

# **Nuxsl® – Network UniX Shell Logger**

Anleitung

<b>NUXSL® – NETWORK UNIX SHELL LOGGER.....</b>	<b>3</b>
<b>Zielsetzung .....</b>	<b>3</b>
<b>Installation .....</b>	<b>3</b>
<b>Kommunikation.....</b>	<b>3</b>
<b>Start .....</b>	<b>4</b>
<b>Administration .....</b>	<b>4</b>
<b>Konfigurationsfiles.....</b>	<b>4</b>
<b>Parameter .....</b>	<b>6</b>
nuxslma .....	6
nuxslsv.....	6
<b>Protokollierung .....</b>	<b>6</b>
<b>SSL PKI Installation.....</b>	<b>7</b>
Wurzelzertifikat.....	7
Zertifikat-Request für Master Server .....	7
Signiere Zertifikats-Request mit Wurzelzertifikat.....	8
Zertifikat-Request für nuxslsv Server .....	8
Signiere Zertifikats-Request mit Wurzelzertifikat.....	9
Vorteile.....	9
<b>Auditing.....</b>	<b>11</b>
Nuxslgui Benutzung .....	11
Session-Flow:.....	11
Sessionauswahl .....	12
Session-Übersicht.....	12
Aktive Sessions.....	13
<b>Übersicht.....</b>	<b>14</b>
Prozesse.....	14
Dateipfade .....	14
<b>Kontakt.....</b>	<b>14</b>

## Nuxsl® – Network UniX Shell Logger

### *Zielsetzung*

Nuxsl dient zur zentralen, revisionskonformen Protokollierung textbasierender Benutzereingaben in Shellumgebungen oder Anwendungen.

Die Kommunikation ist hinsichtlich Sicherheit und Netzwerктаuglichkeit optimiert, z.B. wird nur ein einziger definierter und bei IANA registrierter TCP-Port 5991 benutzt.

Zur Entwicklungszeit wird mittels geeigneter Tools auf memoryleaks und Pufferüberläufe geprüft.

Die einzelnen Daemonen bewirken eine minimale Last auf den jeweiligen Server. Mit nur wenigen kByte memoryfootprint sind sie auf Effizienz optimiert.

### *Installation*

Für die größtmögliche Transparenz der Installation wurde auf ein eigenständiges Setupprogramm verzichtet. Es werden für die meisten Unix-Varianten komprimierte TAR Archive (nuxsl1.0-\*) zur Verfügung gestellt. Diese sollten nach /usr/local entpackt werden.

Getrennt von den eigentlichen Daemonen werden je nach Betriebssystem noch einzelne Laufzeitbibliotheken benötigt. Sollten diese nicht zum Standardumfang des Systems gehören, befinden sie sich im zusätzlichen Paket (libs-\*), welches ebenfalls nach /usr/local entpackt wird.

Für den zentralen DB Server wird im Paket nuxsl-install.tar.gz das benötigte MySQL Datenbank-Schema, sowie ein optionales Installations-Script geliefert. Für die Oracle-Version wird eine geeignete Oracle 8i Installation vorausgesetzt, vorzugsweise mit einem eigenen Tablespace für den nuxsl-Benutzer. Das Oracle-Schema ist im Downloadbereich als SQL-Batch herunterladbar.

Die graphische Administration kommt ebenfalls in 2 Paketen, der eigentlichen Anwendung (nuxslgui1.0-\*), sowie den Laufzeitbibliotheken (nuxslguilib-\*). Die Anwendung ist in Versionen für MySQL, sowie für Oracle8i verfügbar.

Für eine sichere Kommunikation zwischen den Servern wird eine SSL-Verbindung genutzt, die nötigen Zertifikate hierfür sind in Abstimmung mit der lokalen PKI zu erzeugen. Für einen Probetrieb kann auch eine unverschlüsselte Kommunikation per Kommandozeile (-u) gewählt werden.

### *Kommunikation*

Die Kommunikation findet zwischen drei verschiedenen Programmschichten statt, welche durch die Prozesse NUXSLCL, NUXSLSV und NUXSLMA bedient werden.

Der zentrale abgesicherte Datenbankserver empfängt über den Prozess NUXSLMA die eingehenden Verbindungs- und Protokolldaten der einzelnen zu protokollierenden Server.

Für jeden Unix-Server wird ein lokaler NUXSLSV Serverprozess benötigt. Dieser hält während seiner gesamten Lebensdauer eine einzelne dedizierte TCP Verbindung zum sog. Nuxsl-Master (NUXSLMA) aufrecht.

Als transparente Pre-Loginshell wird für die zu protokollierenden Nutzer der Client NUXSLCL konfiguriert. Dieser nuxslcl läuft pro aktivem Nutzer auf einem Unix-Server und kommuniziert mit seinem lokalen nuxslsv Prozess.

Desweiteren wird auf dem zentralen DB-Server der NUXSLCR Prozess in kurzen, regelmäßigen Abständen via cron job die Daten kompaktieren und abgeschlossene Sessions einsammeln.

## ***Start***

Die logische Startreihenfolge der beteiligten Dienste:

- 1) MySQL aktivieren
- 2) Nuxslma einmalig zentral
- 3) Nuxslsv pro Server
- 4) Nuxslcl beim Anmelden pro Server/Nutzer

Die Daemonen sind derart geschrieben, daß ein Wiederaufsetzen der Kommunikation bei Abbruch einer Verbindung in regelmäßigen Intervallen probiert wird. In der Zwischenzeit lokal anfallende Protokollierungen werden in verschlüsselter Form zwischengespeichert.

Die Prozesse benötigen zur Laufzeit keine root Rechte. Da die Kommunikation über einen Port >1024 stattfindet, empfiehlt sich die Einrichtung eines dedizierten nuxsl Users. Dieser benötigt daraufhin nur Leserechte an den Dateien nuxsl.conf und nuxsl.lic, sowie Schreibrechte an einer lokalen Named-Pipe /tmp/nuxsl.pipe.

Beim Verbindungsaufbau eines Servers mit dem Master werden die lokalen Benutzer- und Gruppeninformationen des Servers in den Master übertragen.

## ***Administration***

Zur Einsicht in die protokollierten Daten wird das NUXSLGUI verwendet. Diese X-Windows Anwendung ermöglicht eine effektive Suche nach abgeschlossenen Sessions und deren Text-ein und -Ausgaben, sowie den Export der Daten.

Die lokal zum DB Server befindliche Installation kann ggf. via SSH-Tunnel auch von entfernten Arbeitsplätzen sicher genutzt werden.

Die Anmeldung des Frontends zu Datenbank hängt von deren version ab. In der MySQL Version werden die login-daten aus der nuxsl.conf Datei übernommen. Bei der Oracle Version wird eine Dialogbox beim Programmstart nach Benutzername, Passwort, sowie Connectstring fragen. Der Connectstring besteht aus drei mit Doppelpunkt getrennten Parametern: hostname:portnummer:oracleSID

Der NUXSLCR Daemon sollte in den DB-Server crondaemon aufgenommen werden mit einem Laufzeitintervall von 5 Minuten.

## ***Konfigurationsfiles***

Alle Programme suchen nach der Konfigurationsdatei nuxsl.conf, zuerst im aktuellen Pfad, dann in /etc. Die gleiche Suchfolge wird für die jeweils pro Server benötigte Lizenzdatei nuxsl.lic genutzt.

Im einzelnen werden diese Parameter im Abschnitt „[nuxsl]“ erwartet:

```
- DBHost
MySQL DB-Servername oder IP Nummer
- DBname
MySQL Datenbankname, bzw Oracle Net8 Name
- DBuser
DB Benutzername (benötigt werden die Rechte SELECT, INSERT, UPDATE und
DELETE)
- DBpasswd
DB Benutzerpasswort
- SHELL
Effektive Benutzershell der NUXSLCL Nutzer
- PRIVKEY
Absoluter Dateipfad des lokalen Serverschlüssels
- PRIVPWD
Passwort des lokalen privaten Schlüsselteils
- CERT
Absoluter Dateipfad des vom Wurzelzertifikat signierten Schlüssels
- ROOTCERT
Absoluter Dateipfad des allgemeinen öffentlichen Wurzelzertifikates
```

**Die Datei nuxsl.conf sollte nur für den jeweiligen nuxsl User lesbar sein.**

**Beispiel für nuxsl.conf:**

```
# config file for nuxsl
[ Nuxsl]
DBhost=localhost
DBname=logger
DBuser=nuxsl
DBpasswd=nuxsl

SHELL=/bin/sh

PRIVKEY=/opt/ssl/serverkey.pem
PRIVPWD=Server
CERT=/opt/ssl/servercert.pem
ROOTCERT=/opt/ssl/nuxslCA/CAcert.pem
```

**Beispiel für nuxsl.lic:**

```
# Don't edit this file, you might destroy the licence checksum
IP=192.168.0.1
OS=Linux
VER=1.0
CONTACT=admin@think-safety.de
KEY=242976fced6288327bba57a01a35ff30
```

## Parameter

Die Programme besitzen folgende Steuermöglichkeiten über Kommandozeilenparameter:

### nuxslma

```
-V      Ausgabe der Versionsinformationen und Beendigung
-i      Interaktiver Modus, keine Daemonisierung/Fork
-u      Abschalten von SSL (muß bei allen beteiligten Prozessen gleich
        konfiguriert sein)
-?      Kurze Hilfe
```

### nuxslsv

```
-V      Ausgabe der Versionsinformationen und Beendigung
-i      Interaktiver Modus, keine Daemonisierung/Fork
-u      Abschalten von SSL (muß bei allen beteiligten Prozessen gleich
        konfiguriert sein)
-m master  Servername oder IP des zentralen NUXSLMA Servers
-?      Kurze Hilfe
```

## Protokollierung

Alle Prozesse protokollieren über die lokalen Syslog-Einstellungen (Syslog facility: log\_daemon mit den Prioritäten log\_debug, log\_info bzw. log\_error) für mögliche Fehler und andere Laufzeitinformationen.

Beispiel Protokollierung von nuxslcr:

```
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : starting nuxsl cron
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : search file ./nuxsl.conf
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : search file /etc/nuxsl.conf
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read host localhost
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read name logger
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read user nuxsl
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read passwd XXX
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read shell /bin/sh
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read keyfile
/opt/ssl/serverkey.pem
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read key passwd XXX
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read certfile
/opt/ssl/servercert.pem
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : config read rootcertfile
/opt/ssl/Cacert.pem
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : doExpire
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : doPurge
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : doArchive
Aug 27 17:00:00 fix Nuxsl-cr[ 23200] : ending nuxsl cron
```

## SSL PKI Installation

Für das erstmalige Einrichten der SSL Verbindungen kann nach folgendem Beispiel vorgegangen werden (Nähere Infos gibt es auch bei [www.openssl.org](http://www.openssl.org)).

Auf jedem Server muß zu dem in nuxsl.conf angegebenen Pfad (ROOTCERT) der öffentliche Schlüssel des Wurzelzertifikats abgelegt werden. Außerdem benötigt jeder Server sein mit dem Wurzelzertifikat signiertes Schlüsselpaar (PRIVKEY und CERT).

Zur Zertifikatskettenerzeugung kann ein zentraler OpenSSL Server genutzt werden, von dem aus die erzeugten Zertifikate über einen gesicherten Kanal (SSH) auf die einzelnen nuxsl Server verteilt werden.

Für eine einfachere Ersterstellung einer Menge von Zertifikaten kann das Tool nuxsl-sslgui verwendet werden, welches separat im Downloadbereich der nuxsl Website verfügbar ist. Das Programm besteht aus 2 rpm-Archiven für Linux mit der Anwendung und den Laufzeitbibliotheken.

Eine manuelle Erstellung bietet den meisten Einfluß auf die erzeugten Zertifikate:

### Wurzelzertifikat

Einmaliges Erzeugen des Wurzelzertifikates auf einem Server. Der öffentliche Anteil wird über alle beteiligten nuxsl Server verteilt.

```
#> openssl req -new -x509 -keyout nuxslCA/private/CAkey.pem -out
nuxslCA/private/CAcert.perm -config openssl.cnf
Using configuration from openssl.cnf
Generating a 768 bit RSA private key
.....++++++
.....++++++
writing new private key to 'nuxslCA/private/CAkey.pem'
Enter PEM pass phrase: Master
Verifying password - Enter PEM pass phrase: Master
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [ DE] :
State or Province Name (full name) [ Hessen] :
Locality Name (eg, city) [ Friedrichsdorf] :
Organization Name (eg, company) [ security-gui.de GbR] :
Organizational Unit Name (eg, section) [ ]:nuxsl CA
Common Name (eg, YOUR name) [ ]:nuxsl CA
Email Address [ ca-admin@think-safety.de] :
```

### Zertifikat-Request für Master Server

Der nuxslma Server benötigt für die Authentifizierung gegenüber den nuxslsv Servern einen eigenen Schlüssel und die Signatur mit dem oben erzeugten Wurzelzertifikat.

```
#> openssl req -new -keyout serverkey.pem -out serverreq.pem -days 360 -
config openssl.cnf
Using configuration from openssl.cnf
Generating a 768 bit RSA private key
.....++++++
.+++++++
writing new private key to 'serverkey.pem'
Enter PEM pass phrase: Server
Verifying password - Enter PEM pass phrase: Server
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [ DE ] :
State or Province Name (full name) [ Hessen ] :
Locality Name (eg, city) [ Friedrichsdorf ] :
Organization Name (eg, company) [ security-gui.de GbR ] :
Organizational Unit Name (eg, section) [ ] : nuxsl master
Common Name (eg, YOUR name) [ ] : the.local.servername
Email Address [ admin@think-safety.de ] :

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password [ ] :
An optional company name [ ] :

*** Combined file ***
#> cat serverreq.pem serverkey.pem >new.pem
```

## Signiere Zertifikats-Request mit Wurzelzertifikat

```
#> openssl ca -policy policy_anything -out servercert.pem -config
openssl.cnf -infile new.pem
Using configuration from openssl.cnf
Enter PEM pass phrase: Master
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName          :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Hessen'
localityName         :PRINTABLE:'Friedrichsdorf'
organizationName     :PRINTABLE:'security-gui.de GbR'
organizationalUnitName:PRINTABLE:'nuxsl master'
commonName           :PRINTABLE:'the.local.servername'
emailAddress         :IA5STRING:'admin@think-safety.de'
Certificate is to be certified until Mar 31 15:30:02 2003 GMT (365 days)
Sign the certificate? [ y/n ] : y

1 out of 1 certificate requests certified, commit? [ y/n ] y
Write out database with 1 new entries
Data Base Updated
```

## Zertifikat-Request für nuxslsv Server

Der nuxslsv Server benötigt für die Authentifizierung gegenüber dem nuxslma Master Server einen eigenen Schlüssel und die Signatur mit dem oben erzeugten Wurzelzertifikat.

```
#> openssl req -new -keyout serverkey.pem -out serverreq.pem -days 360 -
config openssl.cnf
Using configuration from openssl.cnf
Generating a 768 bit RSA private key
.....+++++++
.+++++++
writing new private key to 'serverkey.pem'
Enter PEM pass phrase: Server
Verifying password - Enter PEM pass phrase: Server
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [ DE ] :
State or Province Name (full name) [ Hessen ] :
Locality Name (eg, city) [ Friedrichsdorf ] :
Organization Name (eg, company) [ security-gui.de GbR ] :
Organizational Unit Name (eg, section) [ ]:nuxsl server
Common Name (eg, YOUR name) [ ]:the.local.servername
Email Address [ admin@think-safety.de ] :

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password [ ] :
An optional company name [ ] :

*** Combined file ***
#> cat serverreq.pem serverkey.pem >new.pem
```

## Signiere Zertifikats-Request mit Wurzelzertifikat

```
#> openssl ca -policy policy_anything -out servercert.pem -config
openssl.cnf -infile new.pem
Using configuration from openssl.cnf
Enter PEM pass phrase: Master
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName          :PRINTABLE:'DE'
stateOrProvinceName  :PRINTABLE:'Hessen'
localityName         :PRINTABLE:'Friedrichsdorf'
organizationName     :PRINTABLE:'security-gui.de GbR'
organizationalUnitName:PRINTABLE:'nuxsl server'
commonName           :PRINTABLE:'the.local.servername'
emailAddress         :IA5STRING:'admin@think-safety.de'
Certificate is to be certified until Mar 31 15:30:02 2003 GMT (365 days)
Sign the certificate? [ y/n ]:y

1 out of 1 certificate requests certified, commit? [ y/n ]y
Write out database with 1 new entries
Data Base Updated
```

## Vorteile

Durch die Nutzung der Zertifikatskette kann ein beiderseitiges Vertrauenskonzept etabliert werden:

Bei dem Verbindungsaufbau eines Clients mit dem Server überprüfen beide Seiten die Gültigkeit der Zertifikate. Dadurch können vertrauliche Protokolldaten eines Clients nicht via einer Man-In-The-Middle Attacke abgehört werden, ebenso kann kein fremder Client den protokollierenden Server mit unnützen Daten belasten oder sich als ein anderer Client ausgeben.

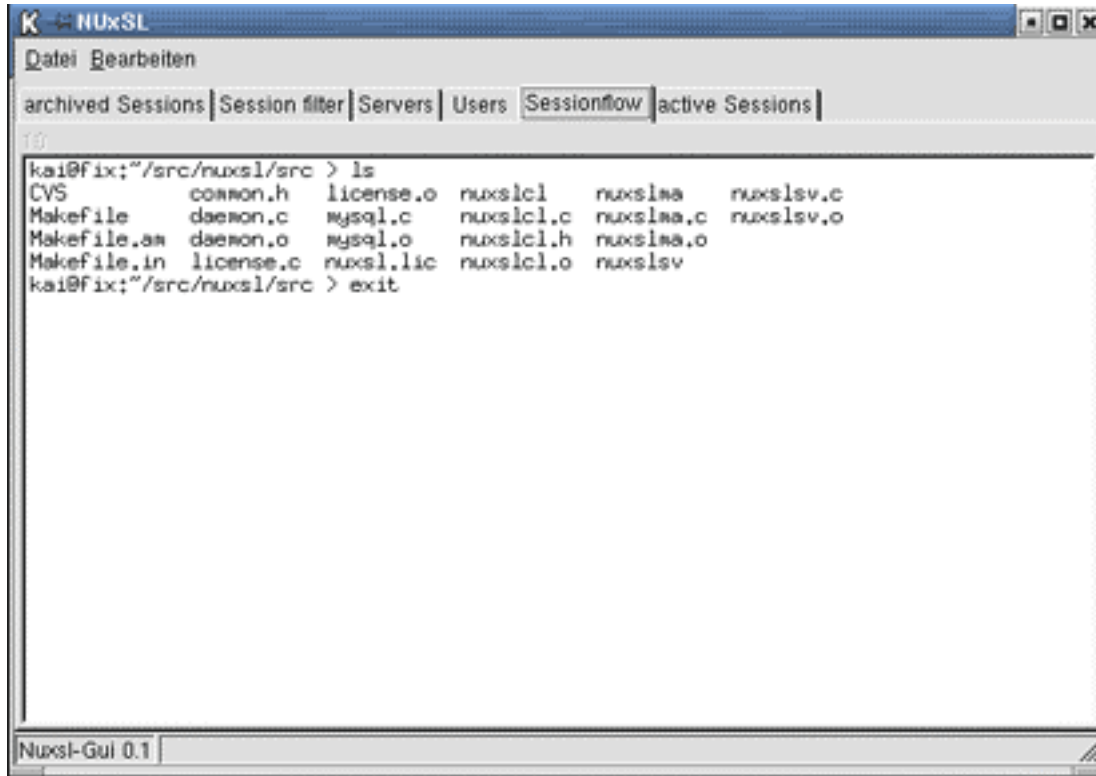
Nicht zuletzt sind dadurch auch alle Daten während der Übertragung verschlüsselt und können so weder manipuliert noch analysiert werden.

## Auditing

### Nuxslgui Benutzung

Die Ausgabe einer protokollierten Session via nuxslgui ist über ein Standard Textfeld mit den üblichen Funktionen für cut&paste verfügbar:

#### Session-Flow:



The screenshot shows a window titled "K - NUXSL" with a menu bar "Datei Bearbeiten". Below the menu bar is a tabbed interface with tabs for "archived Sessions", "Session filter", "Servers", "Users", "Sessionflow" (selected), and "active Sessions". The main area displays a terminal session:

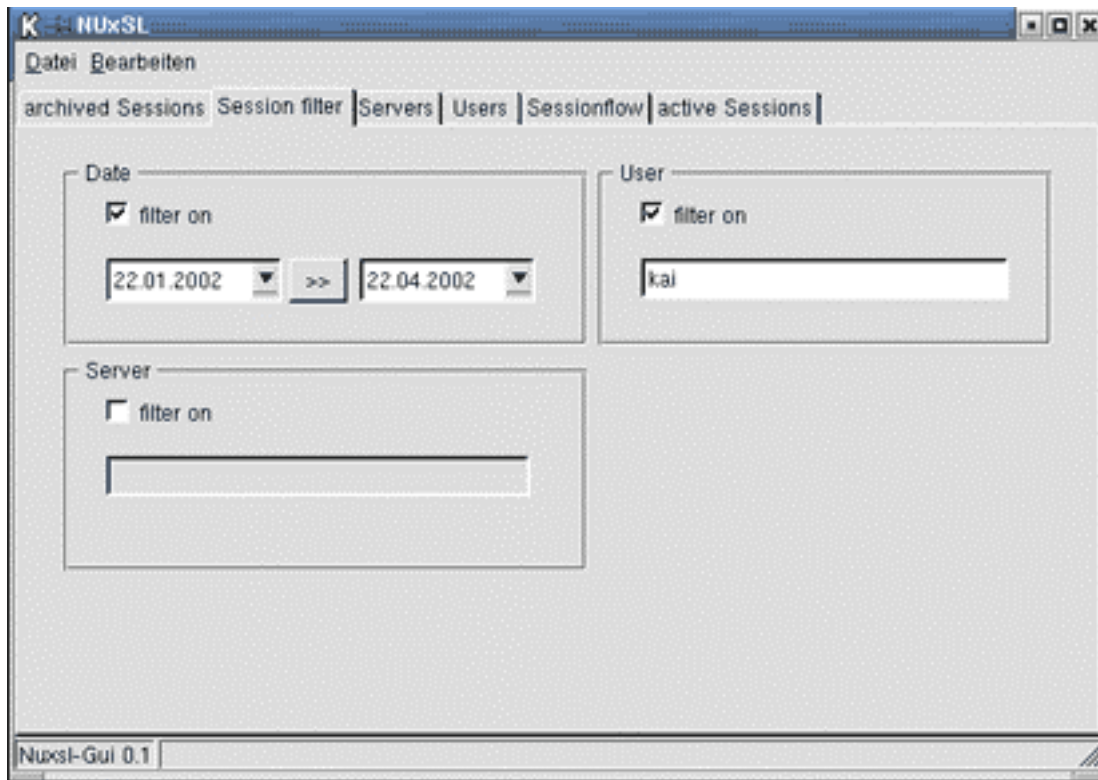
```
kai@fix:~/src/nuxsl/src > ls
CVS          common.h    license.o  nuxslcl   nuxslwa   nuxslsv.c
Makefile     daemon.c   mysql.c    nuxslcl.c nuxslwa.c nuxslsv.o
Makefile.am  daemon.o   mysql.o    nuxslcl.h nuxslwa.o
Makefile.in  license.c  nuxsl.lic nuxslcl.o nuxslsv

kai@fix:~/src/nuxsl/src > exit
```

At the bottom of the window, the text "Nuxsl-Gui 0.1" is visible.

## Sessionauswahl

Die Liste der Sessions kann über Selektion des Zeitbereiches, des Servers oder der Nutzer auf ein überschaubares Maß eingeschränkt werden.:



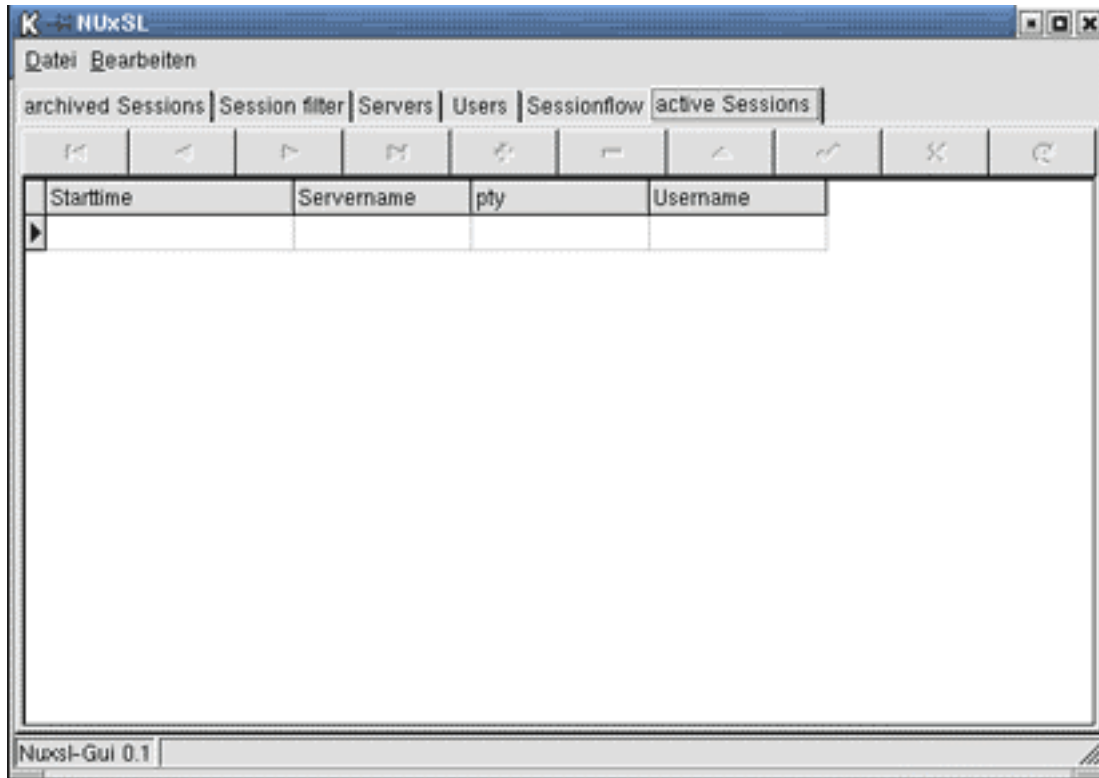
## Session-Übersicht

Alle bislang aufgelaufenen Sitzungen mit Angaben über Zeit, Server und Nutzer:

Starttime	Username	Servername	pty	sessionid
23.01.2002 19:17:04	kai	localhost	/dev/pts/6	10
23.01.2002 19:37:50	kai	localhost	/dev/pts/4	11
23.01.2002 20:09:44	kai	localhost	/dev/pts/6	12
23.01.2002 20:10:40	kai	localhost	/dev/pts/6	13
23.01.2002 20:13:48	kai	localhost	/dev/pts/6	14
23.01.2002 20:16:29	kai	localhost	/dev/pts/6	15
23.01.2002 20:32:14	kai	localhost	/dev/pts/8	16
27.01.2002 18:24:03	kai	localhost	/dev/pts/10	17
28.01.2002 21:38:30	kai	localhost	/dev/pts/2	18
01.02.2002 17:16:46		fussel	/dev/tty0	19
01.02.2002 17:22:21		fussel	/dev/tty0	20
02.02.2002 20:59:57	kai	localhost	/dev/pts/2	21
02.02.2002 21:08:12	kai	localhost	/dev/pts/2	22
04.02.2002 18:17:41	kai	localhost	/dev/pts/5	23

## Aktive Sessions

Noch offene Sitzungen bzw. gerade abgemeldete Nutzer vor dem nächsten NUXSLCR Lauf:



## Übersicht

### Prozesse

nuxslma	Zentraler Master-Server, hat direkte Verbindung zur MySQL Datenbank
nuxslsv	Einer pro Server, verbindet sich direkt mit dem Master
nuxslcl	Eine Instanz pro angemeldetem Nutzer, verbindet sich mit dem lokalen nuxslsv
nuxslcr	Auf dem gleichen Server wie nuxslma über den crond regelmäßig gestartet

### Dateipfade

/etc/nuxsl.conf	Konfigurationsdatei für nuxslma, nuxslsv und nuxslcr
/etc/nuxsl.lic	Lizenzdatei für nuxslma und nuxslsv
/tmp/nuxsl.pipe	IPC Kommunikationsdatei zwischen nuxslcl's und nuxslsv
/.../ssl/serverkey.pem	Privater Schlüssel des nuxslsv oder nuxslma Servers
/.../ssl/servercert.pem	Mit dem Wurzelzertifikat signierter Zertifikatsrequest des Serverschlüssels
/.../ssl/CAcert.pem	Auf allen Servern das selbe Wurzelzertifikat

### Kontakt

Security-gui.de GbR  
 Kiesweg 54  
 D-35396 Giessen  
 Tel.: 0700-secguide  
 eMail: [info@security-gui.de](mailto:info@security-gui.de)

Weitere Infos und ein Supportforum finden sich unter <http://www.nuxsl.org/phpbb/>