X Apps 원격 실행 미니 하우투(Remote X Apps mini-HOWTO)

VincentZweije, zweije@xs4all.nl 14 July 1998

이동규 bookbug@net.kitel.co.kr 1998년7월26일

이 mini-HOWTO는 엑스 윈도우 응용프로그램을 원격으로 실행시키는 방법을 설명한다. 좀더 정확히 말하면, 엑스 윈도우 프로그램을 조작중인 컴퓨터밖에 다른 컴퓨터 display상에 실행시키는 방법에 대한 것이다. 혹은 거꾸로: 당신이 앉아있는 컴퓨터밖에 다른 컴퓨터에서 엑스 윈도우 프로그램이 실행되도록 하는 방법에 대한 것이다. 이 mini-HOWTO의 논점은 보안을 깨뜨리지 않는 수준이다.

목 차

제 1 절 소개(Introduction)

이 mini-HOWTO는 원격으로 조작 가능한 엑스 윈도우 응용프로그램을 어떻게 실행시키는가에 대한 안내서이다. 이 글은 몇 가지 이유로 쓰여졌다.

1. 원격으로 조작 가능한 엑스 윈도우 응용프로그램을 어떻게 실행시키는 가란 물음이 토론그룹에 많이 보여왔다.

2. 나는 엑스 윈도우에 연결하기 위해선 "xhost +hostname"또는 "xhost +"을 "이용"하라는 변죽을 울리는 답변을 수없이 보았다. **이것은 터무니없이 불만족스런 답변이었고**, 더 조리 있는 답변들이 있을 수 있었다.

3. 나는 당신이 프로그램을 실행할 때 택*하지 않을 수 없*는 사항들에 대해 설명하는 간단한 문서가 있다고 듣지 못했다. 만약 당신이 더 좋은 방법을 알고 있다면 나에게 알려주기 바란다.

이 문서는 유닉스계열 시스템들을 염두에 두고 쓰여져왔다. 당신의 로컬 또는 로컬의 운영체제가 다른 성질의 것이라면, 어떻게 동작하는가를 당신이 찾아내야 될지도 모른다. 그러나, 당신의 독특한 시스템에 적용하기 위해서 예제들은 당신 자신이 다른 형식으로 바꾸어야 할 것이다.

이 문서의 최신 버전 대부분은 웹 `http://www.xs4all.nl/\~{}zweije/xauth.html` 에서 항상 손에 넣을 수 있다. 역시 `http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps`에서도 Linux Remote X Apps mini-HOWTO로 손에 넣을 수 있다. Linux (mini-)HOWTO들은 `sunsite.unc.edu`로부터 http나 ftp를 이용해서 손에 넣을 수 있다.

이 글은 버전 0.5.1이다. 글을 쓰는데 수고비는 없었다, 단지 좋은 뜻만이 있었을 뿐이다. 나는 제안, 견해, 추가할 사항, 유익한 지적, (타자)교정 등을 받아들인다. 나는 이 글이 간단하고 읽기 쉬운 문서로 남아 있기를 바란다, 그래도, 가장 의미 있는 양식인 HOWTO 양식으로 남아 있기를 바란다. 잡담은 `/dev/null`에 뿌리기 바란다.

차례는 1998년 7월 14일에 `VincentZweije`가 마지막으로 갱신했다.

제 2 절 참고 자료(Related Reading)

웹에서 참고한 문서는 "What to do when Tk says that your display is insecure", `http://ce-toolkit.crd.ge.com/tkxauth/`이다. `KevinKenny`에 의해서 쓰여졌다. 이 문서에 엑스 윈도우 인증(Xauth)에 대한 부분과 유사한 해결책을 제안한다. 그러나, Kevin은 당신에게 Xauth을 설명할때 xdm을 이용하는 경우에 더 많이 중점을 둔다.

`O'ReillyandAssociates`로부터 출판된 X Window System Vol. 8 "X Window System Administrator's Guide" 또한 정보의 출처로써 나의 주의를 끌어왔다. 불행하게도, 나는 그것을 다 읽을 수 없는 상태에 있다.

다른 문서는, 당신이 지금 당장 읽으려면 `http://ciac.llnl.gov/ciac/documents/ciac2316.html`에서 구할 수 있는, "Securing X Windows" 란 제목의 문서를 무척 좋아한다.

`comp.windows.x`, `comp.os.linux.x`, `comp.os.linux.networking`과 같은 유즈넷 뉴스그룹들에도 잘 갖추어져 있음을 알기 바란다.

제 3 절 상황(The Scene)

당신은 두 대의 컴퓨터를 사용하고 있는 중이다. 당신은 첫 번째 엑스 윈도우 시스템을 워드 작업과 워드 작업의 결과를 보기 위해 사용중이다. 두 번째 엑스 윈도우 시스템은 중요한 몇 가지 그래픽 작업을 위해 사용중이다. 당신은 첫 번째 엑스 윈도우 시스템 디스플레이상에 두번째 엑스 윈도우의 그래픽 작업의 출력이 보여지길 원한다. 엑스 윈도우 시스템은 이 행동을 가능하도록 해준다.

물론, 당신은 이 행동을 위한 네트워크 연결이 하나 필요하다. 택할만한 것으로; X protocol은 네트워크 대역을 아주 많이 사용한다. 그러나 약간 인내만 있으면 되도록 해주는 적절한 porotocol 압축이 있다. 심지어 당신은 모뎀을 통해서도 엑스 윈도우 응용프로그램들을 실행시킬 수 있다. X protocol 압축에 대해선, 당신이 dxpc `http://ccwf.cc.utexas.edu/\~{}zvonler/dxpc/`나 LBX *http://www.ultranet.com/~pauld/faqs/LBX-HOWTO.html* `<http://www.ultranet.com/~pauld/faqs/LBX-HOWTO.html>`를 살펴보기 바랄지도 모르겠다 (`LBXmini-HOWTO`로도 잘 알려져있다.)

당신은 이 행동하기 위해 두 가지 일을 해야 한다:

1. 원격 컴퓨터로부터 연결을 허락하는 로컬 디스플레이(서버)를 일려주기.
2. 당신의 지역 디스플레이에 출력을 뿌릴 원격 엑스 윈도우 응용프로그램(클라이언트)를 일려주기.

제 4 절 약간의 논의(A Little Theory)

DISPLAY는 불가사의한 힘을 가진 단어이다. 엑스 윈도우 시스템에서, 디스플레이 하나는 (평이하게) 키보드 하나, 마우스 하나, 스크린 하나로 이루어져있다. 하나의 디스플레이는 엑스 서버로 알려진, 하나의 서버 프로그램이 관리한다. 이 서버가 각 엑스 윈도우 응용프로그램이 응용프로그램 자신에게 연결된 디스플레이에 출력을 제대로 출력하도록 도와준다.

한 개의 디스플레이는 한가지 이름으로 표시한다, 예를 들면:

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

DISPLAY는 호스트 이름 (light.uni.verse나 localhost와 같은) 하나와 콜론 (:) 하나, 순번 (0나4와 같은) 하나로 이루어져있다. DISPLAY의 호스트 이름은 엑스 서버를 실행시키고 있는 컴퓨터 이름이다. 호스트 이름을 생략하면 로컬 호스트를 지칭한 것이 된다. 순번은 대개 0이다 – 여러 개의 디스플레이들이 한 컴퓨터에 연결되어있다면 다양하게 바뀔 수 있다.

만약 당신이 별도로 .n가 붙은 디스플레이 표시를 접해본 적이 있다면, 그것은 스크린 번호이다. 하나의 디스플레이는 실지로 여러 개의 스크린을 가질 수 있다. 그렇더라도 대개 한 개의 스크린만을 가지고 있으므로,n=0이 디폴트로 되어있다.

DISPLAY의 다른 양식도 존재한다, 그러나 그 부분은 우리의 목적을 넘어서는 것이므로 다루지 않는다.

제 5 절 클라이언트 일려주기(Telling the Client)

클라이언트 프로그램(예를 들어, 당신의 그래픽 응용프로그램)은 환경변수 DISPLAY를 살펴서 어느 디스플레이에 연결해야 하는가를 정한다. 그러나, 이 설정은 명령행 옵션 -display hostname:0이 프로그램이 실행을 시작할 때 클라이언트에 전해지는 것으로, 무시 될 수 있다. 설명을 명확하게 해줄지도 모르는 어떤 예가 있다.

우리의 컴퓨터는 외부에서 light로 식별하고, 도메인 uni.verse 상에 있다. 우리가 보통의 엑스 서버를 실행중이라면, 디스플레이는 light.uni.verse:0로 식별한다. 우리는, dark.matt.er라 불리는, 원격 컴퓨터상에서 그리기 프로그램 xfig를 실행하고, 여기 light에 그것의 출력을 표시하기를 원한다.

당신은 이미 원격 컴퓨터 dark.matt.er에 telnet접속이 되어있다고 가정한다.

당신이 원격 컴퓨터에서 csh을 실행중이라면:

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

또는 대신에:

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

당신이 원격 컴퓨터에서 sh을 실행중이라면:

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

또는 대신에:

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

또는 당연히:

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

telnet의 어떤 버젼은 자동으로 원격 호스트 DISPLAY변수에 로컬 호스트 DISPLAY변수를 전달한다. 만약 당신이 이런 종류의 telnet을 가지고 있다면, 당신은 운이 좋은 것이다, 수동으로 변수를 설정않해도 된다. 그렇지 않은 경우라면, 대부분의 버전의 telnet은 TERM 환경 변수만을 전달한다; 약간의 사려 깊은 해킹으로 DISPLAY 변수가 TERM 변수와 같이 전달되도록 하는 것이 가능하다.

같이 전달되도록 하는 개념을 당신은 다음의 작업들을 수행할 수 있는 약간의 스크립트 작성으로 구현이 가능하다: telnet접속을 하기전에, DISPLAY의 값을 TERM에 붙인다. 그리고 나서 telnet접속을 한다. 원격접속이 이루어진 후에, 쉘에 따라 실행되는 .*shrc 파일이 실행 중에, TERM로부터 DISPLAY값을 읽는다.

제 6 절 서버 일려주기(Telling the Server)

서버는 아무 곳으로부터 접속을 허락하지 않을 것이다. 당신은 당신의 스크린에 아무나 윈도우를 열 수 있는 것을 원하지 않는다. 혹은 당신이 타이프한 것을 아무나 읽을 수 있는 것을 원하지 않는다. – 당신의 키보드는 당신 디스플레이의 일부임을 기억하라!

소수의 사람들은 지나치게 디스플레이에 억세스를 허락하는 것을 보안 위험성을 높이는 것으로 재 정의하는 것같다. 당신의 디스플레이에 억세스 중인 누군가가 당신의 스크린들에 읽고 쓰기와, 당신이 누른 키 읽기와, 당신의 마우스 동작 읽기를 할 수는 있다.

대부분의 서버들은 서버에 연결을 인증하는 방법 두 가지를 알고 있다. host list mechanism (xhost)과 magic cookie mechanism (xauth)이 그것이다. 그 다음으로는 ssh(the secure shell)이 있는데 엑스 윈도우 연결을 향상시킬 수 있다.

6.1 Xhost

Xhost는 호스트 이름에 근거를 두고 엑세스를 허락한다. 서버는 서버에 연결을 허락한 호스트 목록을 유지한다. 역시 호스트 확인을 완전히 불가능하게 할 수도 있다. 주의하라: 이것은 확인을 전혀 하지 않게 됨을 의미한다. 그래서 모든 호스트가 연결이 가능할 것이다!

당신은 xhost 프로그램으로 서버의 호스트 목록을 관리할 수 있다. 이전의 예에서 이 기법(mechanism)을 이용하기 위해서는, 이렇게 하라:

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

이것은 호스트 dark.matt.er로부터 모든 연결을 허락한다. 당신의 엑스 윈도우 클라이언트가 연결을 만들어 창을 하나 표시하자마자, 안전을 위하여, 아래 명령으로 현재 열린 창 이후에 연결을 위한 허가를 무효로 한다:

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

당신은 호스트 확인을 아래 명령으로 불가능하게 할 수 있다:

```
light$ xhost +
```

이것은 호스트 억세스 확인을 불가능하게 하여 *누구*에게나 연결을 허락한다. 모든 이용자를 당신이 신뢰할 수 없는 네트워크(인터넷 같은)상에선 *결코* 이 명령을 내려선 안된다. 당신이 아래 명령으로 호스트 확인을 다시 가능하게 할 수 있다:

```
light$ xhost -
```

xhost - 그 자체는 억세스 리스트로부터 모든 호스트들을 제거*하지 않는다* (모두 제거하는 명령은 별로 쓸모 없을 것이다 - 당신은 어느 곳으로부터도 연결할 수 없을 것이다, 심지어 당신의 로컬 호스트로부터도 연결할 수 없을 것이다).

Xhost는 대단히 위태로운 방법이다. 원격 호스트에 여러 사용자들 관에 구분을 하지 않는다. 역시, 호스트 이름(실제 주소)은 눈속임을 당할 수 있다. 이것이 당신이 신뢰할 수 없는 네트워크 (예를 들어 인터넷에 이미 전화선을 이용한 PPP 억세스를 한 상태)상에 있다면 바람직하지 않다.

6.2 Xauth

Xauth는 올바른 열쇠를 아는 사람에게 억세스를 허락한다. 열쇠는 authorization record나 magic cookie로 불리는 것 따위이다. 이 인가 방법는 정식으로 MIT-MAGIC-COOKIE-1라 불린다.

여러 개의 디스플레이에 대한 쿠키들은 `~/.Xauthority`에 함께 저장한다. 당신의 `~/.Xauthority`은 그룹 구성원이나 다른 사용자들이 가까이하기 어려운 것임에 틀림없다. xauth 프로그램은 이 쿠키들을 관리한다, 여기서부터 이 방법은 약칭으로 xauth라 하겠다.

한 세션이 시작함과 동시에, 서버는 `-auth` 옵션이 가리키는 파일로부터 쿠키 하나를 읽는다. 그리고 나서, 서버는 동일한 쿠키를 숙지하고 있는 클라이언트로 부터의 연결만을 허락한다. `~/.Xauthority`에 쿠키가 바꾸었을 경우에, *서버는 바뀐 것을 손에 넣으려 하지 않을 것이다.*

서버는 클라이언트에게 몹시 분주하게 요구하는 쿠키를 결코 생성할 수 없다. 그렇지만 쿠키들은 서버 내부에 무사히 보존된다; 클라이언트가 서버에 쿠키들을 덮어쓰지 않는다면 쿠키들은 `~/.Xauthority`에서 없어지지 않는다. David Wiggins에 의하면:

> 당신이 관심을 가지고 있을지 모르는 앞선 묘안을 X11R6.3에 추가했다. 새로운 "보안" 확장에 의하여, 엑스 서버 자체가 몹시 분주하게 새로운 쿠키를 생성시키고 되돌릴 수 있다. 더군다나, 쿠키들은 "신뢰할 수 없다"고 지적될 수 있어서 그러한 쿠키들로 연결을 한 응용프로그램은 실행 중에 제지될 수 있다. 예를 들어, 신뢰할 수 없는 것들은 키보드/마우스 입력이나 윈도우 콘텐츠를 여러 신뢰성 있는 클라이언트들로부터 손에 넣을 수 없을 것이다. 안심하기 어렵다면, 웬만한 실력으로도 사용 가능한 새로운 "생성" 하부명령이 있다.

xauth는 xhost 사용상에서 명백한 보안상 이점을 가진다. 당신은 특정한 컴퓨터 상에 특정한 사용자로부터의 억세스를 제한할 수 있다. xauth는 xhost처럼 주소를 속이는 일에 고생하지 않는다. 그리고 당신이 원한다면, xauth가 연결을 허락한 다음에 xhost를 계속 사용할 수 있다.

6.2.1 쿠키 생성하기(Making the Cookie)

xauth를 사용하기 원한다면, 당신은 X server를 `-auth authfile` 옵션으로 시작해야 한다. 당신이 startx 스크립트를 사용한다면, 그 스크립트가 xauth를 사용하기 위한 적절한 장소이다. 당신의 startx 스크립트에 아래와 같이 authorization record를 만들어라.

`/usr/X11R6/bin/startx`로부터 발췌:

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

Mcookie는 리눅스-유틸 패키지(주요 사이트는 ftp://ftp.math.uio.no/pub/linux/) 속에 아주 작은 프로그램이다. 택할만한 것으로, 당신은 임의의 무작위 데이타(예를들어, /dev/urandom나 ps -axl같은데로부터)를 추출해서 쿠키 형태 속에 넣기 위해 md5sum을 사용할 수 있다:

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

당신이 startx 스크립트를 (root가 아니라서) 편집할 수 없다면, startx를 정확히 설정하기 위해 시스템 관리자 권한을 얻거나, 대신에 관리자가 xdm을 설정하게 하라. 관리자가 할 수 없었거나 하려고 하지 않는다면, 당신은 ~/.xserverrc 스크립트를 만들 수 있다. 당신이 이 스크립트를 가지고 있다면, xinit에 의해 실재 X server 대신에 이 스크립트가 실행된다. 그리고 나서 당신은 이 스크립트에서 적당한 옵션으로 실재 X server를 시작할 수 있다. 그렇게 하려면, 당신의 ~/.xserverrc가 쿠키 하나를 먼저 만들고나서 magic cookie 행을 실행하고 이어서 실재 X server를 실행하도록 해라:

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

당신이 당신의 X 세션을 관리하는 xdm을 사용한다면, 당신은 xauth를 쉽게 사용할 수 있다. /etc/X11/xdm/xdm-config에 DisplayManager.authDir 자원을 정의하라. Xdm은 X server가 시작할 때 X server에 -auth 옵션을 넘길 것이다. 당신이 이때 xdm에서 로그인을 했다면, xdm은 당신을 위해 당신의 ~/.Xauthority에 쿠키를 붙인다. 더 많은 정보를 얻으려면 xdm(1) 맨페이지를 보기 바란다. 예를 들면, 나의 /etc/X11/xdm/xdm-config은 내부에 다음 행들을 가지고 있다:

```
DisplayManager.authDir: /var/lib/xdm
```

6.2.2 쿠키 전달하기(Transporting the Cookie)

이제 막 당신은 서버 호스트 light.uni.verse에 당신의 X 세션을 시작하고 ~/.Xauthority에 당신의 쿠키를 얻었다, 당신은 클라이언트 호스트 dark.matt.er에 쿠키를 전해야 할 것이다.

당신의 홈 디렉토리가 밤낮으로 공유되어 있으면 가장 쉬운 경우이다. ~/.Xauthority 파일들은 단조롭다, 그래서 쿠키는 순간적으로 전달된다. 그러나, 붙들릴 수도 있다: 당신이 ~/.Xauthority에 :0에 대한 쿠키 하나를 붙일 때, dark 컴퓨터는 light컴퓨터를 위한 것이 아니고 dark컴퓨터를 위한 것으로 여길 것이다. 당신은 쿠키를 만들 때 뚜렷한 호스트 이름을 사용해야 한다; 당신은 그것을 무시할 수 없다. 당신은 :0과 light:0를 위한 쿠키를 같은 것으로 인스톨할 수 있다:

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

홈 디렉토리가 공유되어있지 않다면, 당신은 rsh(the remote shell)로 쿠키를 전달할 수 있다:

```
light$ xauth nlist :0 | rsh dark.matt.er xauth nmerge -
```

1. 당신의 로컬 ~/.Xauthority에서 쿠키를 빼낸다 (xauth nlist :0).
2. 쿠키를 dark.matt.er에 전달한다 (| rsh dark.matt.er).
3. 거기 ~/.Xauthority에 쿠키를 붙인다 (xauth nmerge -).

rsh가 당신을 위해 동작하지 않고 있는 경우도 있을 수 있다. rsh은 게다가, 보안상 약점(내 기억이 옳다면, 호스트 이름을 거짓으로 대답하는데 속을 수 있다)도 가지고 있다. 당신이 rsh를 사용할 수 없거나 바라지 않는다면, 당신은 다음과 같이 쿠키를 수동으로도 전달할 수 있다:

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

더 많은 정보를 얻으려면 역시 rsh(1)와 xauth(1x)의 맨페이지를 보기 바란다.

당신이 원격 호스트에 telnet 접속을 할 때 TERM이나 DISPLAY 변수 속에 쿠키를 같이 전달하는 것이 가능할지 모른다. 이것은 TERM 변수 내에 DISPLAY 변수를 같이 전달하는 것과 똑같은 방법이 통할 것이다. 5장 : 클라이언트 알려주기(Telling the Client)를 보아라. 나의 지침을 토대로 이 부분은 당신 자신의 힘으로 해보라, 그러나 나는 누군가가 이것을 확인이나 부정을 할 수 있는지 궁금하다.

6.2.3 쿠키 사용하기(Using the Cookie)

dark.matt.er상에, xfig같은 뛰어난, X 응용프로그램은 저절로 자신을 인증받기 위한 쿠키를 그 컴퓨터에 `~/.Xauthority`에서 조사해 볼 것이다.

6.3 Ssh

Authority record들은 암호화하지 않고 발송한다. 당신이 누군가가 당신의 연결을 엿보는 것을 걱정 해보았다면, ssh(the secure shell)을 사용하라. 암호화된 연결 상에서 X protocol 연결을 향상시킬 것이다. 게다가, 그 외에 좋은 점도 있다. 그 외에 좋은 점으로는 당신의 시스템에 좋은 구조상의 개선이 있다. 그냥 `http://www.cs.hut.fi/ssh/`, ssh 홈페이지를 방문해 보라.

인증 방법이나 암호화 X 연결에 관해서 이 밖에 다른 것을 아는 사람이 있겠는가? 아마 지옥을 지키는 개(Kerberos)가 알고 있을까?

제 7 절 문제 해결(Troubleshooting)

처음으로 여러분이 원격 엑스 윈도우 응용프로그램 실행을 시도했을 때, 대개는 응용프로그램이 실행되지 않는다. 여기에 약간의 흔한 에러 메시지들과 그것들의 원인, 의도대로 되도록 당신을 도울 수 있는 해결책이 있다.

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

DISPLAY 환경변수가 존재하지 않는 데다가, 당신은-display 플러그로도 응용프로그램에게 디스플레이를 알려주지 않았다. 응용프로그램 텅빈 문자열을 떠맡았지만, 구문상의 모순만 있었다. 이 문제를 해결하기 위해선, DISPLAY 환경변수 설정을 확실히 하라 (당신의 쉘에 따라 setenv나 export 명령으로).

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

에러 101은 "네트워크가 접근할 수 없다"는 상황이다. 응용프로그램은 서버에 네트워크 연결을 하지 못했다. 당신이 DISPLAY를 올바르게 설정했는지, 당신의 클라이언트에서 서버 컴퓨터로 접근이 가능한지 살펴보아라 (어쨌든 당신은 서버에 대개 로그인 되어있고 클라이언트에 telnet접속을 하고 있어야 한다).

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

에러 111은 "연결이 거절"된 상황이다. 당신이 연결을 시도한 서버 컴퓨터에 접근 할 수는 있지만, 지적한 엑스 서버가 거기에 존재하지 않는다. 당신이 올바른 서버 이름과 디스플레이 순번을 사용했는지 살펴보아라.

```
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
```

클라이언트는 서버에 연결할 수 있었지만, 서버는 클라이언트가 서버를 사용하는 것을 허락하지 않았다 (인증 받지 못했다). 당신은 클라이언트에 올바른 magic cookie를 전달하고, 쿠키가 만기가 안되도록 확실히 하라 (서버는 새로운 세션을 시작하면 새로운 쿠키를 사용한다).