

Secure access to the DESY network using SSH

UCO @ DESY

November 29, 2007, Hamburg

Contents

 1.1 How to reach UCO			4 5
 2 Introduction 3 Coming from Windows 3.1 Basic configuration of PuTTY 3.2 Remote Desktop (Target OS: Windows) 3.3 X11 (Target OS: UNIX/Linux) 			5
 3 Coming from Windows 3.1 Basic configuration of PuTTY			
 3.1 Basic configuration of PuTTY			6
3.2Remote Desktop (Target OS: Windows)3.3X11 (Target OS: UNIX/Linux)			6
3.3 X11 (Target OS: UNIX/Linux)			8
			9
3.4 Remote Shell (Target OS: UNIX/Linux/MacOS X)			9
3.5 Transferring files (Target OS: Windows)			10
3.6 Transferring files (Target OS: UNIX/Linux/MacOS X)			10
4 Coming from UNIX/Linux			12
4.1 Remote Desktop (Target OS: Windows)			12
4.2 X11 (Target OS: UNIX/Linux)			13
4.3 Remote Shell (Target OS: UNIX/Linux/MacOS X)			13
4.4 Transferring files (Target OS: UNIX/Linux/MacOS X)			14
4.5 Single-Sign-On login (Target OS: UNIX/Linux/MacOS X)			15
5 SSH-Proxy			16
6 Ports			17
6.1 Common ports			17
6.2 Free local ports			17
6.2.1 Windows			17
6.2.2 UNIX/Linux/MacOS X			17
7 Public/Private key authentication			19
7.1 Creation on Windows			19
7.2 Creation on UNIX/Linux/MacOSX			20
7.3 Afterwards	•••	•••	20
8 SSH-Agent			21
			01
8.1 Windows		• •	21

1 General Information

This documentation is part of a wide collection of documentations available at **UCO** (User Consulting Office). Please read carefully and feel free to ask questions.

1.1 How to reach UCO

Feel free to call for support or come along:

LOCATION	Phone	Room
Hamburg	040/8998 5005	Building 2b, Room 131d
Zeuthen	030/3762 7324	Building 1R, Room 21

2 Introduction

This document is intended to guide you in setting up secure connections to DESY's network over the Internet. To accomplish this, SSH is used as a protocol. SSH allows for an encrypted peer-to-peer connection running through any amount of relay stations for transferring commands and files. In addition to that SSH can be used to create tunnels for encapsulating and encrypting data packets of any TCP-based protocol into the SSH connection's stream. A listing of common protocols and their assigned default ports can be found on **page 17**.

ATTENTION: Please remember that SSH encryption obfuscates your traffic **only** from your machine to the gateway server. Further encryption behind the gateway server depends on the protocol you are using.

DESY Hamburg and Zeuthen are using special servers for you to connect to and tunnel through.

LOCATION	Server
Hamburg	bastion.desy.de
Zeuthen	pub.ifh.de

Terminal-Server (Windows-RDP-Server) at DESY by location (must be enabled for each account by UCO):

LOCATION	Server
Hamburg	adterm.win.desy.de
Zeuthen	znformica.ifh.de

3 Coming from Windows

This document is intended to help Windows users accessing remote services at DESY as well as anywhere else and uses the very powerful SSH client implementation \mathbf{PuTTY}^1 since it simplifies handling of tunnels and is Open Source Software.

3.1 Basic configuration of PuTTY

Session	*	Basic options for your PuTT	r' session
⊡ Logging ⊡ Terminal — Keyboard — Bell		 Specify the destination you want to contract the end of the state of t	nnect to Port 22
 Features Window Appearance Behaviour Translation Selection 		Connection type: C Raw C Telnet C Rlogin C Load, save or delete a stored session Saved Sessions	<u>S</u> SH C Seria
Colours Connection Data Proxy Telnet Riogin		Default Settings	Load Sa <u>v</u> e Delete
⊢ SSH – Kex – Auth – TTY – Y11		Close <u>w</u> indow on exit: C Always C Never I Only	on clean exit
		0	

Figure 1: **PuTTY settings** (server configuration)

PARAMETER	VALUE
Host Name	bastion.desy.de or pub.ifh.de
Port	22
Protocol	SSH

You can save your settings by assigning it an arbitrary name and clicking **Save after** having set all needed options. Please configure needed tunnels **before** saving. After this step a tunnel has to be configured. It should run through the server specified under **Host Name** and connect you to a computer inside the DESY network.

 $Provide \ the \ following \ settings \ under \ Connection \rightarrow SSH \rightarrow Tunnels:$

Set **YYYYY** to a free port $m(\rightarrow page 17)$ on **localhost** and set **ZZ** to the desired

¹http://en.wikipedia.org/wiki/PuTTY

Port forwarding	
Local ports accept connection Remote ports do the same (Second ports: Add new forwarded port: Source port Source p	ons from other hosts SH-2 only) <u>R</u> emove Add C Dynamic C IPv <u>6</u>
	Forwarded ports: Add new forwarded port: Source port MMMM Destination MMMM Local C Remote Auto C IPv4

Figure 2: **PuTTY settings** (tunnel configuration)

service's port on remote machine. **YYYYY** will pose as the tunnel's local end between **XXXXX:ZZ** and localhost. Click on ADD. **Open** starts a shell where authentication is needed.

Attention: The shell has to remain open during the whole session in order for the tunnel to stay alive.

3.2 Remote Desktop (Target OS: Windows)



Windows' **Remote Desktop**² service uses **port 3389** by default. Set **ZZ** in **Fig. 2** to **3389**. You can use the Terminal Server (\rightarrow **page 5**) if you don't want to connect to another specific machine. You can now start the RDP tool: **Start** \rightarrow **Run** \rightarrow **mstsc.exe**

emote Desktop Connection		
	Lab.	Options >>
	emote Desktop Connection	emote Desktop Connection

Figure 3: Establishing a Remote Desktop session (server configuration)

PARAMETER	VALUE
YYYYY	any

Attention:

You have to be in the access list for Remote Desktop users. This can be achieved as follows:

 $^{^{2}} http://www.microsoft.com/windowsxp/using/mobility/getstarted/remote intro.mspx$

3.3 X11 (Target OS: UNIX/Linux)



Set **ZZ** in **Fig. 2** to **22**. An X-Server has to be installed on the target machine (e.g. Xfree86, Xorg).

The terminal server (page 5) has X-Win32 preinstalled and can be used alternatively. Please adjust the entry Destination in Fig. 2 to look like this: [Terminal server]:3389

Open **X-Win32** after successful installation and start the Interface by left-clicking on the X-Win32 icon next to Windows' clock.

$$\label{eq:Wizard} \begin{split} Wizard \to Enter \ name \to StarNetSSH \to Enter \ hostname \ (localhost) \to enter \\ Login \ \& \ password \to choose \ target \to Finish \end{split}$$

3.4 Remote Shell (Target OS: UNIX/Linux/MacOS X)



The easiest way to access a remote shell service behind gateway server is to create a local proxy using PuTTY.

Set Hostname in **Fig.1** to the shell server **behind** the gateway. Provide the following settings under **Connection** \rightarrow **Proxy**:

PARAMETER	VALUE
Proxy type	local
Telnet command or	plink -l UserID tunnelserver -nc %host:%port
local proxy com-	
mand	This only works if you are using key authentication! See page 19

Set tunnelserver according to the table on page 5.

3.5 Transferring files (Target OS: Windows)

You can access your DESY network drives via the Remote Desktop Protocol. The neccessary steps are as follows:

Follow Chapter 2.2 and in the dialog shown in Fig. 3 click on Options \rightarrow Local Ressources \rightarrow under Local Devices and Ressources \rightarrow More... click on the plus sign next to Drives \rightarrow select the needed local drive \rightarrow OK \rightarrow Connect

3.6 Transferring files (Target OS: UNIX/Linux/MacOS X)

 \mathbf{WinSCP}^3 is a useful tool for file transfer and management between UNIX hosts and Windows clients.

	ruitnumper
Construction Distance in the	
tunnelserver	
Username Pas	sword
UserID	
Private kev file	
1	
Protocol	
Eile protocol SFTP -	Allow SCP fallback
	0.1.1.1
	Select color
	User name Pas UserID Private key file Protocol Eile protocol SFTP

Figure 4: Establishing an SFTP/SCP session (Windows)

Please set up the following parameters.

PARAMETER	VALUE	DESCRIPTION
Host name	page 5	Server to tunnel through
User name	Your DESY UserID	
Private key file	Your private key (.ppk)	See page 19

³http://winscp.net/eng/index.php

Now go to **Connection** \rightarrow **Tunnel** and set up this:

Session Stored sessions Logging	Connect through SSH tunnel	
Environment	<u>H</u> ost name	Po <u>r</u> t number
- Directories	target_host	22 💌
SCP/Shell	User name Pa	assword
Connection	UserID	
Proxy		
Tunnel	Private <u>k</u> ey file	
SSH		
 Key exchange 		
Authentication	Tunnel options	
Bugs	Local tunnel port	Autoselect 💌
Preferences		
Advanced options		

Figure 5: Establishing a tunnelled SFTP/SCP session (Windows)

Don't forget to enable the option Connect through SSH tunnel

PARAMETER	VALUE	DESCRIPTION
Host name	any SSH-enabled ma-	The remote end of the tunnel
User name	Your DESY UserID	

If you want to keep the settings saved for later use click on **Save** and enter an appropriate connection name.

4 Coming from UNIX/Linux

Most UNIX/Linux distributions are equipped with an SSH implementation like OpenSSH⁴ by default. The syntax for creating a tunnel is as follows:

ssh -C -L localport:remote_server:remote_port -l username tunnelserver
-C compresses the transmitted data and helps keeping bandwidth usage low.
-L is the parameter to create a tunnel.

The standard parameters:

PARAMETER	VALUE	DESCRIPTION
localport	any (above 1024 if	Tunnel's end on your computer
	not created by root)	
remote_server	Arbitrary computer be-	Tunnel's target
	hind tunnelserver	
username	Your DESY UserID	
tunnelserver	bastion.desy.de or	Server through which the connection
	pub.ifh.de	gets tunneled

4.1 Remote Desktop (Target OS: Windows)



For accessing an RDP session open a shell, create a tunnel as shown on **page 12** with **remote_port** being set to **3389** and connect to it using the following line:

rdesktop -g resolution -a color_depth -n localhost:localport

Please use the following settings:

PARAMETER	VALUE	DESCRIPTION
resolution	800x600, 1024x768, etc.	Screen resolution in pixels
color_depth	8,16	Color resolution in bits

Reduction of resolution and color depth lowers bandwidth usage and is advisable on slow Internet connections.

⁴http://www.openssh.com/

4.2 X11 (Target OS: UNIX/Linux)



Forwarding SSH connections is accomplished as follows:

ssh -C -X -L localport:remote_server:remote_port -l username tunnelserver

-X activates the forwarding of X11 sessions. Provided you are running **xterm** you can start arbitrary GUI programs after issuing the SSH command.

4.3 Remote Shell (Target OS: UNIX/Linux/MacOS X)



There are two options for connecting to a shell behind tunnelserver.

• Using two seperate SSH-sessions (quick and dirty) Please use the following setting:

PARAMETER	VALUE	DESCRIPTION
remote_port	22	Port for SSH

After having established the SSH tunnel you have to connect to the tunnel's local end using a second SSH session..

• Using a static configuration file (slow and clean) Please edit the file ~/.ssh/config and add the following lines for each target_machine on which you require shell access.

```
Host target_machine
ProxyCommand ssh bastion.desy.de netcat -w 3 target_machine 22
```

4.4 Transferring files (Target OS: UNIX/Linux/MacOS X)

 $gFTP^5$ offers both FTP and SCP/SFTP connections.

gFTP 2.0.17						_ 0 >
<u>ETP L</u> ocal <u>R</u> emote <u>B</u>	ookmarks <u>T</u> ransfers	L <u>ogg</u> ing Tool <u>s</u>	<u>H</u> elp			
Host:		Y Port:	User:		Pass:	SSH2 🝸 🔇
/afs/desy.de/user/s/su [Local] [All Files]] Filename . Mathematica . OldFiles . acrobat . acrobat . adobe . camel_certs . cedit . config . defcon . eggcups . elinks . evolution	chalter	▼ 225 227 227 227 227 227 227 227	Not con	nnected* ename		Size Us
Filename	Progress		11012			
IFTP 2.0.17, Copyright (C bout this program, pleas ww.gftp.org/ IFTP comes with ABSOLU edistribute it under certai) 1998–2003 Brian Masi e feel free to email them TELY NO WARRANTY; for in conditions; for details	ney <masneyb@g to me. You can a details, see the , see the COPYIN</masneyb@g 	ftp.org>. If yoi Ilways find out COPYING file. T G file	u have any questi the latest news a 'his is free softwai	ons, comments bout gFTP from re, and you are	, or suggestions my website at http:/, welcome to

Figure 6: Establishing an SFTP/SCP session (UNIX/Linux)

PARAMETER	VALUE	DESCRIPTION
Host	page 5	Server through which the connection gets
		tunneled
User	Your DESY UserID	
Passwort	Your DESY password	
Mode	SSH2	

⁵http://gftp.seul.org/

4.5 Single-Sign-On login (Target OS: UNIX/Linux/MacOS X)

You can set up the file ~/.ssh/config to allow Single-Sign-On login. This means that you will have to authenticate just once and not on every single system you connect to. Add this to the beginning of the file

Host * ControlMaster auto ControlPath /tmp/master-%r@%h:%p

Please attach the following template for every host you want to connect to:

Host target_machine Hostname target_hostname User UserID ProxyCommand ssh UserID@bastion.desy.de netcat -w3

PARAMETER	VALUE	DESCRIPTION
target_machine	any	target host name without its domain
target_hostname	any	target host name with its domain
UserID	Your DESY UserID	

5 SSH-Proxy

Using an SSH proxy can provide usability benefits. An SSH proxy allows you to tunnel any protocol through it. This allows you to relay the traffic of every **SOCKS-proxy**⁶ aware application through an SSH encrypted machine. You will need to have access to a remote SSH server for this to work. Use this if you connect to untrusted networks (e.g. in internet cafes) to read your mail, etc.

ATTENTION: You will have to setup the newly created Proxy running on localhost:localport in the proxy settings configuration of each application that is supposed to use it. The syntax for creating a tunnel on the client's side is as follows:

ssh -C -D localport -l username tunnelserver

-C compresses the transmitted data and helps keeping bandwidth usage low. -D is the parameter to achieve proxy forwarding.

PARAMETER	VALUE	DESCRIPTION
localport	any (above 1024 if	SOCKS5 compatible proxy port on
	not created by root)	the local machine
username	Your DESY UserID	
tunnelserver	page 5	Server through which the connection
		gets tunneled

tunnelserver can be any reachable system running an SSH daemon. Please note that you need to have an account on tunnelserver before trying this.

⁶http://en.wikipedia.org/wiki/SOCKS

6 Ports

Ports are endpoints of connections between networked services. Each machine manages up to 65536 ports. The first 1024 ports cannot be used by any user except root. The port number is not neccessarily linked to a particular service, i.e. you can use other ports than the default ones for services. However, sticking to the default port usage scheme can ease usability significantly.

6.1 Common ports

A small collection of commonly used ports.

Port Number	SERVICE	Port Number	SERVICE
20, 21	FTP	993	IMAPS
22	SSH	995	POP3S
88	Kerberos	3389	RDP
389	LDAP	3389	RDP
515	LPD	5800, 5900	VNC

6.2 Free local ports

How to determine free ports

6.2.1 Windows

Open a shell using Start / Run / cmd

```
C:\>netstat -an
```

Active Connections

Local Address Foreign Address State
xxx.xxx.xxx:135 xxx.xxx.xxx:0 LISTENING
xxx.xxx.xxx:445 xxx.xxx.xxx:0 LISTENING
xxx.xxx.xxx:1043 xxx.xxx.xxx:0 LISTENING
xxx.xxx.xxx:139 yyy.yyy.yyy:1182 ESTABLISHED
xxx.xxx.xxx:1060 xxx.xxx.xxx:0 LISTENING

The ports 135, 445, 1043, 139 and 1060 are already in use and thus can not be used as tunnel server ports.

6.2.2 UNIX/Linux/MacOS X

Open a shell like **xterm**, etc.

[localhost] ~ \$ netstat -lnt

Active Internet connections (w/o servers) Proto Recv-Q Send-Q Local Address Foreign Address State tcp 0 0 localhost:33921 target1:3389 ESTABLISHED tcp 0 0 localhost:33793 target2:5190 ESTABLISHED

The ports **33921** and **33793** are already in use and thus can not be used in this example.

7 Public/Private key authentication

7.1 Creation on Windows

<ey< th=""><th></th><th></th><th></th></ey<>			
Public key for pastir	ng into OpenSSH auth	iorized_keys file:	
ssh-rsa AAAAB3NzaC1yc2 W3QRinyllAYBGbo c0Dwhh/omtSec3 Dw== rsa-key-2007	2EAAAABJQAAAIB5Vo otsDYIES80dg2NzQc) 10AtP8TQ7TS6U+o2C 71129	qun1zZHZDfqEsyQmNJU /mnU37WtnH58hVZCDK +zd5z4GrYiot4Qy4xvT2n	vTBDioIP6+Bxmur+ kwJqZBhdkvDUcsqw aZTotG2oCJCc0FFZ
Key fingerprint:	ssh-rsa 1023 f3:9	c:77:43:73:79:13:4d:8a:2	a.cd:8f:1c:29:2d:1f
Key <u>c</u> omment:	rsa-key-20071129	9	
Key p <u>a</u> ssphrase:			
C <u>o</u> nfirm passphrase	e: [
C <u>o</u> nfirm passphrase Actions			
C <u>o</u> nfirm passphrase Actions Generate a public/(e e se se		<u>G</u> enerate
Confirm passphrase Actions Generate a public/f Load an existing pri	: private key pair ivate key file		<u>G</u> enerate Load
Confirm passphrase Actions Generate a public/ Goad an existing pri Gave the generated	: private key pair ivate key file d key	Save p <u>u</u> blic key	<u>G</u> enerate Load Save private key
Confirm passphrase Actions Generate a public/p Load an existing pri Gave the generated Parameters	: private key pair ivate key file d key	Save p <u>u</u> blic key	<u>G</u> enerate Load Save private key
Confirm passphrase Actions Generate a public/p Load an existing pri Save the generated Parameters Parameters Cype of key to gene C SSH- <u>1</u> (RSA)	: private key pair ivate key file d key erate:	Save pyblic key	<u>G</u> enerate Load Save private key

Figure 7: **PuTTY Key Generator** (Windows)

PARAMETER	VALUE	DESCRIPTION
Key passphrase	any	The phrase protecting your private key
Type of key to	SSH-2-RSA or SSH-2-	
generate	DSA	

You can use the **PuTTY Key Generator** to create keypairs. It comes with the NetInstall version of PuTTY or can be downloaded⁷.

Start the key generator, choose a key type under $\mathbf{Parameters}$ and click on $\mathbf{Actions} \to \mathbf{Generate}$

Enter a passphrase to protect your keys in the fields 'Key passphrase' and 'Confirm passphrase'. Now save both public and private keys. Your public keys should have the same name as your private keys with the addition of the suffix .pub.

⁷ftp://ftp.chiark.greenend.org.uk/users/sgtatham/putty-latest/x86/puttygen.exe

7.2 Creation on UNIX/Linux/MacOSX

Create a set of keys on the remote machine by doing the following:

ssh-keygen -t rsa ssh-keygen -t dsa

It is highly advisable to **issue a non-trivial passphrase** for the keys to prevent login abuse. Now copy your public keys to the target machine as such:

scp ~/.ssh/*.pub UserID@target_machine:~/.ssh

7.3 Afterwards...

After having performed the above procedures, please log into **bastion.desy.de**, type **faq** and follow the instructions under 'How to use public/private keypairs?'.

8 SSH-Agent

An SSH agent manages key exchange and authentication for any amount of keys. As a result you have to type your key's passphrase only once and the agent stores the decrypted private key in memory for later use. If you initiate a connection to any machine that have correspondent public keys, login will be performed without further input from you.

8.1 Windows

Pageant from the PuTTY package is used as an SSH agent under Windows. It comes with the NetInstall version of PuTTY or can be downloaded⁸. Once started, it can be opened by clicking its icon in Windows' systray.



Figure 8: **PuTTY Pageant** (Windows)

Click on Add Key \rightarrow choose your **private key(s)** (suffix: .ppk) \rightarrow **Open Pageant** will now authenticate automatically when using **PuTTY** to log into a system which is equipped with your public keys.

8.2 UNIX/Linux/MacOSX

Start the SSH-Agent by running ssh-agent in a shell. Load your private key(s) into the agent: ssh-add ~/.ssh/id_dsa for the DSA keypairs ssh-add ~/.ssh/id_rsa for the RSA keypairs

Now connect to a machine which is equipped with your public keys.

 $^{^{8} \}rm ftp://ftp.chiark.greenend.org.uk/users/sgtatham/putty-latest/x86/pageant.exe$