# **VPN**

**VPN heeft 2 modussen**

Tunnel mode:

Deze pakt de volledige data en encrypteerd zowel het IP adress. Dan wordt daarvoor nog eens apparte Ip adressen gezet: Ze zien dus dat ge van gateway naar gateway gaat maar ze weten niet wie of wat, dus ook ni de andere persoon of u. niemand kan gezien worden behalve van gateway naar gateway.

Transport mode:

Neemt een datastukske , die encrypteerd dat , en die zet daarvoor een ipsec header hoe ge zegt da ge dat hebt geyncrepteerd., Niemand kan die data encrypteerd lezen, maar ze kunnen zien wie met wie comuniceerd

**PGP (= betalend)**

Ge hebt uwe mail, ge stuurd die van a naar b 🡪 dus ge stuurd da door en AES 🡪 dan hebt ge een geyncrepteerde mail . Voor AES te gebruiken moet ge die sleutel door RSA gebruiken, voor die sleutel te hebben gebruikt ge de puclic key van B.

Hoe ben je zeker dat die van A komt, gebruik ne Hash 🡪 en die hash stuurde door ne RSA 🡪 met de private key A. En die 3 dingen (Geyncrypteerd bestand. RSA Hash, RSA key) naar B , dan kan die alles den.

PGP gebruikt web of trust (dus geen PKI gebruik) 🡪 dus ge maakt u eigen ertificaat met eigen key. En Dan tekenen mensen die u kennen dat certificaat

**SSL/TLS**

Secure Socket Layer 🡪 Socket = IP met de poort.

Examen: SSL bestaat uit 2 lagen, SSL record protocol en bovenste laag SSL Alert, SSL change chiper spec , SSL H.S

SSL Record (basislaag) 🡪

1. Data wordt in stukskes gekapt en stuurd die dan door 🡪 fragmenteren
2. Data compressen
3. Message Authentication code(MAC) toevoegen, een soort van controle getal

**MAC**

Preshared key en een message door ne Hasher gooien

Ge gaat een Master Secret moeten maken 🡪 dat is een master paswoord en ge gaat daaaruit subpaswoorden berekenen.

1. Dit geheel gaan ge encrypteren met AES (met daarvoor een SSL header)

SSL Alert 🡪 Als er iets misgaat

SSL change cipher spec 🡪 Vanaf nu gaan we naar de encryptiemethode gaan die we net hebben afgesproken gebruiken

SSL Handshake 🡪 Ik bene en client en wil veilig comuniceren, die gaat een bericht sturen 🡪 hier is mijn certificaat , die moet die geldigheid controleren 🡪 OCSP gebruiken voor die public key krijgen. , dan genereer je een pré master secret , dat stuur je terug 🡪 want die anderre persoon heeft enkel die private key. De server gaat een quick check doen, en die doet een Proof of Possesion.

**EXAMEN: Wat is de SSL Handshake afhankelijk van deze figuur (zoek figuur op powerpoint)**

4 Fasen:

1. Client Hello 🡪
   1. Encrypties die je aankan
   2. Alle compressies aan
   3. Random getal van de client

Server Hello 🡪

1. We kiezen deze encyptie
2. We kiezen deze compressie
3. Random getal van de server
4. Server – Side 🡪
   1. Server stuurt certificaat naar de client

Heeft deze geen certificaat, dan stuurt die een server key exchange

* 1. Server vraagt een certificate\_request aan de client
  2. Server Hello Done

1. Client-Side
   1. Vooraleer stap 3 start, Certificaat van server controleren

Stel dat deze check faalt, daan faalt SSL connectie

* 1. Nu start stap 3: Cerrtificaat wordt doorgestuurd naar server

Als de client geen certificaat heeft doet die een SSL-Alert

* 1. Client Key Exchange wordt gestuur naar de server

Dit is de pré-master secret en die is geyncrepteerd met de RSA public key server.

* 1. Proof of Possion check

Certificat verify bericht

Alle berichten die tot nu toe zijn uitgewissled gaat ge een hash van maken en encrypteren met je eigen private key, de server doet dit ook en gaat de gesigneerde hash decrypteren en als die hetzelfde zijn, dan weet je dat het goed is.

1. Client stuurt CCS
   1. Server stuurt Change Cypher Spec
   2. Einde, je hebt cconnectie

Wat is de functie nu van die random getallekes ?

* + - 1. Pre-MS-Secret
         1. AES key, HMAC key
      2. Replay- Attack

Het random getal wordt op de client gebruikt van zowel de server als de de client dus kan je dit niet doen.