Symmetrische Cryptografie

Voorbeelden

DES, AES

Wat is symmetrische cryptografie

Dezelfde sleutel wordt gebruikt om te encrypteren en te decrypteren

Basis werking

Substitutie: X vervangen door Y

Transpositie: X op een andere plaats zetten

Bij DES 16 iteraties 🡪 AKA Number Crunchers

Voordeel

Snel

Nadeel

Sleutelmanagement:

* Sleutel veilig van A naar B krijgen
* hoe meer personen, hoe meer keys je nodig hebt

Avalanche effect

Doordat DES werkt met 16 iteraties, waardoor een kleine fout steeds groter en groter wordt

DES encryptie

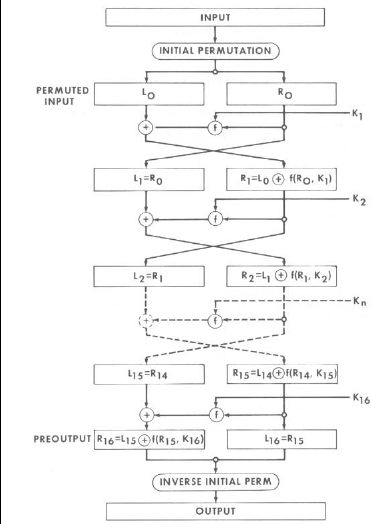
Is een blok cipher: je werkt altijd met blokken van data (bij DES blokken van 64 bit)

Doordat het een symmetrische cryptografie is werk je met substitutie of transpositie (16 iteraties)

Je krijgt een datafile die je opsplitst in stukken van 64 bit

1 stuk van 64 bit stuur je door uw DES encryptie

Eerste stap: door een permutatie matrix sturen (p)

* Permutatiematrix verplaatst de bits door elkaar zodanig dat je 32 Bits vanboven en 32 Bits vanonder krijgt

Tweede stap: 16 iteraties van

* L1 = vorige R
* R1 = vorige L geXOR’d met de uitkomst een functie (feistal functie) f(vorige R, K1) 🡪 K1 = subkey
* Bij laatste iteratie wordt er niet meer geswitched van kant

Laatste stap: terug samenvoegen door het door een permutatiematrix te voegen

Dit doe je voor elk blokske van 64 bits van uw originele datafile

Feistal functie (R0, K1)

R0 = 32 bits

K1 = 48 bits

Deze voeg je samen door een XOR functie, maar daarvoor moeten ze evenveel bits hebben

* Je stuurt R0 door een expansie matrix (bits verplaatsen, en sommige gekopieerd)

De uitkomst van de XOR spreid je uit over 8 S-boxen (elke box 6 bits)

Eigenschap van de S-box is dat hij 6 bits als input heeft, en 4 bits als output

S-boxen terug samenvoegen door een permutatie matrix = 32 bits uitkomst

De S-box is de kracht van de DES functie

Werking van de S-box

Een tabel met 4 rijen en 16 kolommen

Tabel is gevuld met getallen tussen 0 en 15

Werking (vb 011110)

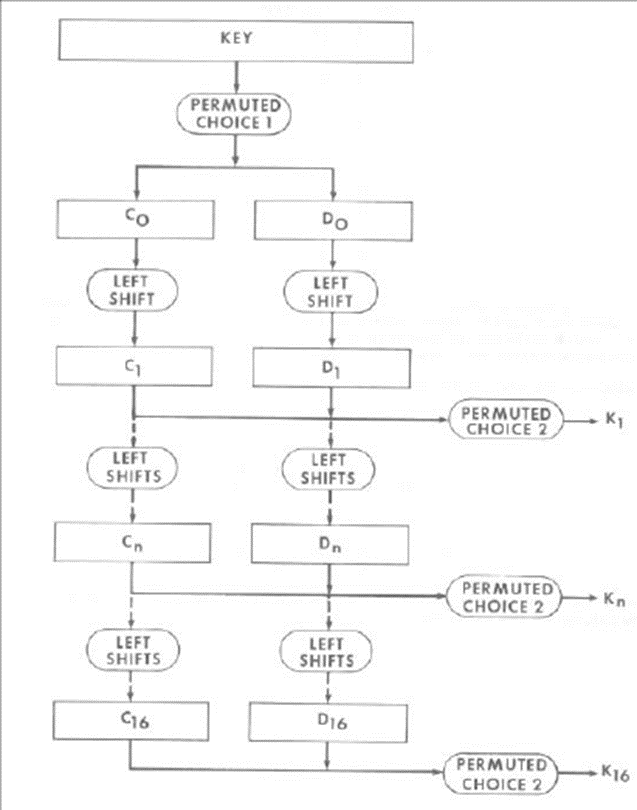
Je neemt de eerste bit en laatste bit van uw 6bit invoer samen (= 00)  
dit bepaald in welke rij je zit (dit is rij 0)

De 4 middelste getallekes bepalen in welke kolom je zit (1111 = kolom 15)  
Die rij, deze kolom = dat getal uit de s-box

Dat getal is een 4 bit getal en dat is uw output

Dit wordt voor alle 8 S-boxen gedaan

Subkey berekenen

DES werkt altijd met blokken van 64 bit

Uwe masterkey is een 64 bit paswoord

Als DES een matrix toepast bij het berekenen van keys zal hij altijd 8 bits negeren/wegsmijten

De 64 bit van de masterkey wordt door een permutatiematrix gegooid, maar 8 bits worden weggesmeten

Dan hou je 56 bits over, deze worden terug gesplitst en krijg je 28 bits links en 28 bits rechts

Hier worden weer 16 iteraties op gevoerd om uw 16 subkeys te bekomen

* De iteratie is een left shift: 01101 wordt 11010 wordt 10101 (maar dan met 28 bits)
* Permutatiematrix, maar het is keyberekening dus smijt hij er 8 weg = 28+28-8=48 is subkey

Left shift kan 1 of 2 bits opschuiven

Bij key berekening van DES keys bestaat zoiets als weak keys

Stel dat alle bits 0’en of 1’nen waren, dan heeft een left shift niet veel zin, end eze moge dus niet gebruikt worden door DES

Decryptie

Zelfde methode, alleen start je met K16 (omgekeerde subkey volgorde)

3DES

Triple DES, 3keer na elkaar DES doen met verschillende sleutels: SIM kaarten op telefoons, encryptie van routerverkeer tussen routers

Asymmetrische cryptografie (public key)

Ontwikkeld om het sleutelmanagement probleemn van sym crypto op te vangen

Werkt met 2 keys, een public en een private

Deze hebben een wiskundig verband, maar ge kunt de 1ne niet van de andere afleiden

Key generatie bijna altijd via priemgetallen en modulus berekening

Basis werking

Basis berekeningen

* Encrypteren: zorgen dat encryptie van persoon A, persoon B de enige is die kan decrypteren
  + Als A een bericht naar B wilt sturen, gebruikt A de public key van B, zo is hij zeker dat B de enige is die het kan decrypteren (met zijn eigen private key)
* Authenticatie: persoon A heeft file geauthentiseerd, dan is B zeker dat het van A afkomstig is
  + A versleuteld bericht met zijn private key. Als je dit kan decrypteren met de public key van A, weet je zeker dat het van A komt

4 principes van asymmetrische cryptografie

* Ontbinden van grote getallen (factorisatie) 🡪 RSA
* Berekenen van discrete logaritmen 🡪 Diffie-Helman
* Knapzaksysteem 🡪 mag niet meer gebruikt worden, zit wiskundige zwakheid in
* Elliptische curven

Ontbinden van grote getallen

Factorisatie: 21 = 3 x 7

Factorisatie probleem: 21 is gegeven, zoek 7 en 3

Cryptografie maakt gebruik van priemgetallen

N = P x Q, N is gegeven, zoek p en q, met p en q een groot priemgetal

Berekenen van discrete logaritmen

Kofferke sturen van A naar B. A hangt slot op koffer en stuurt naar B, B hangt daar zijn slot op en stuurt terug naar A. A haalt zijn slot eraf en stuurt naar B. B haalt zijn slot eraf en koffer kan open.  
Gevoelig voor man in the middle attack

* Daarom best authenticatie methode zoals digitale handtekening gebruiken

Knapzak

Verzameling objecten met een gewicht en waarde

Vul een rugzak van 5kg met zoveel mogelijk items met een zo hoog mogelijke waarde

Niet meer gebruikt wegens wiskundige zwakheid waardoor je private key van public key kan afleiden

Hashing

De belangrijkste hashing algoritmes zijn: MD5 en SHA-1, SHA-2, …

Een hash is een zeer snel algoritme dat een getal terug geeft van een vaste grootte

Wordt vooral gebruikt als fingerprint van een file 🡪 als er 1 bit veranderd aan het bestand, is heel uw hash anders. Het waarborgt dus de integriteit van het bestand

Belangrijkste eigenschap

* pure 1-way function
* Het is heel makkelijk om de hash te berekenen, maar onmogelijk om het origineel te herberekenen
* Collision Free: Moeilijk om 2 keer dezelfde hash te krijgen uit een bestand

Gebruik

Paswoorden

* In een databank wordt de username en de hash bijgehouden
* Van u ingegeven paswoord maakt hij en hash, en controlleert met de hash in de databank
* Komen deze overeen ben je ingelogd

Aanval tegen hashing 🡪 Birthday Attack

Als iemand de databank bemachtigd met de usernames & hatches

* Hackers hebben dictionary met veelgebruikte paswoorden met bijhorende hashes
* Ze gaan in de inlogdatabank kijken of er een match is tussen hashes uit de 2 bestanden

Rainbow table

Een zo optimaal bijgehouden dictionary met veelgebruikte paswoorden

Beveiliging tegen birthday attack

Een salt wordt toegevoegd aan de databank (per user). De hash die nu wordt opgeslagen in de databank is de hash van de combinatie van het ingevoerde wachtwoord met de salt.  
Gevolg voor de hacker: hij moet nu per gebruiker, de salt nemen, achter zijn dictionary met paswoorden zetten, hash herberekenen en gaan vergelijken. En dit per gebruiker in de gehacke databank, héél tijdrovend

Digitale handtekening + encryptie

Je encrypteert het bericht met je private key. Dit geheel encrypteer je met de public key van de ontvanger.  
De ontvanger gebruikt zijn private key om het te decrypteren, en daarna de public van de verzender

Digitale handtekening + hash

RSA was traag.

Bij grote bestanden (bv 1 terra)

* Maak een hash van uw groot bestand, dit geeft een klein bestand
* Encrypteer deze hash met RSA met de private K van A
* Verstuur bestand + genrypteerde hash
* Ontvanger maakt een hash van het verzonden bestand en vergelijkt het met de meegestuurde hash, die hij decrypteerd met de public key van de ontvanger

Hybride cryptografie

Combineert de voordelen van symmetrische, asymmetrische en hashing technieken

1 terra file van A encrypteren met DES sleutel (= klein bestandje)

Die DES key haal je door RSA, met de public key van B =gencrypteerde DES sleutel

Digitale handtekening om te weten dat het echt van A komt: 1 terra file door hash halen = hash getal.  
Dit hash getal haal je door RSA met de private key van A.  
Die 3 files stuur je naar B.

Des sleutel decrypteren met private sleutel van B, geeft de DES sleutel

1 terra file decrypteren met de net gedecrypteerde DES sleutel

Hash decryptern met public van A  
Zelf hash maken van 1 terra file en hashes vergelijken

1 probleem bij hybride cryptografie:

De public key van A, en de persoon A is geen koppeling tussen. Iedereen kan dus beweren dat hij A is. Dit wordt tegengegaan door certificaten

Public Key Infrastructure (PKI) (Certificaten)

Oplossen van de security flaw van koppeling tussen public key en de persoon zelf.

Wat is PKI:

* Technologie: public key technologie gebruikt voor encryptie, samen met digitale certificaten
* Infrastructure: certification authorities en security management
* Services: vertrouwelijkheid, authenticiteit, integriteit en onweerlegbaarheid

Digitale certificaten: verbindt een publieke sleutel aan een individu of organisatie

Wordt vertrouwd omdat het wordt uitgegeven door een Certification Authority (CA)

Inhoud certificaat: public key, id, extra gegevens

Deze CA heeft dat certificaat gesigneerd (digitale handtekening met private key) en garandeerd hiermee dat een certificaat bij een bepaalde persoon hoort

Soorten certificaten

* Persoonsgebonden certificaat: gekoppeld aan 1 persoon
* Rolgebonden certificaatl: rol die een persoon vervult binnen een organisatie
* Toepassingsgebonden certificaat: applicaties die zelf gegevens versturen, gaan deze ondertekenen, bv een server die log-info iedere dag doorstuurt naar een aantal admins
* SSL-servercertificaat:
  + Beveiligd verbinding tussen client en server
  + Identificatie van het IP-adres + domain name van de server
  + Gebruiker zit op de juiste website en gegevens die hij invult kunnen niet door derden worden afgeluisterd
* Signing certificaat: ondertekening van software: code afkomstig van ondertekenaar

PKI = hele systeem van bedrijven die certificaten maken

Componenten van een PKI

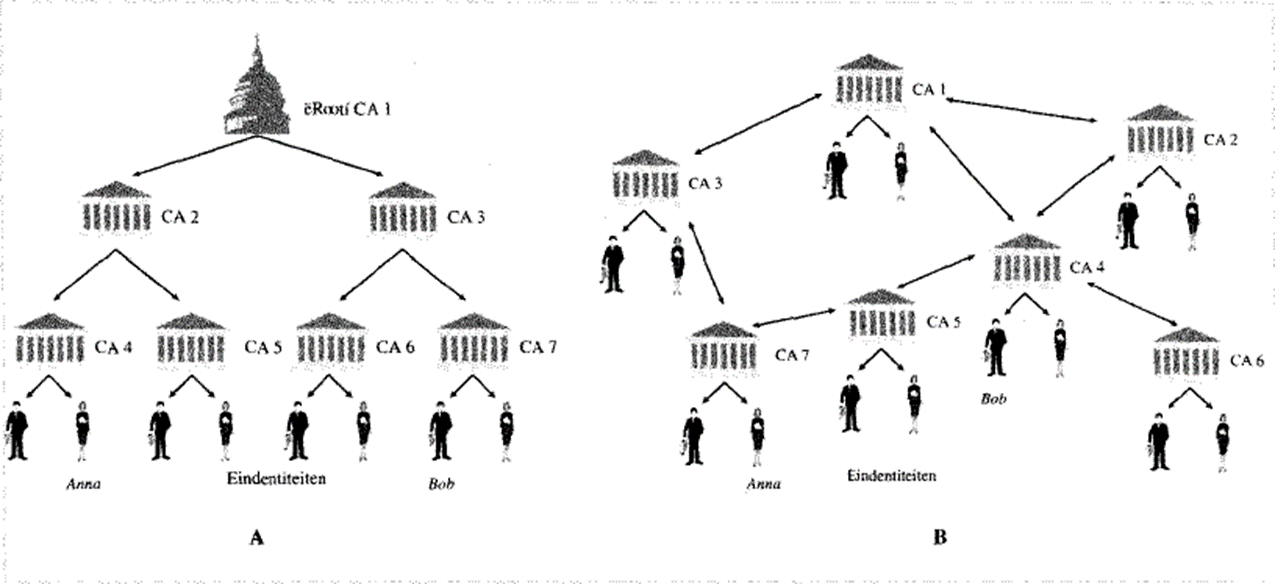
* Certification Authority (CA)
  + Generatie van certificaten
  + Uitgave en distributie
  + Intrekken van certificaten
* Registration Authority (RA)
  + Registratie van gebruikers
  + Vaststellen identiteit van gebruikers (Proof of Posession)
* Validation Authority (VA)
  + Verifieren of certificaat al dan niet is ingetrokken of opgeschort
* Repository/directory
  + Opslag van certificaten
  + Opslag black list (CRL = Certificate Revocation List)
* Eindentiteiten
  + Aanvragen van certificaat
  + Aanmaken van private en public keys
* Certificate Policies en Certificate Practice Statement
  + Organisatorische en juridische richtlijnen

Certificate Policy:

* Regels en normen die aan CA wordt gesteld
* Beleidsniveau
* Gestandaardiseerd document: hoe willen ze uitgifte aanpakken, certificaat formaat – CRL formaat, contractuele bepalingen

Certificate Practice Statement

* De door de CA gevolgde procedures om aan die regels en normen te voldoen
* Hoe ze de CP concreet invullen, welke methodes gebruiken ze van encryptie



PKI-architectuur

2 soorten:

Hierarchische architectuur:

* Veel toegepast
* Iedereen kent public key van root CA
* Root CA:
  + Signieert CA’s van onderliggende level, die dan signeren voor level daaronder, ….
* Public key van CA juist boven eindlevel wordt gebruikt om validiteit te controleren, levelup controleren, … tot root-CA
* Voordeel: duidelijkheid in opbouw en vertrouwens-relatie
* Nadeel: compromittering in structuur heeft grote gevolgen (vertrouwen tussen 2 CA’s valt weg   
  🡪 hele structuur valt in elkaar

Mesh architectuur

* Alle CA’s hebben gelijkwaardig erelaties met elkaar
* Geen root of hierarchie
* Iedereen kent public key van locale CA waar hij bij aangesloten is
* Verificatie kan langs verschillende paden
* Cross-certificering
* Voordeel: flexibiliteit, uitvallen van CA is geen probleem
* Nadeel: niet duidelijk hoe vertrouwensrelatie juist is

Cross-certificering

Verderzijdse erkenning van 2 CA’s in een verschillende omgeving (zij vertrouwen elkaar)

Op basis van CPS (certificate practice statement)

Certificaat aanvragen

* A geeft haar public key + persoonlijke gegevens aan RA 🡪 controle 🡪 ok? Volgende stap
* Certificaat voor A wordt aangevraagd bij CA
* CA keurt dit goed, creatie + ondertekening certificaat
* Terugsturen naar A + toevoeging aan public db
* B is ook gebruiker 🡪 zoekt certificaat op in db
* B controleert certificaat met public key CA om te kijken of het echt van CA komt
* Heeft nu public key van A en is zeker dat het van A is

Intrekken van certificaat

Wanneer?

* Vermoeden dat certificaat niet meer integer is
* Private key gecompromitteerd
* Geldigheidsduur verlopen

Gevolg: certificaat uit public directory halen (revoke)

Certificate Revocatoin List (CRL = Black List)

Intrekking wordt gepubliceerd door CA

Online Certicate Status Protocol (OCSP)

Online realtime controle van certificaat (freshness)

B vraag statusinfo aan OCSP server (naam CA, serienummer, versienummer)

De OCSP server staat in rechtstreeks contact met de CA

Verwerking door OCSP en stuurt terug:

* Good: niet op CRL
* Revoked: Op CRL
* Unknown: OCSP kent CA niet

Proof of possession

RA is zeker dat certificaat van public key hoort bij de aanvrager

Bewijzen dat private key in het bezig is van de aanvrager van het certificaat (met bijhorende pub key)

Key Escrow

VS ziet crypto als een militair aspect

Verbiedt uitvoer van sterke crypto systemen, enkel korte sleutels

Sterke crypto soms toegestaan, maar dan moet gebruiker zijn private key ook laten beheren door de CA

Zo kunnen ze de communicatie bekijken als ze een gerechtelijk bevel hebben zodat de CA de private key moet geven 🡪 nodig in de strijd tegen terrorisme

Zijn big brother toestanden

Nooit de private signeringskey uit handen geven!!!

Web security

Web verkeer beveiligen

3 methoden:

* IPSec: Voordelen
  + Transparant voor eindgebruikers en toepassingen
  + Universele oplossing
  + Filtermogelijkheid: alleen verkeer dat beveiligd moet worden heeft de ipsec overhead
  + netwerklaag
* SSL/TSL
  + Beveiliging direct bovenop TCP
  + De meeste browsers en webservers hebben SSL geimplementeerd
  + Applicatie moet zo ontworpen zijn om SSL/TLS te gebruiken
  + transportlaag
* toepassingsgebonden beveiligingsservices
  + service kan op maat van toepassing gemaakt worden
  + S/MIME , PGP
  + applicatielaag

Electronic Mail (toepassingsgebonden)

Beiden gebruiken het principe van hybrid cryptografie, enige verschil zijn de certificaten

S/MIME: Secure Multipurpose Internet Mail Extension

* Gebruikt PKI certificaten, dus uitgereikt door een CA
* Meestal gebruikt in web-based mailing systemen

PGP: Pretty Good Privacy

* Gebruikt zelf gemaakte certificaten die vertrouwd worden door een web of trust
  + Web of trust: als iemand een certificaat vertrouwt, gaat hij dit signen. Als er voldoende mensen (uit uw vertrouwenslijst) dit doen, vertrouw jij dit certificaat ook

SSL/TLS

2 lagen van protocollen

* Laag 1: Record protocol
  + Levert elementaire beveiligingsservices aan hogere lagen, bv aan http
  + Beveiligde data transfer
* Laag 2: wordt voor beheer van SSL uitwisseling gebruikt
  + SSL handshake protocol
  + SSL change Cipher Spec Protocol
  + SSL alert protocol

SSL Record protocol

* Fragmenteert data in hanteerbare blokken
* Comprimeert elk blokje
* Past H-Mac toe op het blokje (hash met mac secret key)
* Versleutelt dat blokje dan
* Voeg header toe met info over welke versleuteling

Change Cipher spec protocol

Geef gewoon aan dat de afspraken in orde zijn, vanaf dan is er beveiligde communicatie

Alert Protocol

Als er iets mis is gelopen, stuurt deze een alert

Handshake Protocol

4 fasen:

Fase 1:

* Client wilt veilig communiceren: stuurt het ‘Client Hello’ bericht
* Dit hello bericht bevat een aantal parameters:
  + SSL versie van de client
  + Sessie-ID
  + Lijst met vercijferingsalgoritmen die client kent
  + Lijst met compressiealgoritmen die client kent (zip + hash functies)
  + **Random getal van de client**
* Server stuurt ‘Server Hello’ bericht terug
* Met de paramaters
  + Ssl versie
  + Sessie id
  + Gekozen encryptie
  + Gekozen compressie
  + **Random getal van de server**

Fase 2: volledig serverside: server authenticatie

* Stuurt zijn certificaat naar de client (met zijn public key in)
  + Als server geen certificaat heeft 🡪 stuurt een server key exchange bericht
* Vraagt het certificaat van de client
* Server ‘hello done’ bericht

Tussenfase: client controlleert certificaat van de server via OCSP

Fase 3: clientside gegevens: client authenticatie

* Stuurt zijn certificaat (met public key van client)
  + Als hij geen heeft: no certificate alert, server beslist of communicatie verder kan of nie
* Certificate verification bericht (Proof of Posession)
  + Client maakt hash van alle berichten die tot nu toe verstuurd warden in de handshake
  + Versleuteld dit met zijn privé key
  + Server heeft berichten ook, maakt zijn eigen hash, en controlleert met client hash
* Client key exchange
  + Client maakt een 48 bytes pre-master secret
  + Dit versleutelt ge met de publieke sleutel van de server, en stuur naar server

Gevolg fase 3: client en server hebben nu beiden premaster secret

* Beiden berekenen uit premaster secret, de master secret (adhv random getallen fase 1)
* Bereken uit master secret o.a. de encryptie key (ook MAC key)

Fase4:

* Client sturt change cipher spec aan server en een finished bericht
* Server doet hetzelfde

Functie van de random getallen:

Tegen replay attack: aanvaller houdt alle berichten bij die gecommuniceerd worden met de server, en stuurt die op een ander tijdstip gewoonweg terug naar de server 🡪 dit gaat niet omdat er random waarden in het spel zijn en de server zal een ander random getal kiezen voor een andere sessie

Virtual Private Networks (VPN)

Eigen afgesloten netwerk dat gebruik maakt van het internet

* Remote access van medewerker tot het bedrijf
* E-commerce: gedeelte van het netwerk openstellen voor klanten
* Extranet: partners die mee kunnen inloggen om bv voorraad op te vlagen

Private: Beveiligd door allerlei mechanismen, dus privé

Virtual: geen directe lijn, maar over internet door gebruikt te maken van tunneling techniek = virtueel

Technologieen: PPTP (point to point tunneling protocol)

IpSec

SSL/TLS

2 modes:

Transport mode: behouden van originele IP header

* Originele pakket wordt niet geincapsuleerd in een ander IP pakket maar behoudt eigen header
* End-to end security
* End computers doen security processing (nt belangrijk om IP adressen te verbergen vr iemand)

Tunnel mode: alles, inclusief IP header verbergen, en het een nieuwe IP header geven

* Originele pakket incapselen in ander IP-pakket met een nieuwe IP header 🡪 verbergen van originele ip adressen (= privacy)
* Als 1 van de eindgebruikers een gateway is, dan moet tunneling mode gebruikt worden
* Security voor: network to network, host to network, host to host

IPSEC

Framework van open standards

* Werkt op netwerklaag
* Keuze tussen verschillende algoritmes

4 keuzes maken

* Welk IPSec protocol
* Welke encryptie
* Welke authenticatie
* Hoe afspreken shared key (Diffie Hellman)

Firewalls

Firewall = elk apparat dat als toegangscontrole-mechanisme voor een bepaald netwerk wordt gebruikt

* Meestal controle op externe toegang tot netwerk
* Gebruikt regels om toegang te verlenen of niet, controle op: bronadres, doeladres, poortnummers

Extra functionaliteit in sommige firewalls

* Filteren van inhoud (blokkering van toegang tot bepaalde sites, …)
* VPN (veilige tunen van A -> B)
* Network Address Translation (Reserved Address Space)
* Load Balancing (spreading van belasting)
* Intrusiedetectie

Soorten firewalls

Packet Filters (routers, Cisco, …) 🡪 L4 OSI model

* Voordeel: gemakkelijk te implementeren, meestal heeft men al een router
* Nadeel:
  + niet echt voorbereid op DOS-aanval
  + Kunnen meestal geen sessietoestanden bijhouden
  + Grote ACL’s kunnen bij grote belasting netwerkprestaties verminderen

Stateful packet filters (Checkpoint, PIX, …) 🡪 L5 OSI model

* Meestal minimumvereiste voor een firewall voor een network
* Doet pakketfiltering + houdt sessies en verbindingen bij in tabellen
  + Wordt er een protocolstandaard misbruikt (DoS aanval) -> kan men real time in logs zien
  + Kijkt niet alleen naar het pakketje, maar ook naar de context
  + Speciaal ontworpen tegen DoS attacks
  + Kunnen poorten boven 1024 standaard gesloten laten

Firewalls die als proxy fungeren 🡪 L7 OSI model

* Inspecteert verkeer ook op toepassingsniveau
* Deep packet inspection: http pakket arriveert, doorgeven aan http proxyprocedure, ftp pakket arriveert, doorgeven aan ftp proxyprocedure, …
* Meestal trager
* Als er een nieuw protocol wordt uitgevonden is er geen procedure voor

Verschil tussen gewone packetfiltering en stateful

Een ACL (Access Control List) is een lijst met allemaal rules (wie mag wat)

* Standaard wordt al het verkeer dat van het internet komt, en dat naar het intern netwerk wilt geblokkeert
* Als je wilt surfen wordt het antwoord van internet naar intern dus geblocked
  + Oplossing van gewone packetfilters: alles boven poort 1024 toelaten, is immers wrschl een antwoord op een request
  + Oplossing van statefull packetfilters: sessie bijhouden, en laten enkel verkeer binnen als de sessie gestart was in het interne netwerk

Tekortkomingen van firewalls

* Hoe hoger de beveiliging (striktheid), hoe minder de functionaliteit -> gewone gebruik van netwerk kan in gedrang komen
* Vormen soms een vals gevoel van veiligheid 🡪 zijn niet onfeilbaar, kijk goed logs na

Virussen en wormen

Wat is een virus

Een programma dat zich kopieert door andere progs te infecteren zodat zij ook een versie van het virus bevatten

Grootste kenmerk: replicatie

Infectie: het zich hechten van virus aan 1 of meerdere progs op het doelsysteem. Prog start op, virus activeert zich. Meestal dus fysiek gehecht aan dat prog, maar soms is booten al genoeg om te activeren

Werking

* User start legitiem prog (user interaction -> open dit prog, open deze mail, …)
* Virus wordt uitgevoerd ipv originele prog
* Viruscode wordt beeindigd, en geeft controle terug aan legitieme prog

Wat is een worm

Doet ook aan replicatie, maar manier van hechting is anders: het verspreid zich via netwerken/systemen zonder zich ergens aan te hechten

* Besmet zijn dus omgeving (OS, E-mailsysteem) ipv specifieke objecten zoals bestanden
* Meestal ook geen user interaction nodig 🡪 misbruik lekken in software, snellere verspreiding

In the wild: als een virus is ontsnapt of vrijgegeven aan het grote publiek

WildList Organisation (WLO): lijst bijhouden van virussen die momenteel circuleren ‘in the wild’

Virussen kunnen ingedeeld worden in:

* Boot sector infectors (BSI)
* Virussen die een bestand infecteren
* Multivirussen
* Macrovirussen
* Scriptvirussen
* Memetische virussen
* Wormen

Boot Sector Infector (BSI)

Kopieren de originele bootsectorcode naar een andere plaats: nadat het zijn viruscode heeft uitgevoerd, gewoon verder booten

Als een geinfecteerde diskette in de PC blijft: opstarten, PC probeert te booten vanaf diskette/USB,

Laadt viruscode, diskette USB is non-bootable, je krijgt melding dat het een non-bootable disk is, maar ondertussen is je systeem wel infected

Memetische virussen

Virussen zonder code, nepvirussen, hoax

* Zoek u pc voor folder system32, delete it, its a virus
* Meestal kettingbrief

Viruskenmerken

Stealth:

* Zijn min of meer stiekem
* Proberen aanwezigheid te verbergen om zo kans op ongemerkte verspreiding te vergroten
* Verdachte payload meestal vermeden, op onregelmatige tijdstippen verspreid, pas na vertraging activatie van de trigger 🡪 geeft virus de kans om zich te verspreiden
* Meeste virussen bewaren originele code, voor als het OS bepaalde info opvraagt

Polymorfie:

* Vroeger infecteerde virussen door een bijna exacte kopie aan ander object te hechten
* Polymorfie: iedere keer een iets ander virusvorm aan object hechten
* Viruscanners moeten dus soort van patroon herkennen

Trojans en Rootkits

Trojans

Een programma dat beweert een wenselijke of noodzakelijke functie uit te voeren (dat soms ook echt doet), maar dat daarnaast ook een functie uitvoert die onverwacht en ongewenst is voor de gebruiker

Belangrijkste verschil tussen virussen, wormen & trojans

Virussen en wormen zijn replicerende programma’s

Trojans zijn statische code (altijd geassocieerd met misleiding)

Backdoor Trojans

Backdoor: RAT (Remote Access/Admin Tool)

* Alternatieve weg om toegang te krijgen tot applicatie/system
* Meestal een bypass van de authenticatie

Legitiem gebruik: bv teamviewer, ultraVNC, …

Meestal Remote Access Trojan

* Trojan om toegang te krijgen zonder medeweten van het slachtoffer
* Hacker in control van het systeem

Application-level backdoor trojan

* Op victim wordt een server app geinstalleerd
* De hacker heeft een client om op die server te connecteren
* Bekendste voorbeeld: Back Orifice van Cult of the Dead Cow

Rootkit

Traditionele rootkits

* Zoals een RAT maar sneakier
* Gaat key system components aanpassen/vervangen
* Verbergt zich diep in je systeem
* User mode rootkit

Kernel level rootkit

Vervangen/aanpassen van de kernel

* Dus het opvangen of veranderen van system calls
* Zeer moeilijk te detecteren door security suites
* Zo een backdoor maken voor full system access

Sniffers

Apparaten die netwerkpaketten onderscheppen (wireshark)

Legitiem doel: analyse vh netwerkverkeer en signaleren van potentiele gevaren

Zijn dus protocolanalysers

Veiligheidsrisico’s

Sniffers vangen netwerkpakketten op door de NIC in overspel modus (promiscuous mode) te zetten

Ze zijn passief: alleen afluisteren 🡪 moeilijk op te sporen

Promiscuous mode: normaal pakt netwerkkaart alleen pakketten op die voor dat systeem bedoeld zijn (selectief) , rest negeert hij: overspel: alle pakketten die het ziet oppakken

Welke risico’s vormen sniffers

* Sniffers kunnen accountnamen en wachtwoorden afluisteren
* Sniffers kunnen vertrouwelijke of privé info afluisteren
* Kunnen worden gebruikt om de beveiliging van naburige netwerken te ondermijnen
* Goed geplaatste sniffer onderschept niet zomaar een paar wachtwoorden, maar duizenden

Welke info kunnen sniffers opvangen?

Sniffers kunnen alle pakketten opvangen, aanvaller maar moet kieskeuriger zijn

* Duizenden pakketten gaan over een netwerk, alleen eerste 2-300 bytes van een pakket, deze bevatten gebruikersnaam en wachtwoord

Aanmeldingsinfo is waardevolste voor aanvallers

* Info op poorten 23 (telnet), 80 (http), 21 (ftp) is waardevol, gebruikersnamen en wachtwoorden worden in deze protocollen in gewone txt verstuurd
* Gebruik SSL en SSH

Waarschijnlijkste plek om sniffer te vinden

Strategische punten zijn favoriet, vlakbij machine met veel wachtwoorden, of gateway naar ander netwerk-buitenwereld

Aanvallen van sniffers afslaan

Sniffers detecteren en elimineren

* Uitkijken naar NIC die in overspel mode staan
* Bereken van uw systeem een controlegetal (MD5)
* Gespecialiseerde progs

Data protection

* Veilige topologie: Sniffers kunnen alleen gegevens detecteren die hem passeren 🡪 splits uw netwerken in compartimenten (switches, routers, bridges)
* Versleutelde sessies: encrypteer de gegeven, zodat de sniffer het niet kan begrijpen

Spoofing

Geavanceerde techniek om de ene machine bij de andere te authenticeren door pakketten die afkomstig zijn van een vertrouwd bronadres, te vervalsen

Grondbeginselen van internet

* Vertrouwen
* Authenticatie
* Vertrouwen en authenticatie zijn omgekeerd evenredig

Rhosts (remote hosts)

Die en die mogen remote services gebruiken

Identificatie op basis van IP

Mechanisme van spoofing

Bronadres faken is niet genoeg, er is een dialoog tussen de machines nodig

* IP doet het pakkettransport, is onbetrouwbaar, er kunnen pakketten verloren gaan… Eerste stap in de verbinding is dus het aankomen van pakketten
* TCP neemt nu over. Kijkt na of de pakketten ok zijn: stuurt bericht terug dat het pakket is ontvangen en dat het ok is
* 3way handshake:
  + Client stuurt pakket met zijn seq-nr
  + Server stuurt ACK terug naar client met seq-nr+1, en stuurt zijn eigen seq-nr door
  + Client stuurt nu een ACK terug naar server met daarin het seq nr van de server +1

Aanvaller moet dus niet alleen bronadres vervalsen, hij moet ook de volgordenummerdialoog in stand houden (moet weten met welk seq nr de server start)

Non-blind spoofing: je ziet het server antwoord en dus zijn startseqnr, dankzij een sniffer

* Ip source routing: het pakketje komt voorbij de hacker, dus kan hij het kwijtspelen

Blind spofing: je ziet dit niet, dus moet je een beredeneerde gok doen naar dat seqnr

* je stuurt een gok naar de server
* server antwoordt naar de echte client (die je wilde spoofen, voordoen dat je hem bent)
* als die echt bestaat gaat hij een antwoord terug naar server sturen dat hij geen communicatie gestart had (reset)
* server stopt dan alle communiatie
* 🡪 je moet dus de client doen inslapen, of maken dat hij non-responsive is
  + Wachten tot hij niet ingelogd is
  + SYN-flooding:
    - Heel veel aanvragen tot connectie sturen naar client
    - Veel aanvragen, client kan de inkomende berichten van de server niet verwerken, want hij is continu bezig met die andere verzoeken, smijt het bericht zelfs weg bij overflow

Sequentie nummer raden: contact maken met server onder echte adres, je krijgt seqnr van server

Dit paar keer herhalen en een patroon proberen herkennen in de seqnr’s

Backdoor maken: eenmaal verbinding en authenticatie gedaan met server, server instellingen aanpassen zodat hij een trusted user is

Ingredienten geslaagde spoofing

* Hacker moet doelwitten identificeren
* Moet de source waarvoor hij zicht wil uitgeven verdoven (bv syn flooden)
* Vervalsen van het adres van de source
* Connectie maken met server als die source
* Backdoor maken

IP spoofing voorkomen

* Ingress filtering: blokkeren van pakketten die van buiten het netwerk komen, maar een intern IP adres hebben als source adres
* Egress filtering: blokkeren van pakketten die van interne lan komen, met als source een extern adres
* Geen ip source routing toelaten
* Geen proxy authenticatie van een firewall doen dmv IP bronadres
* Logprocedures maken en traffic monitoren

ARP Poisoning

Hacker stuurt ARP response naar target

* Response staat in: ikke met IP zoveel (trusted IP) heb MAC zoveel
* Binnenkomde ARP’s worden in de ARP cache gezet (ook ongevraagde)
* Target wil nu bericht sturen nar trusted IP, komt aan bij hacker want target zoekt MAC adres voor een bepaald IP om te addresseren

Meest gebruikte target voor ARP spoofing: Default gateway van LAN (de router)

* Hacker stuurt naar alle devices in LAN een ARP response
* Ieder device stuurt nu zijn data bestemd voor gateway naar device van hacker
* Hacker kun nu kiezen
  + Passive sniffing: alleen pakketjes bekijken en dan doorsturen naar echte gateway
  + Man in the middle attack: pakketjes veranderen en dan doorsturen naar gateway
  + DOS-attack: pakketjes allemaal tegenhouden, niemand kan buiten lan communiceren

Problemen bij ARP spoofing

* Cache wordt regelmatig vernieuwd op devices 🡪 hacker moet regelmatig ARP responses sturen om dit in stand te houden
* Werkt meestal enkel binnen dezelfde LAN

Countermeasures tegen ARP spoofing

* Statisch maken van de linking: niet automatisch cache updaten bij binnenkomdende ARP response
* ARPWATCH: houdt wijzigingen in IP mapping in de gaten, stuurt mail bij wijzigingen
* DHCP snooping: DHCP geeft IP adressen aan PC’s, (MAC’s), switches gaan die info gebruiken om te controleren of de mapping tussen IP en MAC klopt in de pakketjes

DNS spoofing

* Wijziging van DNS tabellen die url’s met IP-adressen linkt
* Makkelijk te ontdekken: pollen naar andere DNS servers om te kijken of die zelfde resultaat geven
* Hacker moet zowel voorwaardse als achterwaardse tabellen wijzigen

Denial of Service aanvallen

Aanval die tot gevolg heeft dat een bepaalde service verloren gaat of niet meer kan functioneren

Hoe werkt DoS

Alleen mogelijk omdat software programmeer fouten bevat

Aanval beoogt:

* zoveel mogelijk bandbreedte in beslag nemen
* Zoveel mogelijk bronnen verbruiken
* Programma’s en systemen laten crashen

Inbeslagname bandbreedte

Netwerk kan maar bepaalde hoeveelheid netwerk traffiek aan. Als de bandbreedte opgesoupeerd is, kunnen er geen nieuwe gegevens meer verzonden worden. Na aanval komt de bb weer ter beschikking

Meestal gebruikt gemaakt van fouten in het protocol

Smurf-aanval: aanvaller stuurt een ping request naar verschillende broadcast IP’s met een gespoofed IP-adres. 1 berichtje naar broadcast (veel berichtjes) die allemaal tegelijkertijd antwoord sturen naar target

Zoveel mogelijk bronnen verbruiken

Computer heeft maar een beperkt aantal bronnen beschikbaar (geheugen, opslag, processortime, …)

Syn-Aanval: veel halfopen verbindingen met doelserver tot stand brengen, misbruik van 3way handshake

http-aanval (op webserver): tegelijkertijd veel verbindingen tot stand brengen met de webserver

* Extra info vragen aan server die hij moet opzoeken (query)
* Server complexe berekeningen laten doen

PHP scripts: script zorgt ervoor dat server veel gegevens gaat opslagen, heel veel gegevens -> HD vol

Oplossing= human check/captcha

Programma’s en systemen laten crashen

Ping of death: grote ICMP echo (ping) verzoeken verznden

* Ping is normaal 64bytes, POD is max size IP pakket = 65536 bytes
* Meestal illegaal om over een netwerk te sturen 🡪 fragmenteren
* Target reassembleert de ping 🡪 buffer overflow want hij had geen 90k ping verwacht, en zeker geen 1000
* Voorkomen: zorgen dat alleen een selecte, goed beveiligde groep van computers de gateways mag beheren

E-mailbommen:

* Heel veel berichten naar uw postvak sturen 🡪 vult uw HD of HD van de mailserver
* Postvak vol: kunnen geen nieuwe berichten meer naar toe gestuurd worden
* HD mailserver vol: niemand kan nog e-mails naar die mailserver versturen
* Kan leiden tot serieus gegevensverlies, verhoogd BB verbruik, hogere netwerkkosten
* Oplossing: mailfilters en uitsluitschema’s

Distributed Dos aanvallen (DDOS)

Functioneert op basis van Master-Slave(zombie) principe

* Master is controlerende station: aanvaller definieert daarop zijn doel en methode
* Slaves zijn externe systemen waarop het aanvalsprog is geinstalleerd
* Master zegt tegen slaves: voer aanval uit, stop aanval
  + Moeilijk tegen te houden omdat het van zoveel verschillende systemen komt. Iedere keer checken of het systeem wel gemachtigd was of niet
  + Duurt dus redelijk lang om alles na te gaan 🡪 veel processingtime
  + Server kan het niet meer aan -> crash
* Bot-net, Zombie-net, master-slaves

Countermeasures:

* Filter onnodige services en netwerkgegevens uit je systeem
* Gebruik krachtig verificatiemechanisme
* Controleer toegang tot externe systemen grondig
* Actualiseer uw systemen en software regelmatige (lekken dichten)
* Load balancing

Port Scanners

Hoe wordt ik het doelwit van hackers

* Willekeurig
  + Poortscanner: bekendste = nmap
  + Geeft het prog een blok IP adressen om te controleren
  + Geeft terug welke computers verbonden zijn met die IP-adressen, welke poorten daar open staan, welk OS die comp gebruikt, … Aanvaller gebruikt die info om gekende kwetsbaarheden van het OS te misbruiken
* Gericht
  + Heeft u van te voren geselecteerd als doelwit
  + Eerst uitgebreid onderzoek naar u 🡪 spear phishing attack

Poortscanner

Scan een netwerkdevice voor openstaande poorten

* Crackers: hoe en waar in te breken
* Admins: hoe is de security, zijn er lekken

Portscan: kijk welke poorten er openstaan op 1 host

Portsweep: kijk of specifieke poort openstaat bij verschillende hosts

Gebaseerd op 3way handshake: krijg je ACK terug van de server op die poort

Kwetsbaarheidscanners

Zijn gereedschapen om kwetsbare plekken op te sporen in uw beveiliging

* Port scanner
* kwetsbaarheidsDB
* rapport

Port Scanner

Welk besturingssyteem heeft de PC, welke poorten staan open

Verschil met poort scanner: kan ook bepalen welke service op welke poort draait

Bepalen welke welke service op welke poort draait en welke versie het is

* Banner grabbing methode: connecteren naar bepaalde poort -> geeft tekst info -> regex gebruiken om nuttige info eruit te halen

Kwetsbaarheid DB

Interne database met info over zwakke plekken die bepaalde versies van services die draaien op een PC kunnen hebben

Database is meestal gevuld met alle CVE’s know to date

CVE : Common Vulerabilities and Exposures

Rapport

Helder meedelen wat ze gevonden hebben, zodat er actie kan ondernomen worden

Is classificatie van problemen, er kan dus een false positive of een false negative tussen zitten (confusion matrix)

Confusion matrix

Hoe goed is de classificatie

True positive: hoe dikwijls zeg je p, en was het ook echt p

False positive: hoe dikwijls zeg je p, en was het eigenlijk n

True negative: hoe dikwijls zeg je n, en was het ook echt n

False negative: hoe dikwijls zeg je n, en was het eigenlijk p

Intrusiedetectiesystemen (IDS)

Belooft een technologie te zijn die

* Vijandige indringers automatisch ontdekt
* Beheerders waarschuwt
* Indringers mogelijk tegenhoudt

Nog erg nieuw 🡪 kinderziektes

Soort inbraakalarm

Mogelijkheden tot detectie

* Logboeken van systeem, weg, firewall, routers bijhouden en controleren op voorvallen
* Binaire integriteitscontroles toepassing (hashing HDD)
* Eventslogs bijhouden en speuren naar mislukte aanmeldingspogingen
* Netwerkpackets checken op fingerprints van known attack vectors

2 grote groepen IDS’en

Misbruikdetectiemodellen

* Netwerk-IDS (snort): sniffers die alledaagse pakketten parsen, vergelijken dit met verzameling bekende aanvalspatronen of handtekeningen
* Host-IDS (tripwire): specifiek per systeem. Analyseren van logboeken, eventlogs.

Anomaliedetecitemodellen

* Gedrag van een gebruiker en het verkeer op het netwerk volgt enkele wetmatigheden
* Afwijkend gedrag hierop: onderzoek

Log management

Logboeken kunnen problemen helpen opsporen, trends weergeven of anomalieën op het netwerk ontdekken.  
Hacker leert als een van de eerste dingen hoe hij logboeken moet wijzigen om sporen te wissen

Logboek strategieën

Syslog-protocol

* Alle network devices op uw netwerk sturen hun systeemlogboeken naar een syslog server
  + Centrale opslag
  + Enige verkeer dat toegelaten is naar syslog server is via syslog poort (moeilijk om het daar nog te veranderen voor een cracker)
  + Hacker kan wel proberen de logs die gemaakt worden op de router te veranderen

Auditgereedschappen van een derde partij gebruiken (TTP= trusted third party)

* In combinatie met de standaard loggereedschappen op uw systeem
* Onafhankelijk logboekprog stelt zijn logs onafhankelijk samen

Je kan de 2 logboeken vergelijken 🡪 verschil = Hacker

Logprogs en analyseprogs

Er bestaan progs die niet alleen logs maken, maar ook logs analyseren en daar rapport van genereren

* Kan AI op toegepast worden om aanvals patronen te herkennen

Samenvatting Log management

* Verzamelen van grote hoeveelheden logs
* Logs naar centrale server sturen
* Logs analyseren om rapport te genereren
* Logs lang genoeg bijhouden

Siem

SIEM = Security Information and Event Management

Bestaat uit 2 delen: Sim & SEM

SIM

Security information management

* Is log management, maar dan specifiek voor security doeleinden om historische analyse te doen
* Long term storage, analyse en rapportering van logs voor security doeleinden

SEM

Security Event Management

* Real time monitoring van events
* Correlatie van alle data (context) is nodig om real time analyse te kunnen doen
  + Niet alleen meer syslogs, maar ook user accounts, firewall events, …

Big Data SIEM

Internet of (Every)Things

* Meer en meer devices op het network
* Ieder device genereert context informatie en logs
* Geeft een enorme berg aan data
* Big data tool snodig om al die data op een nuttige manier te verwerken

Big data tools:

* Niet alleen om security events uit te halen
* Heel wat mogelijkheden om non security events te filteren uit de data, om nieuwe inzichten te krijgen in het gebruik van de infrastructuur, optimalisatie van de servers, … = Operational intelligence

Splunk: Beste bid data engine van het moment

Privacy

Soorten aanvallers

Script kiddies

* Grootste groep, meestal onervaren, jong
* Surfen op web naar utilities, scripts, … die ervaren aanvallers daar gepost hebben
* Meestal mislukt hun aanval
* Eerder gegevens op een kwaadaardige manier beschadigen

Black hats: the dark side

* Zeer veel ervaring
* Zelden gepakt, doen grondig onderzoek, gerichte aanval
* Je hoort weinig over hun vaardigheden of capaciteiten (ninja’s van het internet)

White hats: the good guys

* Beveiligingsprofessionals die gaten opsporen en dichten
* Bedrijven huren hen in om beveiliging te testen
* Meestal zijn ze begonnen als black hat

BEsprionering

2 soorten:

* Collectieve informatievergaring: info verzamelen zonder direct contact te maken
* Penetrerende informatievergaring: contact maken, vertrouwen winnen, meer info vekrijgen

Browserbeveiliging

3 belangrijke methoden via de browser

* Sniffen naar ip-adres en in de cache
* Cookies
* Banneradvertenties en webbugs

Snuffelen naar IP-adressen in de cache:

* Meeste webservers registreren uw gegevens als je hen passeert
* Al die gegevens zorgen ervoor dat een cracker enorm veel info over jouw systeem te weten komt

Cookies:

* Worden gebruikt om u activiteit op een webpagina op te slaan
* Wordt ook gebruikt om login gegevens bij te houden: authentication cookie

Bannderadvertenties en webbugs

* Banneradvertentie: wanneer je een pagina met zo’n banner bezoekt wordt deze banner geladen vanuit een webserver die beheerd wordt door reclame bureau. Die gaat dan info opslaan over de gebruiker, eventueel in combi met cookie, javascript, … om profiel te maken = social engineering
* Webbug: transparant gif bestand, 1x1px groot, dat op zelfde manier als pixel werkt, maar je ziet het niet

Bescherming tegen webbugs gebruik een goede proxy-server en pop-up blokker

Cookie stealing with XSS

Cookies zijn authenticatie

* Een cookie bewijst aan de server wie je bent, welke rechten je hebt
* Als een hacker je cookie te pakken krijgt, kan hij zich voordoen als u

Same origin principle

* Een script van site X kan enkel die cookies gebruiken afkomstig van domein X

XSS: cross-site scripting: cookie stealing

* Xss: proberen om het same origin principle te omzeilen en dus scripts van de hacker zijn domein te laten uitvoeren, zoals scripts die van een trusted domein komen
* Hacker wilt uw facebook cookie stelen 🡪 script om dit te doen moet door het facebook.com domein aan u gegeven worden, anders mag het niet aan de cookie

Stored (persistent) XSS:

* Vulnerabilities van de website zelf gaan exploiten om daar uw cookie stealing script te plaatsen
* Gelijk welke user komt naar die site 🡪 hacker kan aan zijn cookie

Reflected (non persistent) XSS:

* Soort van witwassen van het script, door het te laten returnen door een trusted site aan het target
* Principe is om code injection door de client te laten doen zonder zijn medeweten
  + Client misleiden om script code naar een site te sturen
  + Site valideert code niet goed en returned zijn antwoord, met de scriptcode in, terug aan de client
* Op deze manier is de script code witgewassen

Voorbeeld Reflected XSS:

* Victim klikt op een link in een bad email
* Deze link stuurt data (script-x) naar de searchbar van website X via de victim zijn browser
* Deze data is zo opgesteld dat de searchbar van site X deze niet goed begrijpt
* Website x stuurt dus een antwoord terug naar de client: ik weet niet wat ‘script-x’ is
* Dit antwoord komt in de victim zijn browser window
* Maar nu komt het script wel van het domein, en mag het script dus wel aan de cookie van X
* En het script suurt nu het echte cookie van site X naar de hacker