Web security: bedreigingen

	Threats	Consequences	Countermeasures
Integrity	•Modification of user data •Trojan horse browser •Modification of memory •Modification of message traffic in transit	•Loss of information •Compromise of machine •Vulnerabilty to all other threats	Cryptographic checksums
Confidentiality	•Eavesdropping on the Net •Theft of info from server •Theft of data from client •Info about network configuration •Info about which client talks to server	•Loss of information •Loss of privacy	Encryption, web proxies
Denial of Service	•Killing of user threads •Flooding machine with bogus requests •Filling up disk or memory •Isolating machine by DNS attacks	•Disruptive •Annoying •Prevent user from getting work done	Difficult to prevent
Authentication	•Impersonation of legitimate users •Data forgery	•Misrepresentation of user •Belief that false information is valid	Cryptographic techniques

SSL & TLS 1

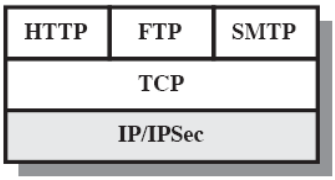
Web security: bedreigingen

- Soort
  - actief
  - passief
- Waar
  - Web server
  - Web browser (client)
  - netwerk trafiek tussen client en server
- focus op laatste item (trafiek)
- meerdere benaderingen zijn mogelijk

SSL & TLS 2

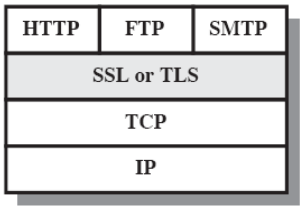
Beveiliging webtrafiek: IP-niveau

- d.m.v. IPSec
- transparant voor eindgebruikers en toepassingen
- algemene oplossing
- bovendien: trafiek kan gefilterd worden
  ➔ vermindert verwerking IPSec



(a) Network Level

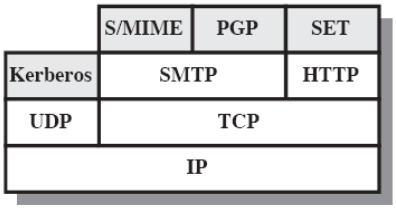
Beveiliging webtrafiek: boven TCP



(b) Transport Level

- beveiliging net boven TCP-laag
- algemeen toepasbaar voor ≠ toepassingen
- belangrijkste voorbeeld:
  - SSL: secure socket layer
  - TLS: transport layer security (= internetstandaard)
- implementatie als aparte laag of als deel van applicatie

Beveiliging webtrafiek: applicatie



(c) Application Level

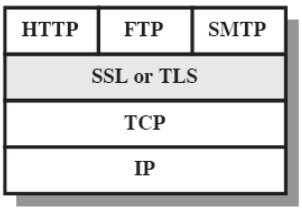
- beveiliging geïntegreerd in applicaties
- voordeel: beveiligingsbehoeften kunnen op maat gemaakt worden
- voorbeeld: SET secure electronic transaction
  - bovenop HTTP
  - als aparte applicatie bovenop TCP

SSL en TLS

- SSL : Secure Socket Layer
  TLS : Transport Layer Security
- Geschiedenis
  - SSL ontwikkeld door Netscape (samenwerking met RSA)
  - gepubliceerd als Internet draft
  - oprichting van TLS-werkgroep binnen IETF (Internet Engineering Task Force)
  - TLS versie 1.0 kan bekeken worden als SSL versie 3.1
  - ook compatibel met SSL versie 3.0
  - SSL 3.1 t.o.v. 2.0: client-auth en meer encryptieschema's

SSL en TLS: opzet

- gebruikt TCP
- opzetten van privaat kanaal tussen 2 applicaties
- meest geïmplementeerd in HTTP (via https)
- andere toepassingen: NNTP, Telnet, LDAP, POP3, DNS, ...



(b) Transport Level

SSL & TLS 7

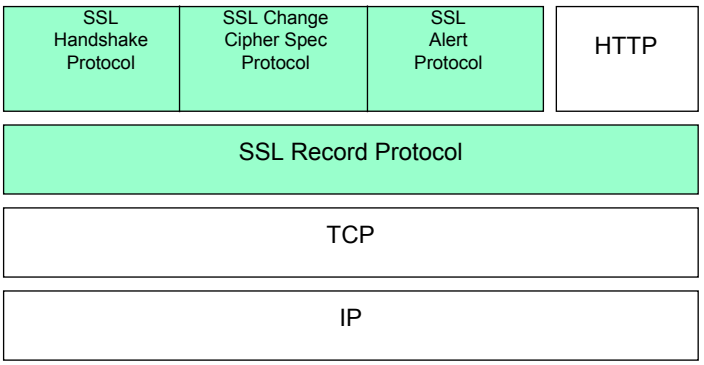
SSL overzicht

- Extra laag tussen toepassing en TCP
  - in theorie:
    - aparte socket interface tussen TCP en willekeurige toepassing
  - in praktijk:
    - maakt vaak deel uit van de toepassing
- SSL bestaat eigenlijk uit twee lagen
  - SSL record protocol: uitwisseling van gegevens, ook die van de applicatie
  - protocols voor uitwisseling van
    - initiële authenticatie
    - encryptiesleutels

  (zie figuur op volgende slide)

SSL & TLS 8

SSL overzicht (2)



SSL overzicht (3)

- SSL protocol voorziet in
  - geheimhouding
    - **gebruik van geheime sleutel (uitgewisseld in initiële handshake) voor encryptie van boodschappen**
  - integriteit
    - **boodschappen bevatten een MAC**
  - authenticatie
    - **server wordt geauthenticeerd door client via publieke sleutel bij begin van een SSL sessie of via certificaten**
- Vergt “veel” processorkracht
  - encryptie en decryptie van elke boodschap

SSL overzicht (4)

- via browser vraagt gebruiker document via url *https://. . .*
- client software (browser)
  - herkent SSL-aanvraag
  - verwezenlijkt verbinding met SSL code op server via TCP poort 443
  - initieert SSL Handshake fase
  - gebruikt SSL Record Protocol
  - hier nog geen encryptie of integriteitcontrole
- kan ook vb via e-mailclient (secure smtp of pop)

SSL concepten

- Verbinding (connection)
  - is een transport (OSI) dat een service levert
  - peer-to-peer relatie
  - tijdelijk
  - hoort bij 1 sessie
  - meerdere verbindingen tegelijkertijd mogelijk
- Sessie
  - associatie tussen client en server
  - gecreëerd door Handshake Protocol
  - definiëren beveiligingsparameters
  - te gebruiken door meerdere verbindingen
  - vermijdt nieuwe onderhandelingen voor elke verbinding
- tssn client en server meerdere connections in 1 sessie

Staten van SSL verbinding

- **sessie** status
  - complex, zo min mogelijk gebruikt
  - onderhandelt parameters
- **connectie** status
  - peer-to-peer relatie

meerdere connecties voor één sessie mogelijk

Staten van SSL verbinding (2)

1. Sessie status

- associatie tussen client en server
- gecreëerd door SSL Handshake Protocol
- definieert set van cryptografische parameters, bruikbaar voor meerdere verbindingen
- gedefinieerd door
  - **session ID**: gekozen door server
  - **peer certificaat**: volgens X.509 (kan leeg zijn)
  - **compressiemethode**: null in huidige standaard
  - **cipher spec**: vr encryptie (AES …) & MAC berekening (SHA-1…)
  - **master secret**: 48 bytes gedeeld door client en server
  - **is resumable**: vlag die aangeeft of sessie nog nieuwe connnecties kan initiëren

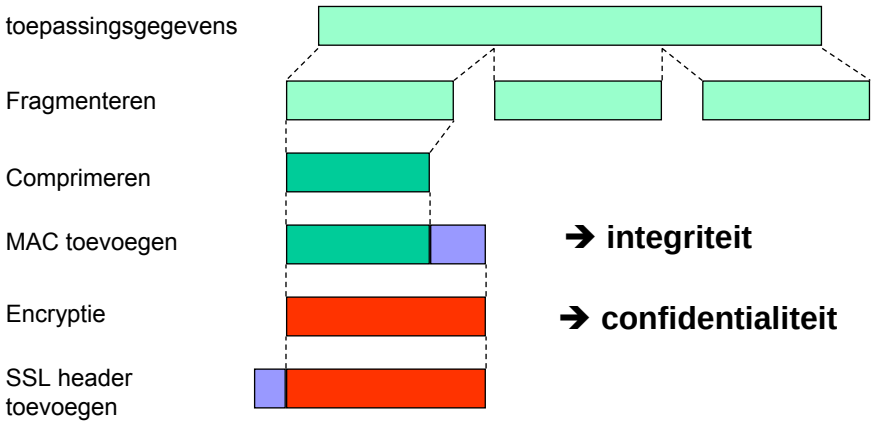
Staten van SSL verbinding (3)

2. Connectie status

- is geassocieerd met één sessie
- legt gegevens vast voor zenden en ontvangen
- gedefinieerd door
  - **server en client random**: gekozen door server en client voor elke connectie
  - **server write MAC secret**
  - **client write MAC secret**
  - **server write key:** voor conventionele encryptie
  - **client write key**
  - **initialization vector**: voor CBC mode cipher
  - **sequence numbers**: elke uitgewisselde boodschap krijgt een volgnummer (=0 bij nieuwe cipher specs)

SSL Record Protocol

- gebruikt voor elke SSL uitwisseling



SSL Protocol

- Taken van zender
  - boodschap van applicatielaag fragmenteren tot behandelbare blokken (≤ 2^{14} of 16384 bytes)
  - optioneel: comprimeert gegevens
  - berekent MAC met behulp van afgesproken sleutel en hash-algo
  - encrypteert data met behulp van afgesproken sleutel en algo
  - geeft resultaat door aan TCP-laag
- Taken van ontvanger
  - decrypteert gegevens van TCP-laag
  - verifieert data met onderhandelde MAC-sleutel
  - optioneel: decomprimeert gegevens
  - assembleert boodschap
  - geeft boodschap door aan applicatielaag

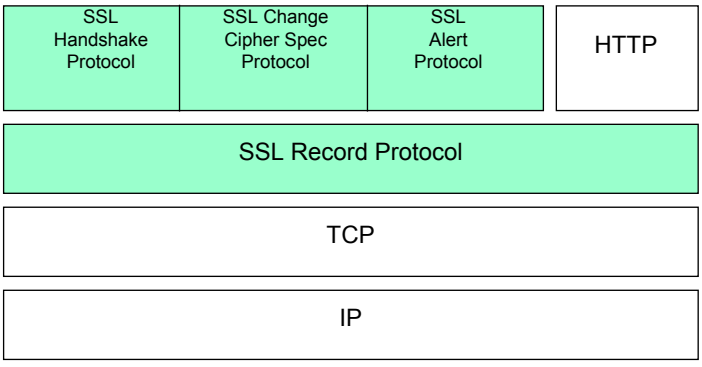
SSL Record Protocol (2)

- gefragmenteerde blokken <= 2^{14} bytes (16k)
- compressie is optioneel; in huidige standaard niet gespecificeerd
- berekening **MAC** via hash algoritme in functie van
  - MAC_write_secret
  - hash (algoritme-aanduiding, i.e. MD5 of SHA-1)
  - pad_1: 0x36 (48x voor MD5, 40x voor SHA-1)
  - pad_2: 0x5C (48x voor MD5, 40x voor SHA-1)
  - sequence number of message
  - protocoltype van hogere laag
  - lengte van (gecomprimeerd) fragment
  - (gecomprimeerd) fragment
  - **is dus vergelijkbaar met HMAC ! (concat ipv xor)**

SSL Record Protocol (3)

- **encryptie** met een van volgende algoritmen:
  - block ciphers
    - IDEA(128), RC2-40 (40), DES-40 (40), DES (56), 3DES (168)
    - volgens RFC 5246 (TLS 1.2, aug 2008): 3DES, AES
  - stream ciphers
    - RC4-40 (40), RC4-128 (128)
- SSL-**header** bevat
  - Content Type (8 bits): definieert protocol op hogere laag
    - change_cipher_spec, alert, handshake, application_data
  - Major Version Number (8 bits): typisch 3
  - Minor Version Number (8 bits): typisch 0
  - Compressed Length (16 bits): bytelengte van plaintext fragment (al dan niet gecomprimeerd)

SSL overzicht (2)



Change Cipher Protocol

- één boodschap van 1 byte met waarde 1
- zegt aan de ontvanger dat de overeengekomen sleutelwaarden vanaf nu mogen worden gebruikt
- de *status in afwachting* wordt gekopieerd in de *huidige status*

Alert protocol

- verwittigt tegenpartij van een fout
- bestaat uit 2 bytes:
  - waarde: 1 (warning) of 2 (fatal)
  - soort fout
- fatal: (SSL beëindigt verbinding)
  - unexpected_message, bad_record_mac, decompression_failure, handshake_failure, illegal_parameter
- overige:
  - close_notify: zender zal deze verbinding niet meer gebruiken
  - no_certificate, bad_certificate, unsupported_certificate, certificate_revoked, certificate_expired, certificate_unknown

Handshake Protocol (1)

- bepaalt parameters voor een SSL gegevensuitwisseling
- zorgt voor de authenticatie van client en server
- wordt gebruikt vóór verzending van applicatiegegevens

Handshake Protocol (2)

- bestaat uit een reeks berichten uitgewisseld tussen client en server
- elk bericht bevat
  - Type (1 byte): (10 soorten)
  - Length (3 bytes): lengte van het bericht in bytes
  - Content (>= 1 byte): parameters geassocieerd met bericht

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

Handshake Protocol (3)

4 fasen

- opzetten van veiligheidsmogelijkheden
- server authenticatie en uitwisseling van sleutels
- client authenticatie en uitwisseling van sleutels
- einde

SSL Handshake Protocol (4)

Client		Server
Client Hello	→	
	←	Server Hello
	←	(Certificate)
	←	(Server Key Exchange)
	←	(Certificate Request)
	←	Server Hello Done
(Certificate)	→	
(Client Key Exchange)	→	
(Certificate Verify)	→	
Change Cipher Specs	→	
Finished	→	
	←	Change Cipher Specs
	←	Finished

SSL Handshake Protocol (5)

Fase 1

- **Client Hello:** connectie-aanvraag
  - gewenste SSL-versie
  - random getal (nonce)
  - ciphersuite: bepaalt een reeks …
    - cryptografische algoritmen in dalende orde van voorkeur
    - manieren voor authenticatie en sleuteluitwisseling, encryptie en mac-algoritmen
  - ondersteunde compressiemethoden
  - Session-ID:
    - = 0 betekent nieuwe sessie en nieuwe connectie
    - ≠ 0 betekent bestaande connectie aanpassen of nieuwe connectie in bestaande sessie

SSL Handshake Protocol (6)

Fase 1

- **Server Hello**
  - gekozen SSL-versie
  - nieuw random getal (nonce)
  - ciphersuite: bevat
    - uitgekozen manier voor sleuteluitwisseling (RSA, DH, ...)
    - uitgekozen cipher specificaties
      - encryptiealgoritme + type (stream, block)
      - MAC algoritme (MD5, SHA-1)
      - materiaal om sleutels te kunnen aanmaken
  - ondersteunde compressiemethoden
  - Session-ID:
    - gelijk aan dat van client indien dat niet nul was
    - nieuw indien nieuwe sessie

SSL Handshake Protocol (7)

Fase 2

- Server Certificate: X.509 certificaat van server
- Server Key Exchange + authenticatie
  - noodzakelijk o.a. indien geen certificaat aanwezig of als certificaat alleen geldig is voor handtekening
  - niet nodig indien o.a. RSA-sleuteluitwisseling
  - getekende hash van boodschap bevat ook random nonces van 1 en 2
- Certificate Request
  - server vraagt client certificaat met vermelding van soort en aanvaardbare CA's (indien clientauthenticatie gewenst is)
- Server Hello Done
  - betekent dat de server nu zal wachten op het antwoord van de client

SSL Handshake Protocol (8)

Fase 3

client controleert servercertificaat en parameters fase 2

- Client Certificate
  - kan ook no_certificate alert zijn
- Client Key Exchange
  - op basis van publieke-sleutelalgoritmen afgesproken in fase1
- Certificate Verify
  - indien van toepassing wordt hash genomen van gegevens van client certificaat en certificeringspad
  - zorgt voor expliciete verificatie van client certificaat
  - wordt als extra handtekening teruggestuurd naar server

SSL Handshake Protocol (9)

Fase 4

- Client zendt
  - Change_Cipher_Spec
    - valt niet onder handshake protocol
    - kopieert voorlopige cipher specs in de definitieve
  - Finished
    - hash
    - onderhandelingsfase is afgelopen
    - deze boodschap wordt met nieuwe algoritmen en sleutels verzonden
- Server zendt als antwoord
  - Change_Cipher_Spec
  - Finished

Gegevens kunnen nu veilig worden uitgewisseld nadat sleutels zijn berekend

IPSec = SSL ?

Gelijkaardig want

- IPSec (via IKE) en SSL voorzien in client en server authenticatie
- IPSec en SSL voorzien in authenticatie en geheimhouding van data, evenwel op verschillende lagen
- ze gebruiken sterke cryptografe algoritmen en certificaten
- ze kunnen sleutels genereren “uit het niets” zonder dat deze in clear text worden verzonden

IPSec ≠ SSL

Verschillend want

- niveau van implementeren
  - SSL als API tussen applicatie- en transportlaag
  - IPSec in internetwerklaag
- veiligheid
  - SSL: applicatie tot applicatie
  - IPSec: device tot device
- SSL beveiligt niet de IP-headers, IPSec wel
- SSL beveiligt geen UDP trafiek, IPSec wel
- SSL werkt alleen end-to-end (geen tunneling), IPSec kan beide
- applicaties moeten SSL-aware zijn, IPSec is transparant

SSL & TLS 33