

# Remote Administration von Windows Servern mit Microsoft Terminal Services und OpenSSH

von Dominick Baier ([dbaier@ernw.de](mailto:dbaier@ernw.de)) und  
Jens Franke ([jfranke@ernw.de](mailto:jfranke@ernw.de))

## 1 Einleitung

Dieses Dokument behandelt die flexible Remote-Administration von Windows Servern mit SSH und Microsoft Terminal Services.

### 1.1 Ziel

Es soll ein universelles Remote-Administrations-Konzept für Windows Server entwickelt werden.

Folgende Anforderungen sollen hierbei berücksichtigt werden:

- Kommandozeilen-orientierte Remote Administration
- Grafische Remote-Administration
- Der Terminal Service Port (3389) soll nicht von außen ansprechbar sein
- Maximale Software-basierende Sicherheit soll gewährleistet sein
- Es sollen möglichst viele Client-Plattformen mit diesem Konzept abgedeckt werden

## 2 Lösung

### 2.1 Terminal Services

Microsoft Terminal Services ist eine Standard-Komponente von Windows 2000, XP und Windows Server 2003 zur grafischen Remote-Administration.

Terminal Services können in zwei Ausprägungen installiert werden.

#### **Application Mode**

Der Application Mode lässt eine theoretisch unbegrenzte Anzahl von gleichzeitigen Benutzern zu. Im Application Mode ist für jeden Benutzer eine Lizenz notwendig

#### **Remote Admin Mode**

Dieser Modus erlaubt zwei gleichzeitig angemeldete Benutzer und ist Lizenz-frei. Somit ist dies der Modus der Wahl zur entfernten Administration von Windows Servern.

In diesem Dokument wird lediglich der Remote-Admin Modus betrachtet.

### 2.2 SSH

SSH ist eine Kommandozeilen-basierende Remote-Shell. SSH unterstützt starke Verschlüsselung und Client Zertifikate. SSH wird im Allgemeinen als „sicher“ angesehen.

SSH Clients gibt es für nahezu jede Plattform und gehören zum Standard-Lieferumfang jedes Unix basierenden Betriebssystems. In diesem Dokument wird die Cygwin basierende OpenSSH Windows Portierung betrachtet.

## 2.3 Konzept

Die Kombination der beiden Produkte OpenSSH und Terminal Services entspricht allen Anforderungen.

### Kommando-Zeilen orientierte Administration

Mit dem SSH Client kann eine Kommando-Zeilen Verbindung zu dem SSH Server aufgebaut werden. Dies ermöglicht volle Kontrolle des Ziel-Servers

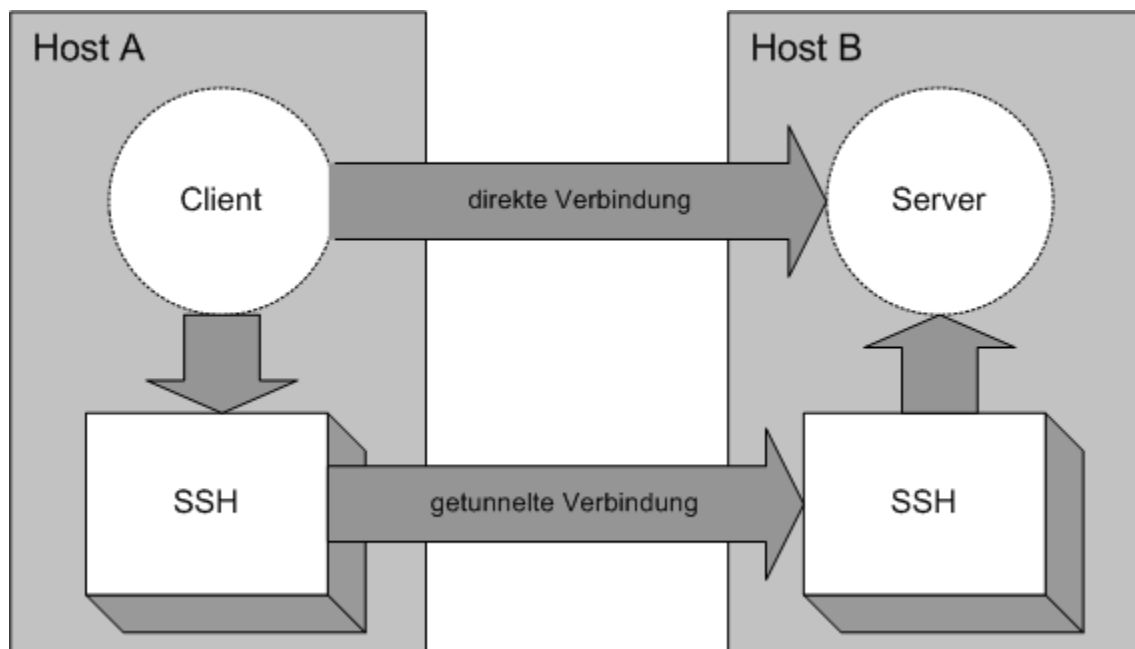
### Grafische Administration

Die Windows Terminal Services ermöglichen die Administration des Servers über eine GUI Oberfläche

### Terminal Server Port

In der Konstellation SSH – Terminal Services, ist es nicht notwendig den Port TCP/3389 nach außen anzubieten.

SSH bietet das Konzept des „Tunnelings“, d.h. die primäre Netzwerkverbindung wird mit dem SSH Dienst hergestellt (TCP/22). Danach dient SSH als „Relay“ zu den, auf dem Server lokal installierten Terminal Diensten. Port 3389 kann somit entweder mit einem vorgelagerten Packet-Filter (Router, Firewall) oder einem lokalen (RRAS, IP Filter) geschlossen werden.



**Maximale Software-basierende Sicherheit**

Das SSH Protokoll gilt als „sicher“. Weiterhin kann zusätzlich zu einem Login Passwort mit Client Zertifikaten authentifiziert werden.

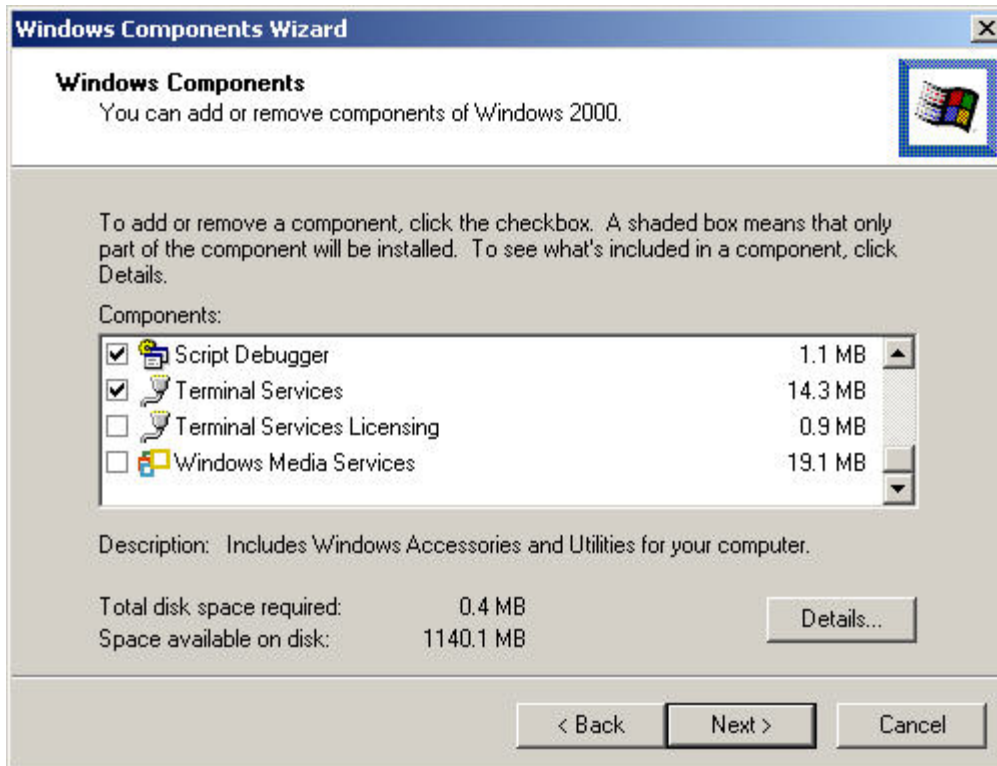
**Möglichst viele Client-Plattformen**

SSH Clients gibt es für nahezu jede Plattform. Für die optionale grafische Verwaltung gibt es Clients von Microsoft für jede Windows Plattform. Nicht-Windows Client werden von der Firma Citrix geliefert (Unix, DOS, MacOS).

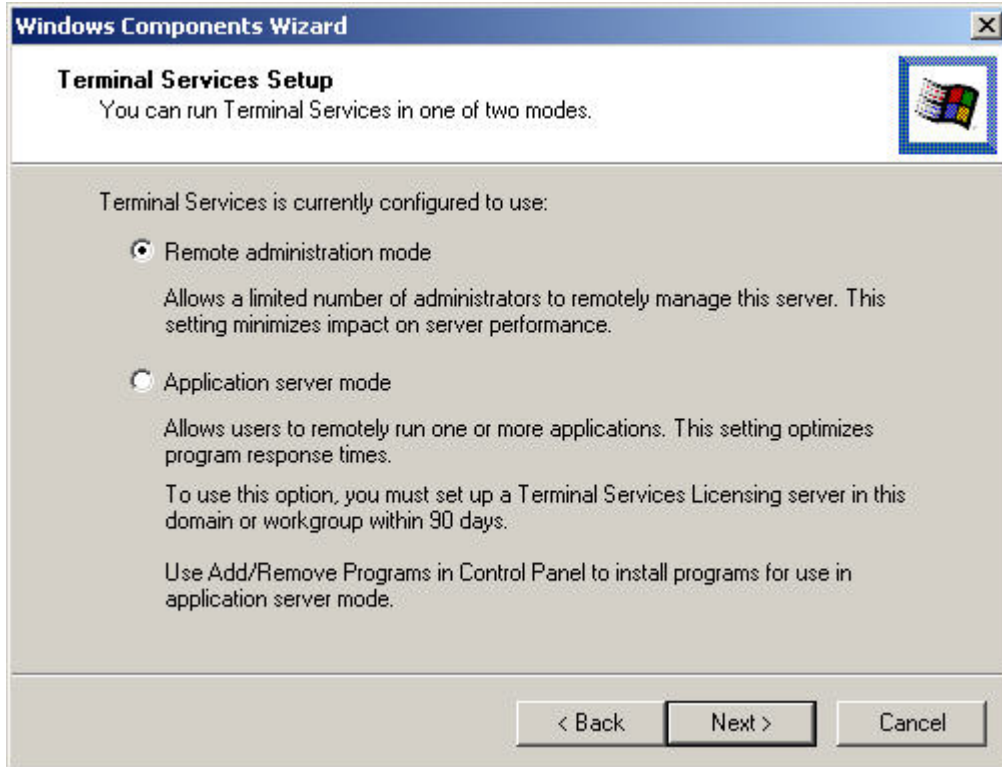
## 3 Implementierung

### 3.1 Terminal Services

Die Terminal Services werden und Control Panel / Software installiert.



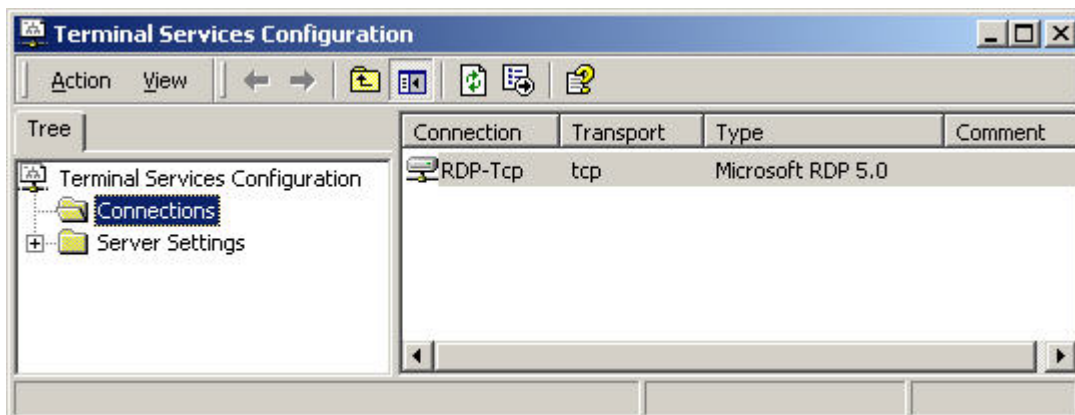
Außerdem ist der Remote-Administrations-Modus zu wählen.



Im Anschluss an die Installation sollte man noch einige Konfigurations-Änderungen vornehmen.

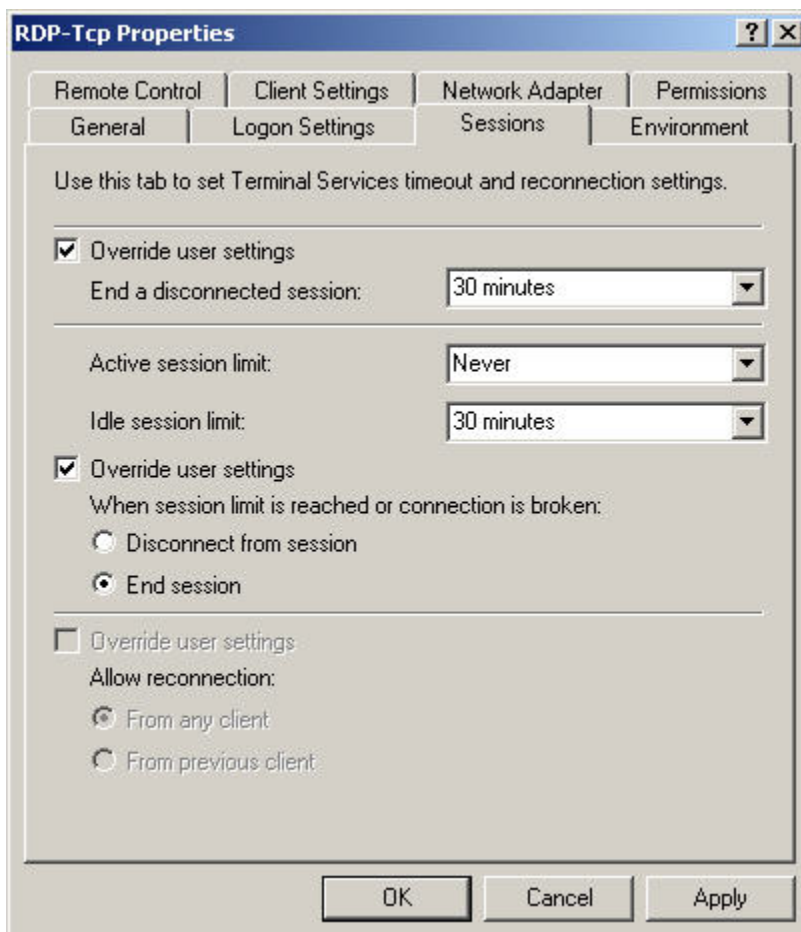
Die Terminal Services Konfiguration erreicht man unter Administrative Tools / Terminal Services Konfiguration.

Danach kann man die RDP-TCP Eigenschaften mit rechter Maustaste / Properties bearbeiten.

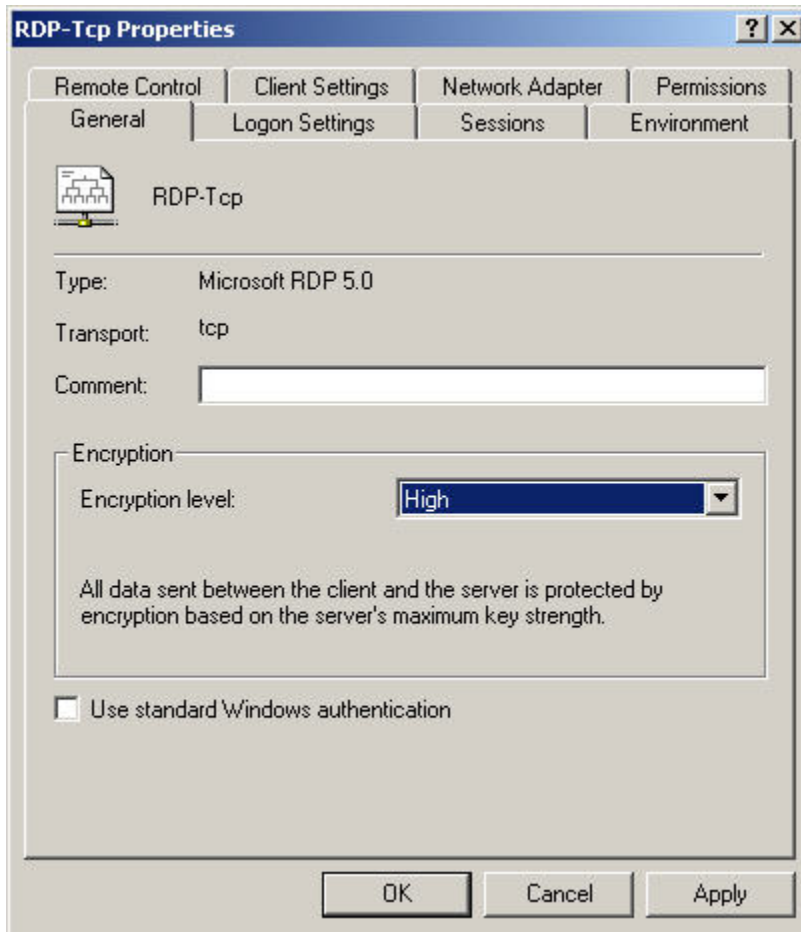


Folgende Einstellungen werden empfohlen:

Einstellung	Beschreibung	Empfohlener Wert
End a disconnected Session	Eine disconnected Session entsteht wenn der Benutzer den Client schließt ohne sich abzumelden	30 minutes
Active Session Limit	Maximale Session Dauer	Never
Idle Session Limit	Maximale Session Dauer, wenn keine Benutzer-Interaktion stattfindet	30 minutes
When Session Limit is reached or connection is broken	Automatischer Abbau der Session bei Verbindungs-Abbruch	End Session

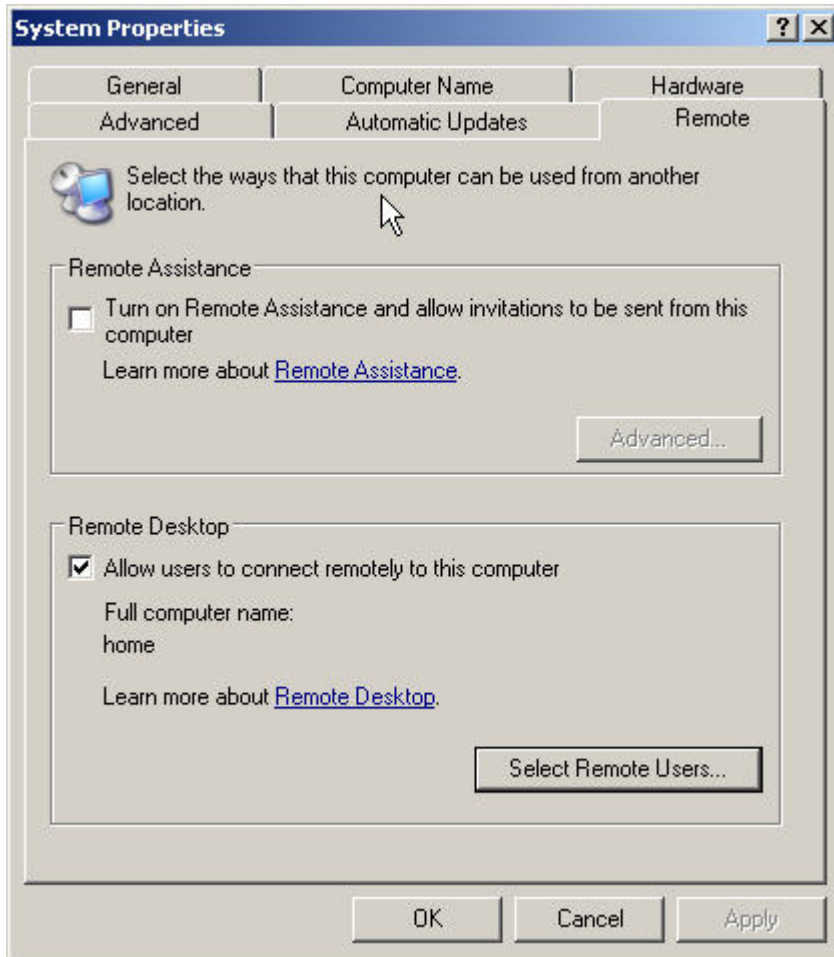


Weiterhin sollte die Verschlüsselungs-Tiefe auf „High“ gestellt werden, was 128 Bit entspricht.



Terminal Services legt weiterhin einen lokalen Benutzer ‚TSInternetUser‘ an. Dieser kann gelöscht werden.

Unter Windows Server 2003 werden die Terminal-Services für den Remote Admin Modus automatisch installiert, man muss sie lediglich aktivieren (My Computer -> Properties -> Remote)

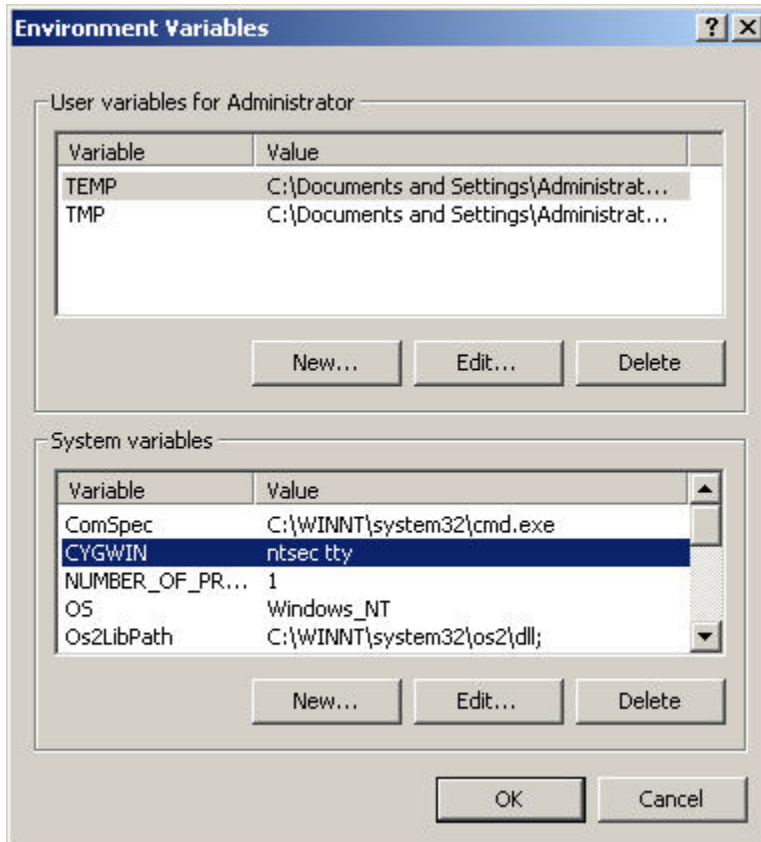


### 3.2 OpenSSH

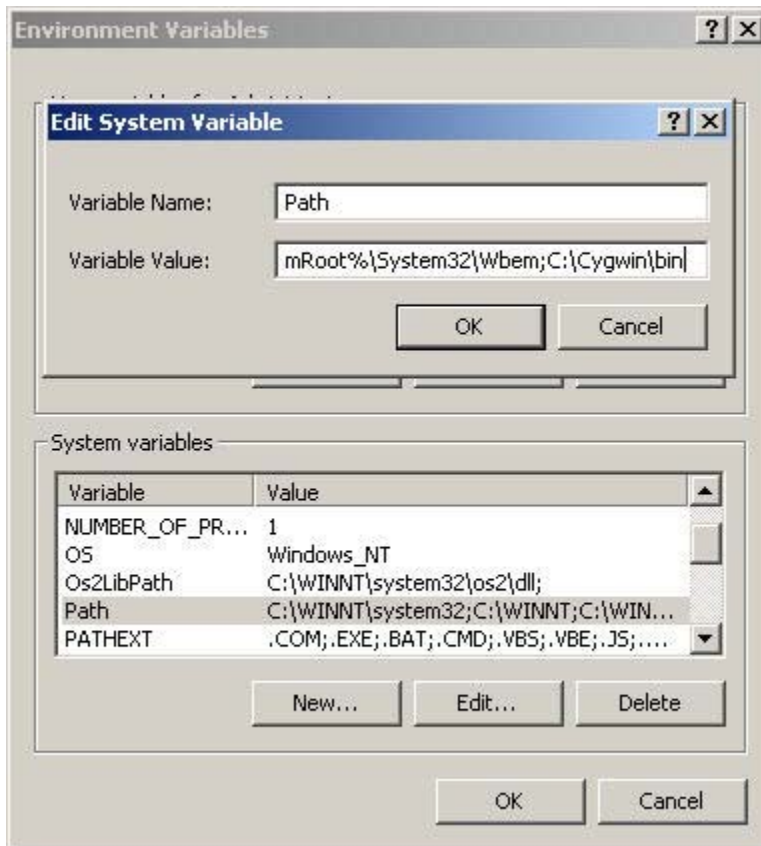
OpenSSH wird über Cygwin installiert.

Dazu müssen erst zwei Umgebungs-Variable angepasst werden.

- a. Die Umgebungsvariable ,cygwin' muss auf ,ntsec tty' gesetzt werden.



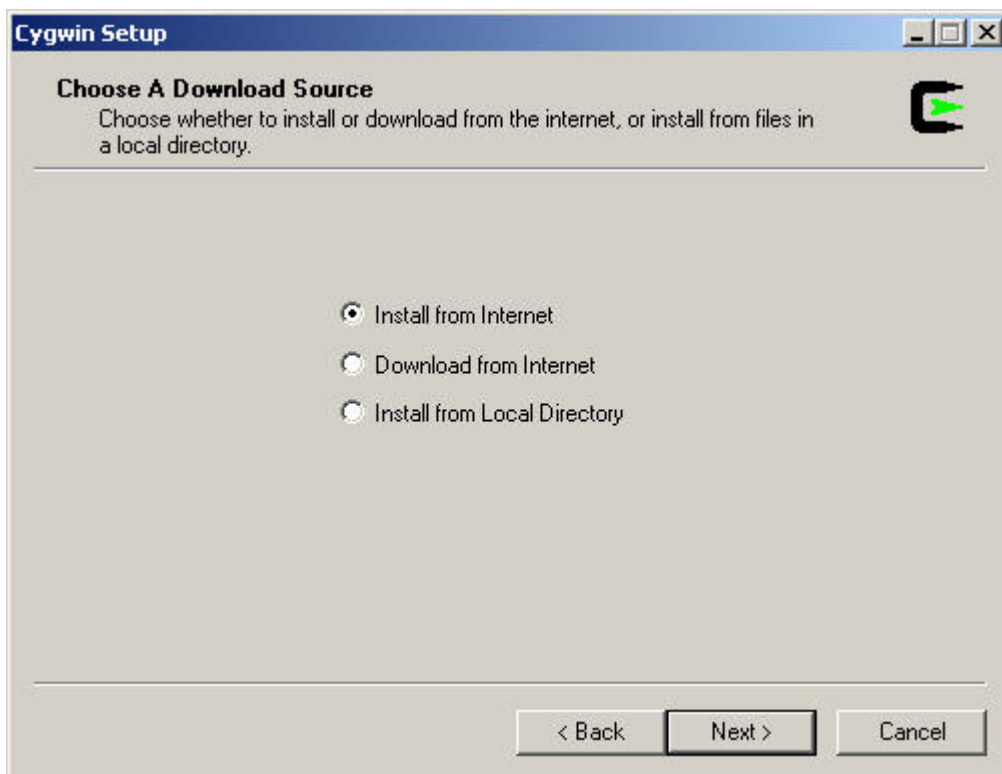
b. die Cygwin Binaries müssen in den Suchpfad aufgenommen werden.



Danach kann man das Internet-basierende Setup von Cygwin von <http://cygwin.com/setup.exe> aufrufen.

Hat der Rechner keinen Zugriff auf das Internet kann die Distribution auch vorher auf einen anderen Rechner geladen werden und dann mit dem Setup Programm auf dem Ziel-Server installiert werden. Das Cygwin Setup Programm macht nach dem Download der erforderlichen Dateien einen Integritäts-Check anhand der MD5 Checksumme.

Die Cygwin Distribution kann auch manuell von einem Cygwin Mirror heruntergeladen werden. In diesem Fall kann man die Integritäts-Prüfung manuell durchführen. ([www.cygwin.com](http://www.cygwin.com))



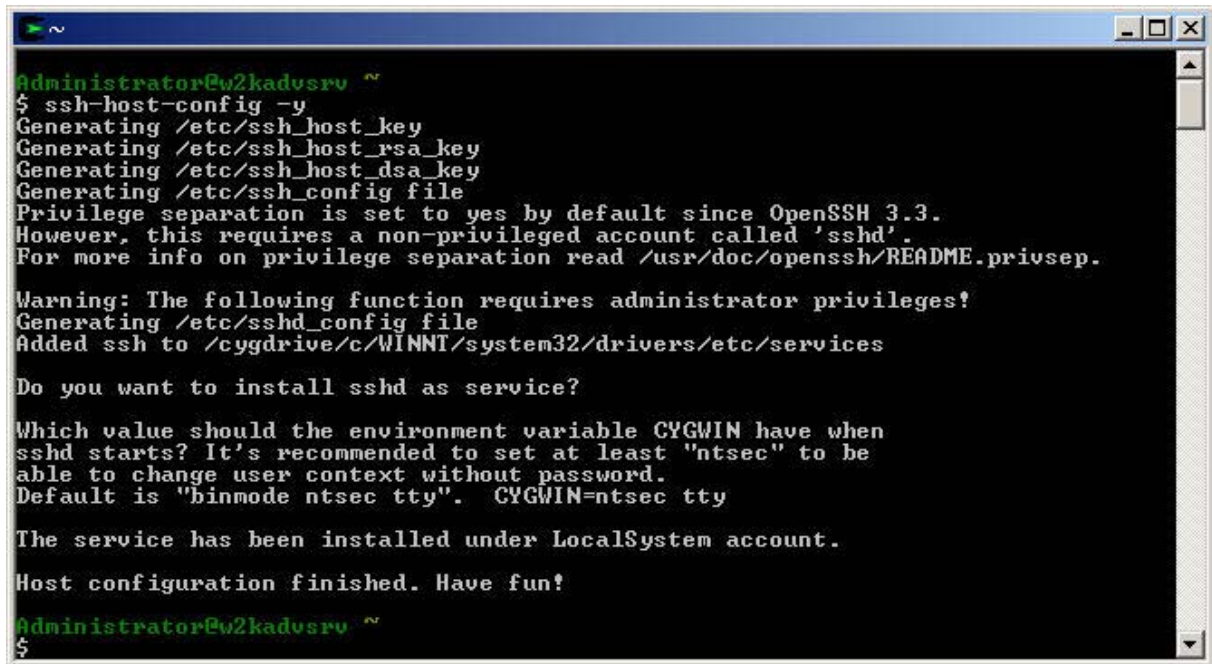
Folgende Komponenten von Cygwin sollten installiert werden:

<b>Sektion</b>	<b>Komponente</b>
Admin	Cygrunsv
Doc	Cywindoc
Editor	Vim
Net	OpenSSH

Nach der Installation von Cygwin muss die Shell gestartet werden und das Kommando

```
ssh-host-config -y
```

ausgeführt werden.



```
Administrator@w2kadvsrv ~
$ ssh-host-config -y
Generating /etc/ssh_host_key
Generating /etc/ssh_host_rsa_key
Generating /etc/ssh_host_dsa_key
Generating /etc/ssh_config file
Privilege separation is set to yes by default since OpenSSH 3.3.
However, this requires a non-privileged account called 'sshd'.
For more info on privilege separation read /usr/doc/openssh/README.privsep.

Warning: The following function requires administrator privileges!
Generating /etc/sshd_config file
Added ssh to /cygdrive/c/WINNT/system32/drivers/etc/services

Do you want to install sshd as service?

Which value should the environment variable CYGWIN have when
sshd starts? It's recommended to set at least "ntsec" to be
able to change user context without password.
Default is "binmode ntsec tty".  CYGWIN=ntsec tty

The service has been installed under LocalSystem account.

Host configuration finished. Have fun!

Administrator@w2kadvsrv ~
$
```

Danach wird mit dem Befehl

```
cygrunsrv -S sshd
```

der OpenSSH Server als Windows Dienst eingerichtet.

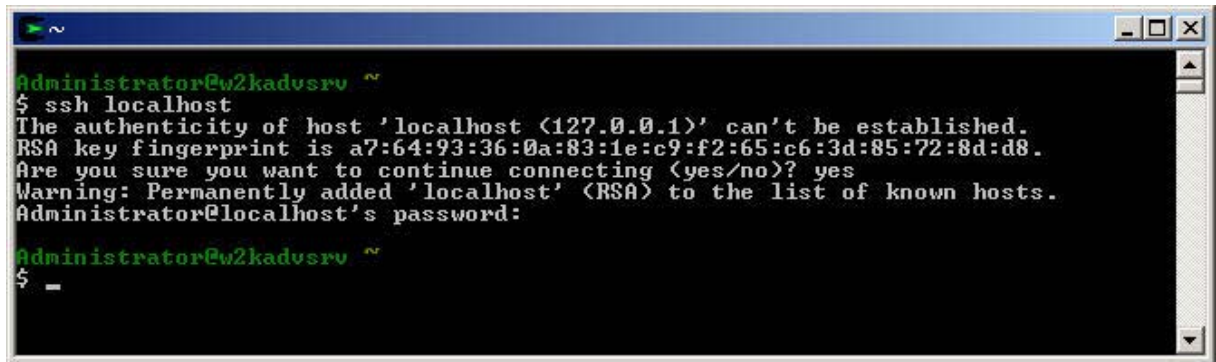


```
Administrator@w2kadvsrv ~
$ cygrunsrv -S sshd_
```

Nun kann man mit

```
ssh localhost
```

den Server testen.

A terminal window with a blue title bar and standard window controls. The text inside is as follows:

```
Administrator@w2kadvsrv ~  
$ ssh localhost  
The authenticity of host 'localhost (127.0.0.1)' can't be established.  
RSA key fingerprint is a7:64:93:36:0a:83:1e:c9:f2:65:c6:3d:85:72:8d:d8.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.  
Administrator@localhost's password:  
Administrator@w2kadvsrv ~  
$ -
```

### 3.3 Tunelling

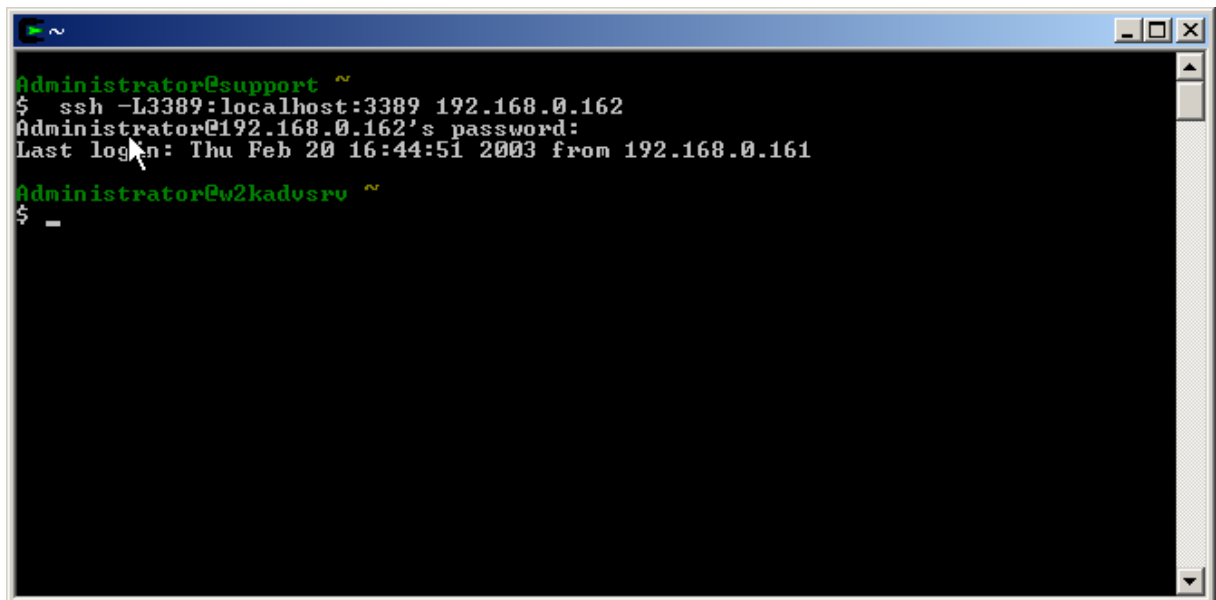
Als letzter Schritt muss der Tunnel für die Terminal Services eingerichtet werden.

Am SSH Client muss ein Port-Forwarding eingerichtet werden, so dass der Terminal Service Netzwerk-Verkehr an den Zielsever weitergeleitet wird.

Dies wird mit folgendem Befehl bewerkstelligt:

```
ssh -L3389:localhost:3389 192.168.0.162
```

Dies heißt soviel wie: "wenn am lokalen Rechner eine Anfrage an Port 3389 gerichtet wird, leite diese über SSH an den Port 3389 des Rechners mit der IP Adresse 192.168.0.102 weiter."



```
Administrator@support ~  
$ ssh -L3389:localhost:3389 192.168.0.162  
Administrator@192.168.0.162's password:  
Last login: Thu Feb 20 16:44:51 2003 from 192.168.0.161  
Administrator@w2kadvsrv ~  
$ -  
$
```

Jetzt muss lediglich noch am Client-Rechner ein Terminal Server Profil für den localhost eingerichtet werden.

**Clientverbindungs-Assistent**

**Eine Verbindung erstellen**  
Der Name, den Sie für die Clientverbindung eingeben, identifiziert die Verknüpfung.

Geben Sie einen kurzen Namen ein, der die Verbindung beschreibt.  
Verbindungsname:

Geben Sie den Namen oder die IP-Adresse des Terminalservers ein.  
Servername oder IP-Adresse:

## 4 Links

- [1] <http://www-user.tu-chemnitz.de/~hot/ssh/ssh.html>
- [2] <http://www.openssh.com/>
- [3] <http://www.ietf.org/html.charters/secsh-charter.html>
- [4] <http://www.secnetix.de/~olli/ssh/>
- [5] <http://www.oreilly.de/catalog/sshtdgger/>