Remote X Apps mini-HOWTO

Autheur: *Vincent Zweije, zweije@xs4all.nl*
Vertaler: *Reggy Ekkebus, reggy@zeelandnet.nl*
v, 19 November 1999

Deze mini-HOWTO beschrijft hoe je remote X applicaties kan draaien. Dit betekent dat je een X programma van de ene computer op een andere computer kunt draaien. Simpel gezegd: hoe kan ik een X programma draaien op een andere computer dan de computer waar ik achter zit. Het kenmerk van deze mini-HOWTO is veiligheid. Deze mini-HOWTO bevat ook informatie over het runnen van X applicaties lokaal met verschillende ussers.

Inhoudsopgave

1 Introductie

Deze mini-HOWTO is een handleiding over hoe te doen met remote X applicaties. Hij is geschreven om verschillende redenen.

1. Er zij al veel vragen gesteld over het draaien van remote X applicaties in de newsgroepen.

2. Ik zie heel veel hints als "gebruik `xhost + hostname`"of zelfs "`xhost +`" om X connecties toe te staan. **Dit is belachelijk onveilig**, want er zijn betere methodes.
3. Ik ken geen eenvoudig document dat de opties beschrijft die je hebt. Mail mij *zweije@xs4all.nl* als jij iets beters weet.

Dit document is geschreven met unix achtige systemen in mijn gedachten. Maar ook al is je lokaal of remote besturings systeem van een ander merk, dan kun je hier toch vinden hoe sommige dingen werken, je zal wel enige dingen moeten aanpassen voor gebruik met je eigen systeem(en).

De meest recente versie van dit document is altijd aanwezig op WWW *http://www.xs4all.nl/˜zweije/xauth.html*. Het is ook altijd aanwezig op de Remote X Apps mini-HOWTO op *http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps*. Linux (mini-)HOWTOs zijn aanwezig op http of ftp van *sunsite.unc.edu*.

Dit is versie 0.6.1. Geen waarborg, enkel goede bedoelingen. Ik ben open voor suggesties, ideeën, nuttige toevoegingen, (fout) correcties, Etc .. Ik wil dat dit een simpel document blijft, alhoewel met de beste bedoeling In HOWTO stijl. De laatste update dateert van 19 November 1999 door *Vincent Zweije*

2 Aanverwante documentatie

Een vergelijkbaar document op WWW is "What to do when Tk says that your display is insecure", *http://ce-toolkit.crd.ge.com/tkxauth/*. Dit is geschreven door *Kevin Kenny*. het is een vergelijkbare oplossing voor X authentificatie zoals (xauth) in dit document. Alhoewel Kevin zich meer spitst op het gebruik van xdm met xauth.

Het X Windows systeem Vol 8 X "Window System Administrator's Guide" van *O'Reilly and Associates* is ook een goede bron van informatie. Enkel ben ik niet in staat geweest om dit te checken.

Een ander document dat veel lijkt op dat wat je nu leest met als titel "Securing X Windows" is aanwezig op *http://ciac.llnl.gov/ciac/documents/ciac2316.html*.

Kijk ook eens in de newsgroepen, zoals `comp.windows.x`, `comp.os.linux.x`, en `comp.os.linux.networking`.

3 Een voorbeeld

Je gebruikt twee computers, en je gebruikt het X windows systeem van de eerste om te typen en om naar te kijken. Je gebruikt de tweede om wat belangrijk grafisch werk op te doen. Je wilt de tweede gebruiken om naar de uitvoer op het scherm te kijken van de eerste. Dit maakt het X windows systeem mogelijk.

Je hebt hier natuurlijk een netwerk-connectie voor nodig. Het liefst een snelle; Het X protocol is een netwerk slak. Maar met een beetje geduld en een geschikt compressie protocol, kun je zelfs X applicaties runnen via een modem. Voor het X compressie protocol moet je even kijken op dxpc *http://ccwf.cc.utexas.edu/˜zvonler/dxpc/* of LBX *http://www.ultranet.com/˜pauld/faqs/LBX-HOWTO.html* `<http://www.ultranet.com/~pauld/faqs/LBX-HOWTO.html>` (ook bekent als *LBX mini-HOWTO*).

Je moet twee dingen doen om dit te volbrengen:

1. Vertel het lokale display (server) dat hij connecties moet accepteren van een remote computer
2. Vertel de remote applicatie (client) dat hij zijn output naar het lokale display moet sturen

4 Een beetje theorie

Het magische woord is `DISPLAY`. In het X window systeem bestaat het display (simpel gezegd) uit een keyboard, muis en een scherm. Een display wordt aangestuurd door een server programma beter bekend als de X server. De server bediend de display mogelijkheden van programma's die ermee connecten.

Een display wordt aangegeven met een naam bv:

- `DISPLAY=light.uni.verse:0`
- `DISPLAY=localhost:4`
- `DISPLAY=:0`

Een display bestaat uit een host naam (zoals `light.uni.verse` en `localhost`), een dubbele punt (`:`), een volgorde nummer (zoals `0` en `4`). De host-naam van de display is de naam van de computer waar de X server draait. Als je de host-naam weg laat betekent dit de lokale host. Het volgorde nummer is normaal 0 - maar het kan varieer-en als er meerdere Displays zijn aangesloten op een computer.

Als je ooit een display tegenkomt met een extra `.n` indicatie erachter, dan is dat het scherm nummer. Een display kan eigenlijk meer schermen aan. Een display kan eigenlijk meerdere schermen aan. Normaal is er maar een scherm, met het nummer `n=0`, dus dat is standaard.

Andere formaten van `DISPLAY` zijn er, maar die hierboven doet het goed voor onze plannen.

Voor technische nieuwsgierigheid:

- `hostname:D.S` Betekend `S` op display `D` van host `hostname`; de X server voor dit display luistert op de TCP poort `6000+D`.
- `host/unix:D.S` betekend scherm `S` op display `D` van host `host`; de X server voor dit display luistert op de UNIX domein socket `/tmp/.X11-unix/XD` listening (dus hij is alleen berijkbaar vanaf `host`)
- `:D.S` is hetzelfde als `host/unix:D.S`, waar `host` de locale hostname is.

5 De client vertellen

Het client programma (Bijvoorbeeld je grafische applicatie) weet naar welk display hij moet connecten door te kijken naar het `DISPLAY` variabel. Deze instelling kan veranderd worden, door de client als optie `-display hostname:0` te geven als je hem opstart. Sommige van deze voorbeelden zullen dit duidelijker maken.

Onze computer, is bekend voor de andere computers als light, en we zitten in het domein uni.verse. Als we een normale X server draaien, de display is bekend als `light.uni.verse:0`. We willen het tekenprogramma xfig runnen op een remote computer, die als naam heeft `dark.matt.er`, en hij moet zijn output laten zien op light.

Ik ga er van uit dat je op de remote computer (`dark.matt.er`) bent in gelogd.

Als je csh draait op de remote computer:

```
dark% setenv DISPLAY light.uni.verse:0
dark% xfig &
```

of:

```
dark% xfig -display light.uni.verse:0 &
```

Als je sh of bash draait op de remote computer:

```
dark$ DISPLAY=light.uni.verse:0
dark$ export DISPLAY
dark$ xfig &
```

of:

```
dark$ DISPLAY=light.uni.verse:0 xfig &
```

of:

```
dark$ xfig -display light.uni.verse:0 &
```

Het ziet er naar uit dat sommige versies van telnet automatisch het `DISPLAY` variabel gelijk goed zetten op de remote host. Als je er zo een hebt dan heb veel geluk en moet je het niet allemaal met de hand doen. Zo niet, de meeste versies van telnet trans porteren het `TERM` variabel; met een beetje hacken is het mogelijk om het `DISPLAY` variabele met het `TERM` variabel mee te laten komen.

Het idee van het transporteren is een beetje scripting om het volgende te bereiken: voordat je telnet, bevestigen we het variabel `DISPLAY` aan `TERM`. Dan telnet op de remote host, in de `.*shrc` file, lees het variabel `DISPLAY` van `TERM`.

6 De server vertellen

De server zal niet zomaar een connectie van iedereen accepteren. Je wilt toch niet dat iedereen zo maar een window op je scherm kan zetten. Of lezen wat je schrijft – onthoud je keyboard is onderdeel van het display!

Veel mensen schijnen zich niet te realiseren dat het toestaan van toegang tot je display een groot security risico is. Iemand met toegang to je display kan lezen van en schrijven naar je scherm, alles lezen wat je typt, en je muis acties registreren.

De meeste servers kennen twee manieren voor het legaliseren van connecties naar de server: het host lijst mechanisme (xhost) en het magic cookie mechanisme (xauth). Dan is er ook nog ssh, de secure shell, die kan X connecties ook doorsturen.

6.1 Xhost

Xhost laat connecties toe op hostnaam. De server houd een lijst bij van host die mogen connecten. Het kan ook host checking volledig uitschakelen. Onthoud: dat betekend dat er geen checks meer uitgevoerd worden, dus *iedere* host mag connecten!

Je kan de servers host list bijhouden met het programmatje xhost. Om dit te gebruiken moet je het mechanisme uit het volgende voorbeeld gebruiken.

```
light$ xhost +dark.matt.er
```

Dit staat connecties van de host `dark.matt.er` toe. Zodra je X client een connectie heeft gemaakt en een window weergeeft, schakel voor de veiligheid meer connecties uit met:

```
light$ xhost -dark.matt.er
```

Je kan host verificatie uitschakelen met:

```
light$ xhost +
```

Dit schakelt de host toegang verificatie uit en dus *iedereen* mag connecten. Je moet dit *nooit* doen op een netwerk waar je niet *alle* gebruikers vertrouwd (internet bijvoorbeeld). Je kunt host verificatie weer aanzetten met:

```
light$ xhost -
```

xhost - verwijderd *niet* alle host-Namen uit zichzelf van de toeganslijst (dat zou niet echt slim zijn - je kan dan niet meer connecten naar je Xserver van waar dan ook - zelfs niet vanaf localhost).

Xhost is een zeer onveilig mechanisme. Het maakt geen onderscheid tussen verschillende gebruikers op de remote host. Ook host-Namen (eigenlijk adressen) kunnen gespoofd worden. Dit is vervelend als je op een onbetrouwbaar netwerk zit (internet bijvoorbeeld).

6.2 Xauth

Xauth staat connectie toe aan iedereen die het juiste geheim weet. Zo'n geheim wordt authorization record genoemd, of magic cookie. Dit legalisatie schema is formeel genoemd MIT-MAGIC-COOKIE-1.

De cookies voor verschillende displays staan samen is `~/.Xauthority`. Jouw `~/.Xaurthority` moet niet toegangkelijk zijn voor andere groepen en gebruikers. Het xauth programmatje houd deze cookies bij, vandaar de bijnaam xauth voor het schema.

Bij het starten van een sessie, leest de server de cookie uit het bestand die aangegeven wordt door de optie `-auth`. Nadat, de server alleen connecties vanaf clients toelaat met de zelfde cookie. Als de cookie in `~/.Xauthority` veranderd, *de server zal de verandering dan niet doorvoeren.*

Nieuwere servers genereren cookies gelijk als clients er om vragen. Cookies blijven nog steeds behouden binnenin de server; ze komen niet terecht in `~/.Xauthority` behalve als een client ze daar zet. Volgens David Wiggins:

> Een volgende plooi is toegevoegd aan X11R6.3 daar zou je in geïnteresseerd kunnen zijn. Via het nieuwe SECURITY extensie, kan de X server zelf nieuwe cookies genereren en terugplaatsen ad-hoc. Verder, de cookies kunnen on vertrouwd worden aan gesteld daarom worden applicaties met zulke cookies beperkt in hun handeling. Bijvoorbeeld, ze kunnen dan niet de invoer van muis en keyboard lezen, of de inhoud van windows, van andere vertrouwde gebruikers. Er is een nieuw sub commando gemaakt voor xauth om dit mogelijk te maken voor gebruik.

Xauth heeft is duidelijk veiliger dan xhost. Je kan beperkte toegang voor bepaalde gebruikers en hosts instellen. Het struikelt niet over gespoofde adressen zoals xhost. En als je wilt kun je xhost er ook nog bij gebruiken.

6.2.1 De Cookie maken

Als je xauth wilt gebruiken, moet je de X server starten met de optie `-auth authfile`. Als je het startx script gebruikt, dan is dat goede plaats om het te doen. Maak een authorization record net als hieronder in je startx script.

Stukje uit `/usr/X11R6/bin/startx`:

```
mcookie|sed -e 's/^/add :0 . /'|xauth -q
xinit -- -auth "$HOME/.Xauthority"
```

Mcookie is een heel klein programmatje in het util-linux pakkage, hoofd site *ftp://ftp.math.uio.no/pub/linux/*. Als alternatief kun je ook md5sum gebruiken om verschillende data (van, bijvoorbeeld `/dev/urandom` of `ps -axl`) in cookie om te zetten:

```
dd if=/dev/urandom count=1|md5sum|sed -e 's/^/add :0 . /'|xauth -q
xinit -- -auth "$HOME/.Xauthority"
```

Als je het startx script niet kunt aanpassen (omdat je geen root bent) ga dan naar je systeem administrator en vraag hem om dit te doen, of laat hem xdm starten. Als hij het niet kan of wil, dan maak je een `~/.xserverrc` script. Als je het script hebt, zal xinit het runnen ipv de echte X server. Dan kun je de echte X server starten vanuit het script met de goed opties natuurlijk. Om dat te doen, laat je `~/.xserverrc` de magic cookie regel gebruiken en dan de echte X server starten:

```
#!/bin/sh
mcookie|sed -e 's/^/add :0 . /'|xauth -q
exec /usr/X11R6/bin/X "$@" -auth "$HOME/.Xauthority"
```

Als je xdm gebruikt om je X sessie bij te houden, kun je xauth gemakkelijk gebruiken. Zet het regeltje 'DisplayManager.authDir in `/etc/X11/xdm/xdm-config`. Xdm zal nu de optie `-auth` gebruiken als de server start. Als je inlogd bent met xdm, dan zet xdm de cookie in `~/.Xauthority` voor je. Zie xdm(1) voor meer informatie. Bijvoorbeeld, mijn `/etc/X11/xdm/xdm-config` heeft de volgend regel:

```
Display Manager.authDir: /var/lib/xdm
```

6.2.2 De cookie transporteren

Als je de X server gestart hebt op de server host `light.uni.verse` en je hebt je cookie in `~/.Xauthority`, dan moet je de cookie transporteren naar de client `dark.matt.er`.

Het makkelijkst is als je home dir op light en dark zijn gedeeld. De `~/.Xauthority` files zijn het zelfde, dus de cookie wordt gelijk getransporteerd. Maar, er zit een addertje onder het gras: als je een cookie voor `:0` in de file zet dan denkt dark dat het voor zichzelf is ipv voor light. Dus je moet de volledige host-naam gebruiken als je de cookie maakt; Je kunt het niet weg laten. Je kunt dezelfde cookie installeren voor `:0` en `light:0` met:

```
#!/bin/sh
cookie=`mcookie`
xauth add :0 . $cookie
xauth add "$HOST:0" . $cookie
exec /usr/X11R6/bin/X "$@" -auth "$HOME/.Xauthority"
```

Als de homedirectory's niet zijn gedeeld, kun je de cookie transporteren door middel van rsh, de remote shell:

```
light$ xauth list "${HOST}:0" | rsh dark.matt.er xauth nmerge -
```

1. Haal de cookie uit je locale `~/.Xauthority` (`xauth nlist :0`).
2. Verplaats het naar dark.matt.er (`| rsh dark.matt.er`).

3. Zet het in `~/.Xauthority` daar (`xauth nmerge -`).

Notitie voor het gebruik van `${HOST}`. Je moet de cookie transporteren die samenhangt met local host. Een remote X applicatie zal het display value `:0` gebruiken op de remote machine, dat is niet wat je wilde!

Het is mogelijk dat rsh niet voor je werkt. Naast dat, heeft rsh ook een security probleem (gespoofde host namen). Als je niet wil of kan gebruik maken van rsh, kun je het ook handmatig transporteren, zoals dit:

```
light$ echo $DISPLAY
:0
light$ xauth list $DISPLAY
light/unix:0 MIT-MAGIC-COOKIE-1 076aaecfd370fd2af6bb9f5550b26926
light$ rlogin dark.matt.er
Pass-word:
dark% setenv DISPLAY light.uni.verse:0
dark% xauth add $DISPLAY . 076aaecfd370fd2af6bb9f5550b26926
dark% xfig &
[15332]
dark% log out
light$
```

Zie ook rsh(1) en xauth(1) voor meer informatie

Het is mogelijk om de cookie te transporteren met het `TERM` variabel of `DISPLAY` variabel als je tel-net doet naar een remote host. Dit werk het zelfde als het transporteren van het `DISPLAY` variabel met het `TERM` variabel. Zie Sectie 5: De client vertellen.

6.2.3 De Cookie gebruiken

Een X applicatie op dark.matt.er, zoals xfig, zal automatisch kijken in `~/.Xauthority` om zichzelf te legaliseren.

Er is een klein probleem als je gebruikt `localhost:D`. X client applicaties kunnen dit vertalen in `host/unix:D` voor het doel om de cookie te ontvangen. Dat betekend dat de cookie voor `localhost:D` in je `~/.Xauthority` *geen* zin meer heeft.

6.3 Ssh

Authorizatie records worden gezonden zonder encryptie. Als bang bent dat iemand je connectie afluistert, gebruik dan ssh, de secure shell. Het gebruikt X forwarding over geëncrypte connecties. En daarnaast is het geweldig op andere manieren. Het is een goede structurele toevoeging aan je systeem. Ga naar *http://www.cs.hut.fi/ssh/* , de ssh home page.

Wie weet er andere manieren voor legaliserings schema's of encrypted X connecties? Misschien kerberos?

7 X Applicatie van een andere User-id

Stel dat je een grafische applicatie wilt draaien met root authenthificatie. Alhoewel je X sessie onder je eigen account draait. Op het eerste gezicht lijkt dit vreemd, maar de X server kan de tool toegang *niet* toestaan op jou display. Hoe is dit mogelijk, als de root normaal alles mag doen? en hoe ga je dit probleem oplossen ?

Laat ons focussen op de situatie, je wilt een X applicatie onder een User-id `clientuser`, maar de X sessie is gestart door `serveruser`. Als je de sectie over cookies hebt gelezen, dan is het duidelijk waarom `clientuser` geen toegang krijgt tot jou display: `~clientuser/.Xauthority` heeft niet de goede magic cookie voor toegang tot de display. De goede cookie kun je vinden in `~serveruser/.Xauthority`.

7.1 Verschillende gebruikers op dezelfde Host

Alles dat werkt voor remote X werkt natuurlijk ook voor X van een ander User-id (vooral `slogin localhost -l clientuser`). het is waar dat de client host en de server toevallig hetzelfde zijn. Alhoewel, als de hosts hetzelfde zijn, dan is er een kortere weg om de magic cookie te transporteren.

We nemen aan dat jij gebruikt maakt van `su` om van User-id te wisselen. Wat je normaal kunt doen is een script schrijven om `su` aan te roepen, maar zet in het script dat `su` execute enkel de dingen die nodig zijn voor remote X. Dit zijn de `DISPLAY` variabele en de transfer van de magic cookie.

Het vaststellen van het `DISPLAY` variabel het is best simpel; het betekend alleen het vaststellen van `DISPLAY="$DISPLAY"`

voordat je het su commando optie start. Dus je kunt het zo doen:

```
su - clientuser -c "env DISPLAY=$DISPLAY client-program &"
```

Dit werkt nog niet, omdat we nog steeds de cookie moeten transporteren. We kunnen de cookie ophalen door middel van `xauth list "$DISPLAY"`

. Die commando gebeurd om de cookie in een format goed formaat te krijgen voor het terug sturen naar `xauth`; dat is wat we willen! Dus we sturen de goede cookie terug naar `xauth` in het `su` commando, het variabel `DISPLAY daar`, en het commando starten wat we willen.

```
su - clientuser -c "xauth add `xauth list $DISPLAY`; \
                    exec env DISPLAY=$DISPLAY client-program"
```

We kunnen een scripts rond dit alles schrijven met de volgende variabelen `clientuser` en `client-program`. Laten we het script een beetje verbeteren en het minder leesbaar maken. Het ziet eruit als dit:

```
#!/bin/sh
if [ $# -lt 2 ]
then echo "usage: `basename $0` clientuser command" >&2
     exit 2
fi
CLIENTUSER="$1"; shift
exec su - "$CLIENTUSER" -c "xauth add `xauth list \"$DISPLAY\"`; \
                            exec env DISPLAY='$DISPLAY' "'"$SHELL"'" -c '$*'"
```

Ik denk dat het wel werkt in de meeste situaties. De enige tekort komming die ik met kan met nu kan indenken is dat, door het gebruik van `'$*'`, single quotes in `command` zullen het `su` commando met argument (`'$*'`) een beetje in de war brengen. Als je denk dat er iets echt mis is mail me.

Roep het script aan `/usr/local/bin/xsu`, en je kunt doen:

```
xsu clientuser 'command &'
```

Makkelijk?, nee

7.2 Client gebruiker is Root

Blijkbaar kan alles dat voor niet root client gebruikers werkt ook werken voor root. Alhoewel als root kun je het gemakkelijker maken , dit omdat de root iedereen zijn `~/.Xauthority` file mag lezen. Het is niet nodig de cookie te transporteren. Het enigste dat je hebt te doen is het `DISPLAY` zetten, en `XAUTHORITY` naar `~serveruser/.Xauthority`. Dus je kan:

```
su - -c "exec env DISPLAY='$DISPLAY' \
                  XAUTHORITY='${XAUTHORITY-$HOME/.Xauthority}' \
                  command"
```

Als je hier een script van maakt, dan ziet het er ongeveer zo uit:

```
#!/bin/sh
if [ $# -lt 1 ]
then echo "usage: `basename $0` command" >&2
     exit 2
fi
su - -c "exec env DISPLAY='$DISPLAY' \
                  XAUTHORITY='${XAUTHORITY-$HOME/.Xauthority}' \
                  "'"$SHELL"'" -c '$*'"
```

Noem het script `/usr/local/bin/xroot`, en dan kun je:

```
xroot 'control-panel &'
```

Kan het nog eenvoudiger, nee?

8 Een Remote Window Manager draaien

Een window manager (als `twm`, `wmaker`, of `fvwm95`) is een applicatie als elk ander. De normale procedure zou dus moeten werken.

Vaak, in elk geval een window manager kan op elke tijd op een display draaien. Als je al een lokale window manager hebt draaien, dan kun je niet nog een remote starten (het wordt te ingewikkeld en stopt) je moet dan de lokale quiten of killen.

Veel X sessie scripts eindigen ongelukkig met:

```
exec window-manager-of-choice
```

En dat betekend dat als de lokale window manager stopt, ook je sessie stopt. Het X systeem (xdm of xinit) neemt je sessie over en logt jou uit.

Je moet een beetje moeite doen, maar het kan en het is niet echt moeilijk, speel eens wat met je session script (normaal `~/.xsession` of `~/.xinitrc`) om het voor elkaar te krijgen.

Onthoud dat sommige window managers vaak de optie bieden om nieuwe programma's te starten en dat zal runnen op de lokale machine. Dat is, lokaal waar de window manager draait. Als je een remote window manager draait, zal het ook remote applicaties starten en dat is niet wat je wilt. Natuurlijk zullen ze wel verschijnen op wat voor jouw lokaal is.

9 Problemen oplossen

De eerste keer dat je een remote applicatie probeert te draaien, is het normaal dat het niet werkt. Hier een paar normale fout boodschappen, en mogelijke oplossingen om je op weg te helpen

```
xterm Xt error: Can't open display:
```

Er is geen `DISPLAY` variabele in de omgeving, en je hebt de applicatie niet vertelt welke `-display` joker. De applicatie heeft een lege string aangenomen, maar dat is syntactisch niet goed. Om dit op te lossen moet je de goede `DISPLAY` omgevings variabele zetten (met `setenv` of `export` afhankelijk van je shell).

```
_X11TransSocketINETConnect: Can't connect: errno = 101
xterm Xt error: Can't open display: love.dial.xs4all.nl:0
```

Errno 101 is "Network is unreachable". De applicatie kan geen netwerk- connectie maken naar de server. Kijk of je de goede `DISPLAY` heb gezet, en of de server berijkbaar vanaf jou machine (dit zou goed moeten wanneer je al op de server bent in gelogd en kan tel-netten naar de client)

```
_X11TransSocketINETConnect: Can't connect: errno = 111
xterm Xt error: Can't open display: love.dial.xs4all.nl:0
```

Errno 111 is "Connection refused". De server waar je naar probeert te connecten is bereikbaar, maar de desbetreffende server bestaat daar niet Kijk of je de goede host-naam gebruik en het goede display nummer.

```
Xlib: connection to ":0.0" refused by server
Xlib: Client is not authorized to connect to Server
xterm Xt error: Can't open display: love.dial.xs4all.nl:0.0
```

De client kan connectie maken met de server, maar de server geeft geen toestemming voor de client (niet geauthoriseerd). Weet je zeker dat je magic cookie goed hebt getransporteerd, en dat deze nog geldig is (de server gebruikt steeds een nieuwe cookie als er een nieuwe sessie wordt gestart).