen van het gebruik van ChatGPT?

Wat zouden de nadelen vannetgebruik vanChatGPT kunnen zijn?

Eindopdracht

Wat is de relative tussen Knight Capital enChatGPT?

Conclussies

Fouten zijn normal. MS Word 30 000 per maand.

Ook $$$ programeurs maken fouten.

Testen, testen en testen.

Maak nette code!

Gebruik geen code die je niet begrijpt.