xinetd – Der Internetdämon

- Kurzakte –

Installation	
• xinetd aus Paketgruppe Productivity/Networking/System (nach /usr/sbin/xinetd)	
Konfiguration	
/etc/xinetd.conf	• Hauptkonfigurationsdatei (siehe unten)
/etc/xinetd.d/	• Verzeichnis für die Konfigurationsdateien der einzelnen Dienste
/usr/sbin/xconv.pl	• konvertiert alte inetd.conf in neue xinetd.conf
/etc/services	• Zuordnung von passenden symbolischen Namen zu Portnummern
Start als Stand-Alone-Dienst	
/etc/init.d/xinetd	• start → startet xinetd manuell • stop → stoppt xinetd • restart → Neustart • status → Statusabfrage (Checking for xinetd: OK)
kill –HUP `cat /var/run/xinetd.pid`	• Neueinlesen der Konfigurationsdatei
Dokumentation	
man: xinetd(8), xinetd.conf(5), xinetd.log(5), tcpd(8), inetd(8)	

/etc/xinetd.conf

Enthält unter SuSE nur die default-Sektion für die Voreinstellungen:
`default`
`{`
`<Schlüssel> <Operator> <Parameter>`
`<Parameter> ...`
`}`
Die Konfiguration für die jeweiligen Netzwerkdienste liegt in eigenen externen Dateien und wird eingebunden mit `includedir /etc/xinetd.d`

Dienstspezifische Konfigurationsdateien unter /etc/xinetd.d/

Der Name des Dienstes wird in /etc/services einem Port zugeordnet.
`service <Service-Name>`
`{`
`<Schlüssel> <Operatoren> <Parameter>`
`<Parameter> ...`
`}`
Für Zugriffsbeschränkungen stehen die Schlüssel *only_from*, *no_access*, *access_times* und *disabled* zur Verfügung.

Operatoren, mit denen Attributen Werte zugewiesen werden: = Setzt einen Wert für ein Attribut. Die meisten Attribute lassen nur diesen Wert zu += Fügt einem Attribut einen Wert zu den bereits bestehenden Werten hinzu. -= Entfernt einem Attribut den angegebenen Wert.

/etc/xinetd.conf

```
defaults
{
        log_type        = FILE /var/log/xinetd.log
        log_on_success  = HOST PID USERID DURATION EXIT
        log_on_failure  = HOST USERID ATTEMPT RECORD
        only_from       = localhost
        instances       = 30            # gegen DOS
        cps             = 50 10         # gegen DOS
        per_source      = 2             # gegen DOS
        max_load        = 3.0           # gegen DOS
        interface       = 127.0.0.1

            disabled = shell login exec comsat
            disabled += telnet ftp
            disabled += name uucp tftp
            disabled += finger systat netstat
}
```

Dienstspezifische Konfigurationsdatei

```
service ftp
  {
    socket_type  = stream
    wait         = no
    user         = root
    server       = /usr/sbin/in.ftpd
    server_args  = -l
    instances    = 4
    access_times = 7:00-12:30 13:30-21:00
    only_from    = 192.168.1.0/24
    only-from    += 192.168.200.3 192.168.200.7 192.168.200.9
    only-from    += 192.168.200.10 192.168.200.12
    no_access    = haxor.evil.org
    nice         = 10
    # disable    = yes
 }
```

In der `defaults`-Section wird das Attribut `instances` mit dem Wert 30 definiert. Damit können nicht mehr als 15 Dienste gleichzeitig gestartet werden. In der Definition des FTP-Dienstes steht der Wert auf 4. In diesem Fall bezieht er sich auf die Menge der gleichzeitigen FTP-Verbindungen.

Die Angabe `log_type = FILE /var/log/xinetd.log` bestimmt, daß nicht der **syslogd** benutzt werden soll, sondern **xinetd** direkt in die angegebene Datei schreibt. Anstatt diesem Eintrag wäre auch möglich gewesen:

```
log_type = SYSLOG daemon info
```

Dann wären alle Meldungen über den Syslog-Daemon geschrieben worden und zwar mit der Herkunft `daemon` und der Priorität `info`.

Die beiden Angaben `log_on_success` und `log_on_failure` legen fest, was alles in das Logbuch aufgenommen werden soll, wenn ein Zugriff erfolgreich war oder abgelehnt wurde.
Die Angabe `only_from = 192.0.0.0/8` bestimmt, dass grundsätzlich (wenn beim entsprechenden Dienst nichts anderes angegeben wurde) nur die Rechner Zugriff haben, die auf die angegebene Adresse passen. Das `/8` bestimmt, dass nur die ersten 8 Bit der Adresse gewertet werden. Im Beispiel werden also alle Rechner Zugriff bekommen, deren erstes Adressenbyte 192 ist.

Die folgenden Zeilen der `default section` schalten die angegebenen Dienste mit dem Attribut `disabled` komplett ab.

Im Abschnitt FTP werden jetzt die einzelnen Attribute für den Dienst gesetzt. Interessant sind Angaben über die erlaubten Tageszeiten, in denen der Dienst zur Verfügung steht und der gesetzte NICE-Wert, unter dem der Daemon laufen soll. Zugreifen dürfen nur Rechner, deren Adresse mit 192.168.1 beginnt, nicht aber haxor.

Attribut	Werte und Beschreibung
`cps`	Beschränkt die Zahl der zugelassenen Verbindungen. Als erster Parameter wird die Zahl selbst angeben. Der zweite Wert legt die Zeit fest, die der Dienst für weitere Verbindungen erreichbar ist, sobald das Limit erreicht wurde.
`Flags`	• IDONLY : es werden nur Verbindungen mit Client-Rechnern zugelassen, auf denen ein Server zur Identifikation läuft (z. B.: identd) • NORETRY : unterbindet das Anlegen eines neuen Prozesses im Fehlerfall
`instances`	Maximale Zahl an Servern des gleichen Typs, die simultan laufen dürfen
`log_on_failure`	Hierdurch kann eine Vielzahl an Informationen protokolliert werden, sobald ein Server nicht gestartet werden kann, sei es aufgrund fehlender Ressourcen oder der Verletzung der Zugriffsregeln: • HOST, USERID : wie oben • ATTEMPT : hält einen Zugriffsversuch fest. Diese Option wird automatisch gewählt, sobald ein weiterer Wert gesetzt wird • RECORD : protokolliert alle Informationen, die über den Client verfügbar sind
`log_on_success`	Bei Start eines Servers können mehrere unterschiedliche Informationen festgehalten werden: • PID : die Server PID (handelt es sich um einen internen Dienst von `xinetd` ist die PID 0) • HOST : Die Adresse des Clients • USERID : die ID des Benutzers, der den Dienst in Anspruch nimmt, entsprechend dem in der RFC1413 definierten Protokoll (Flag auf IDONLY; identd muss auf Client laufen) • EXIT : Der Statuswert, den der Prozess bei Beendigung zurückliefert • DURATION : Verbindungsdauer
`log_type`	xinetd verwendet `syslogd`, als Selektor wird standardmäßig `daemon.info` gewählt. • SYSLOG Selektor [level] : Auswahl zwischen `daemon`, `auth`, `user` oder `local0-7` des `syslogd` Daemons. • FILE [max_size [absolute_max_size]] : In die hier angegebene Datei wird die Ausgabe gesichert. Die beiden Optionen begrenzen die Dateigröße. Wird der erste Wert erreicht, wird eine Nachricht an `syslogd` gesendet. Die Protokollierung des jeweiligen Dienstes wird eingestellt, sobald die Dateigröße den zweiten Parameter erreicht (sollte es sich um eine mehrfach verwandte Datei oder um eine Standardeinstellung in `xinetd.conf` handeln, können auch mehrere Dienste betroffen sein).
`max_load`	Tatsächliche maximale Last eines Servers (z.B. 2 oder 2.5). Wird dieser Wert überschritten, nimmt der Server keine weiteren Anfragen an
`nice`	Ändert die Priorität des Servers, analog dem Unixbefehl `nice`.
`no_access`	Liste der Clients, denen der Zugriff auf den jeweiligen Dienst verweigert werden soll
`only_from`	Liste aller akzeptierten Rechner. Wird diesem Attribut kein Wert zugewiesen, so ist **kein** Zugriff auf den Dienst möglich, aber es wird ein möglicher unautorisierter Zugriff protokolliert!!!!
per_source	Ein Zahlenwert oder `UNLIMITED`. Ermöglicht es, die Zahl der Verbindungen, die ein Dienst mit einem einzelnen Clientrechner unterhält, zu beschränken.
`port`	Port, unter dem der Dienst erreichbar ist. Sollte dieser auch in der Datei `/etc/services` stehen, müssen beide Wert übereinstimmen.
`protocol`	Das hier angegebene Protokoll muss in der Datei `/etc/protocols` aufgeführt sein. Wird dieses Attribut nicht gesetzt, wird das Standardprotokoll des Servers verwendete
`server`	Pfad, unter dem das Serverprogramm zu finden ist
`server_args`	Parameter, die an den Server übergeben werden sollen
`socket_type`	`stream` (TCP), `dgram` (UDP), `raw` (IP Direktzugriff) oder `seqpacket ()`
`type`	xinetd verwaltet drei Arten von Diensten: 1. RPC : alle, die in der Datei `/etc/rpc` zu finden sind...funktioniert nicht allzu gut 2. INTERNAL :Dienste, die direkt von xinetd verwaltet werden (`echo`, `time`, daytime, `chargen` und `discard`) 3. UNLISTED : Dienste, die weder in `/etc/rpc` noch in `etc/services` stehen.
`user`	User-ID, unter der der Dienst laufen muss
`wait`	Legt das threading Verhalten des Dienstes fest. Es gibt zwei Möglichkeiten: • `yes` : der Dienst ist mono-threaded, dass heißt es kann immer nur eine Verbindungen dieses Types unterhalten werden (wichtig bei verbindungslosen Diensten wie UDP) • `no` : für jede neue Anfrage startet `xinetd` einen neuen parallelen Serverprozess, unter Berücksichtigung etwaiger Beschränkungen der maximalen Zahl an Servern (Multithraeding)

Signalbehandlung des xinetd
Zum Neueinlesen der Konfigurationsdatei sollten nicht die Befehle kill -1 pid oder kill -HUP pid genutzt werden, da der Daemon aus Sicherheitsgründen darauf mit einem schweren Fehler und der Erstellung eines Speicherabzugs reagiert. Man muss die Signale SIGUSR1 oder SIGUSR2 schicken.
SIGUSR1 xinetd liest die Konfigurationsdatei neu ein und paßt seine Aktionen entsprechend an; gerade laufende Netzwerkdienste bleiben davon unberührt. SIGUSR2 xinetd liest ebenfalls seine Konfigurationsdatei neu ein, jedoch werden Netzwerkdienste, die aus der Konfiguration genommen wurden, sofort beendet. Noch laufende Netzwerkdienste werden auf evtl. neu definierte Zugriffsrechte, Anzahl der Daemonen eines Services etc. überprüft und ggf. beendet. SIGQUIT beendet xinetd. SIGTERM beendet alle laufenden Services, dann erst den xinetd. SIGHUP Erstellung eines Speicherabzuges SIGIOT veranlasst xinetd, eine interne Konsistenzprüfung vorzunehmen, um sicherzustellen, dass das Programm nicht verfälscht wurde.

Dienst deaktivieren
disable = yes