DIS - Sikkerhed

Security Threads

Masquerade
Hopper ind i en forbindelse og agere som enten modtageren eller afsenderen. Lidt sværere at beskytte sig i mod

Leakage
wikileaks, IBM ansat der var konsulent hos Nets som har siddet og overvåget personer på deres dankort

Acquisition of information by an unauthorized recipient.

Eksempel: Edward Snowden, eller Helle Thorning Schmidts skattesag.

Vandalism
Hvad gør man for at undgå det – Prepared statements – man afslutter alle strengene selv, før de sættes ind i databasen. Beskyttelsen er rimelig nem – men heletiden en balance man skal finde, fordi det hele tiden koster noget at beskytte sig..

Eksempel: DDos angreb, SQL dumps, take-overs

Interference with the proper operation of a system without a gain to the perpetrator.

hærværk, sker oftest. SQL Injections – når man injecter en SQL statement.... SQL Injection – typisk eksempel hvor virksomheder er mest sårbarer

Eavesdropping
lytter til vores trafik – får en kopi – fx bankinformation. Hvordan kan det undgås? – Kryptere sin forbindelse – secret key og public private key.

Tampering
Unauthorized altering of information

Eksempel: Afstemningsfusk, biler

Hvad kan man gøre for at undgå – Kan man lave noget validering – Hvis valideringen foregår på klient-siden, så kan man fx gå ind i javascript koden og ændre koden, og på den måde ændre koden og blive valideret fx

Vigtigt ved tampering er også at kigge på hvor lang tid denne her pakke har været på vejs. Hvis jeg nu sender en pakke med tidsstempel og med en checksum, så har man

Replaying
Opsnappe data og så sende det afsted på et andet tidspunkt.

Hvad kan man gøre? Tidslås – fx at købe for præcis det samme beløb. Denne her pakke har nu været en uge eller dag undervejs. Man kan ikke se hvor lang tid en pakke er undervejs, men jeg kan se hvornår den er afsendt. Så er der så tidsforskydningen mellem serverne.

Social Engineering

Denial of Service
Man kan bare lukke for en IP adresse. Hvis en IP adresse skyder 200.000 forespørgsler afsted, så lukker vi bare for den. Typisk være firewall'en der gør det her. I stedet for block ham, kan man også bare sende ham til ingenting – grunden til at man gør det, er fordi at han så tror, at siden er nede, fordi han ikke får noget svar.

DDOS – til eksamenen må man meget gerne lade være med at sige, at en løsning er at stoppe ens egen hjemmeside – hvad er meningen med det? Hvad kan man gøre i stedet for?

- Man kan skalerer, men det giver ikke meningen – skalerer for at opdele trafikken på flere servere
- Den samme pakke bliver sendt rigtig mange gange – det man kan få firewallen til sige, at hvis denne her pakke går ind og rammer dig, så smid den væk – null router vi den. Hvad er det vi kan kigge på i en firewall? Hvilken protokol – TCP UDP?, hvor kommer det fra? Porten? Man kan godt kryptere trafikken, men det er stadig den samme pakke. Så man skal enten ændre pakken. Det er samme pakke der kommer.

Det er ikke i en browser – man laver bare et script der sender pakker.

Når vi begynder at sidde og kode, så kan man komme til at lukke sig selv

Flooding a channel or other resource with messages in order to deny access for others

Et DDoS angreb er når flere computere samtidigt "angriber"/overflyder en servers kapacitet, så serveren til sidst ikke længere kan håndtere det, og eventuelt går ned eller slår fejl.
Man kan benytte sig af en firewall, der kigger på de pakker der bliver sendt til serveren - er det den samme pakke? hvor kommer den fra? hvilken protokol og port benytter den? På den måde opsætte en firewall der ikke tillader for mange "entry's" af samme pakke.

Hashing
- Rainbow Tables
- SALT

SSL

The Handshake
Encryption

1. Computers agree on how to encrypt - Client sends this to the server:
- Key exchange method (RSA)
- Cipher (AES, XOR)
- Hash (MD5, SHA) - Generate a message authentication code - send with the messages, to ensure the integrity of them
- Version of SSL (TLS)
- Random number used to compute the master secret which is then used from then on to calculate the encryption keys

2. Server sends certificate - Who the server belongs to
- How long the certificate is valid for
- Serial numbers
- **The public key**

3. Your computer says 'start encrypting'
- Client key exchange - Both computers calculate a master secret code (used from now on)
- Change cipher spec: your computer is asking server to encrypt
- Finished

4. The server says 'start encrypting'
- Change cipher spec: Im going to send encrypted messages now
- Finished: lets go (this is encrypted)

5. All messages are now encrypted

Why it exists

Encryption
Hiding what is sent from one computer to another

Identification
Making sure the computer you are speaking to is the one you trust. You can have an SSL certificate but that doesnt mean that the one you are talking to is the one you think it is, it just means that the communication between you is secure.

1. Company asks CA (Certification Authority) for a certificate
2. CA creates certificate and signs it (hashed) (encrypts the hash with a private key) (so everyone with a public key can see it is correct)
3. Certificate installed in server
4. Browser issued with root certificates
5. Browser trusts correctly signed certificates

Wiki
The public key algorithms known thus far are relatively computationally costly compared with most symmetric key algorithms of apparently equivalent security. The difference factor is the use of typically quite large keys. This has important implications for their practical use. Most are used in hybrid cryptosystems for reasons of efficiency – in such a cryptosystem, a shared secret key ("session key") is generated by one party, and this much briefer session key is then encrypted by each recipient's public key. Each recipient then uses his own private key to decrypt the session key. Once all parties have obtained the session key, they can use a much faster symmetric algorithm to encrypt and decrypt messages. In many of these schemes, the session key is unique to each message exchange, being pseudo-randomly chosen for each message.

Firewall
The use of firewalls offers no protection against attacks from inside an organization, and it is crude in its control of external access

Kryptering
C = E_k(m)

Encryption is the process of encoding messages or information in such a way that only authorized parties can read it

Digital Signature
Når en server sender information afsted, så hasher den en besked med, som netop er den digitale signatur. Derefter krypterer den den hashede værdi med **private key**.

Når noget er krypteret med private-keyen, så kan den kun dekrypteres med public keyen. Derfor hvis en MitM intercepter og prøver at tamper med beskeden før modtageren får den, vil modtageren ikke længere kunne dekryptere, da MitM's forsøg har betyder at han har krypteret med noget andet end private-keyen, da den ikke er tilgængelig hos andre end afsenderen.

As long as the receiver can decrypt the message with the sender's public key, they know that the sender is the correct sender, as it is encrypted with the private key