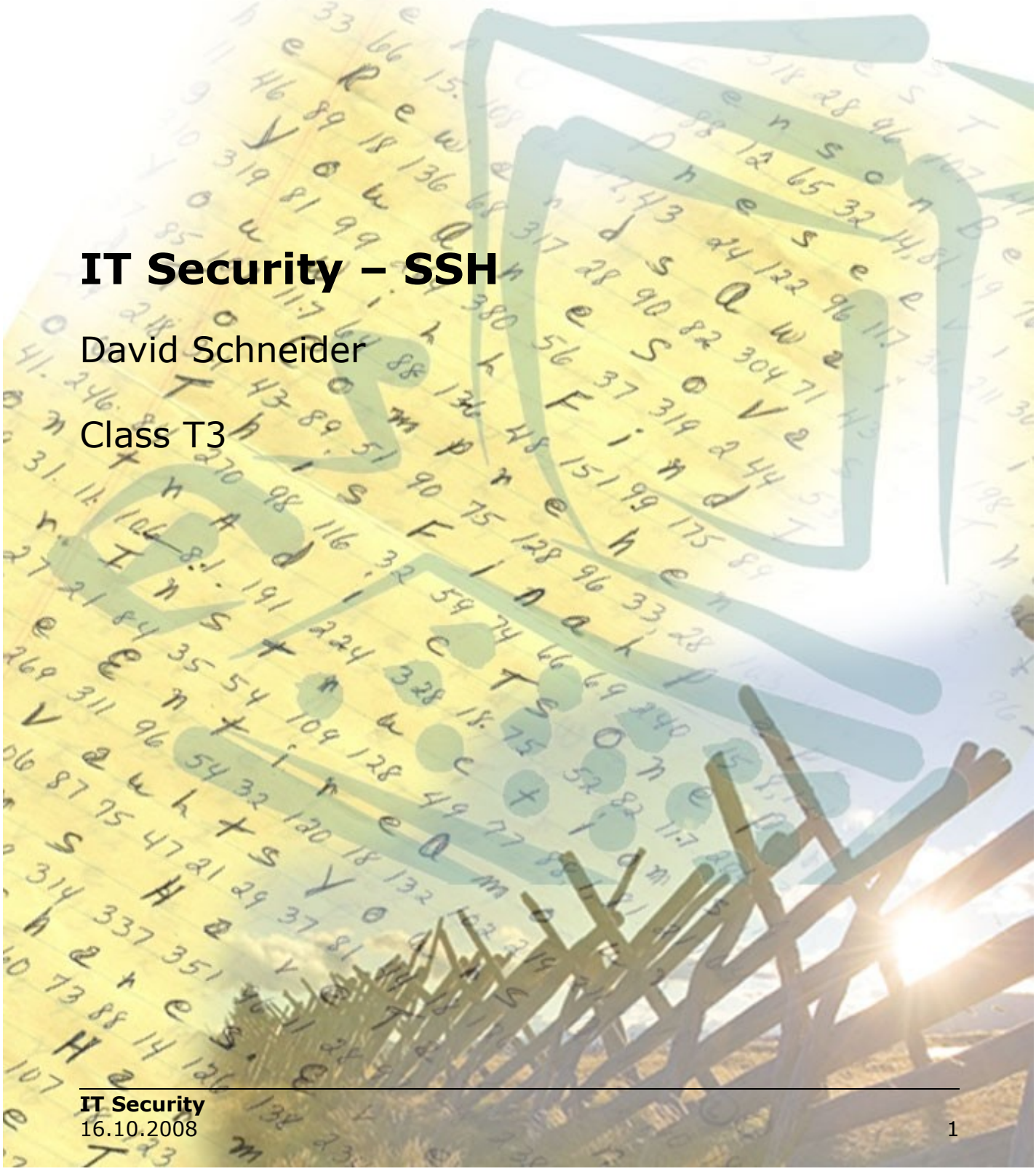
IT Security



IT Security – SSH

David Schneider

Class T3

Table of content

1 Check the SSH daemon

SSH or Secure Shell is a network protocol to establish a secure channel between two network devices. SSH is encapsulated in TCP and uses the well-known port 22. From the version 2 SSH provides advanced security trough Deffie-Hellman key exchange.

ifconfig

Due to SSH connects trough TCP we need to now the servers IP address. The linux shell command **ifconfig** displays the interface configuration. In this case just the address of the interface eth1 is interesting.

```
[root@localhost ~]# ifconfig eth1
eth1   Link encap:Ethernet  HWaddr 00:13:20:56:2D:B6
       inet addr:160.98.31.6  Bcast:160.98.31.255  Mask:255.255.254.0
       inet6 addr: 2001:620:40b:1:213:20ff:fe56:2db6/64 Scope:Global
       inet6 addr: fe80::213:20ff:fe56:2db6/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:1300 errors:0 dropped:0 overruns:0 frame:0
       TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:161621 (157.8 KiB)  TX bytes:5022 (4.9 KiB)
```

netstat

Netstat is the command used to print the network connections, routing tables, interface statics...
We use the netstat command to check if the SSH daemon is started. **netstat -atunp** shows all sockets TCP and UDP in numerical manner with the program name.

```
[root@localhost ~]# netstat -atunp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:56353           0.0.0.0:*               LISTEN      1750/rpc.statd
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1731/rpcbind
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2696/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      2157/cupsd
tcp        0      0 :::22                   :::*                    LISTEN      2696/sshd
udp        0      0 0.0.0.0:703             0.0.0.0:*                           1750/rpc.statd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           2143/dhclient
udp        0      0 0.0.0.0:60252           0.0.0.0:*                           1750/rpc.statd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           2145/avahi-daemon
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1731/rpcbind
udp        0      0 0.0.0.0:631             0.0.0.0:*                           2157/cupsd
udp        0      0 0.0.0.0:634             0.0.0.0:*                           1731/rpcbind
```

Alternative the started servers could be printed with the **chkconfig** command.

nmap

The network exploration tool (port scanner) shows us on which ports another system listens for incoming traffic.

The command **nmap -sT -p 1-100 server_ip_address** prints the open ports for the TCP protocol.

```
[root@localhost ~]# nmap -sT -p 1-100 localhost

Starting Nmap 4.53 ( http://insecure.org ) at 2008-09-23 13:30 CEST
Interesting ports on localhost.localdomain (127.0.0.1):
Not shown: 98 closed ports
PORT    STATE SERVICE
22/tcp open  ssh
25/tcp open  smtp
```

2 Start/Stop the SSH Daemon

We are default in the runlevel 3. If we like to stop the sshd we use the command **/etc/rc.d/rc5.d/S55sshd stop** to shut the service down. With the command start we can start it or with restart apply the new config.

```
[root@localhost ~]# /etc/rc.d/rc5.d/S55sshd stop
Stopping sshd:                                             [  OK  ]
[root@localhost ~]# /etc/rc.d/rc5.d/S55sshd start
Starting sshd:
```

3 Linux boot

The file /etc/inittab specifies the default runlevel. The different runlevels are:

0 halt
1 Single user mode
2 Multiuser, without NFS
3 Full multiuser mode
4 unused
5 X11
6 reboot

The default is init 5 specified in the file with **id:5:initdefault:**

To change the runlevel we just need to type for example **init 6** to reboot.

Started applications

Each runlevel has a director in **/etc/rc.d/**. In the directory there are symbolic links for all the applications which are executed at boot.

```
[root@localhost rc5.d]# ls
K01smartd           K72ntpd             K91capi             S25netfs
K01smolt            K73winbind          K95firstboot        S25pcscd
K05saslauthd        K73ypbind           S00microcode_ctl    S26acpid
K10psacct           K74lm_sensors       S06cpuspeed         S26haldaemon
K15gpm              K74nscd             S08ip6tables        S26udev-post
K15httpd            K75ntpdate          S08iptables         S27NetworkManager
K20jetty            K76openvpn          S09isdn             S29setroubleshoot
K20nfs              K83named            S11auditd           S50bluetooth
K20tomcat5          K84btseed           S12restorecond      S55sshd
K24irda             K84bttrack          S12rsyslog          S80sendmail
K35backuppc         K84wpa_supplicant   S13irqbalance       S90crond
K35nmb              K85racoon           S13rpcbind          S90kerneloops
K35smb              K87multipathd       S14nfslock          S95atd
K50netconsole       K89dund             S15mdmonitor        S96avahi-daemon
K50snmpd            K89netplugd         S18rpcidmapd        S98cups
K50snmptrapd        K89pand             S19rpcgssd          S99anacron
K50vsftpd           K89rdisc            S22messagebus       S99local
K69rpcsvcgssd       K90network          S25fuse
```

The symbolic links are executet at start-up in alphabetic order. All the links which begins with S are started first, after the applications which starts with the letter K are killed.

With the number after the letter we can change the order of execution.

The command **chkconfig** mentioned before shows the system services and the runlevel information for this service.

```
[root@localhost rc5.d]# chkconfig
NetworkManager      0:off   1:off   2:on    3:on    4:on    5:on    6:off
acpid               0:off   1:off   2:on    3:on    4:on    5:on    6:off
anacron             0:off   1:off   2:on    3:on    4:on    5:on    6:off
atd                 0:off   1:off   2:off   3:on    4:on    5:on    6:off
auditd              0:off   1:off   2:on    3:on    4:on    5:on    6:off
avahi-daemon        0:off   1:off   2:off   3:on    4:on    5:on    6:off
backuppc            0:off   1:off   2:off   3:off   4:off   5:off   6:off
bluetooth           0:off   1:off   2:on    3:on    4:on    5:on    6:off
btseed              0:off   1:off   2:off   3:off   4:off   5:off   6:off
bttrack             0:off   1:off   2:off   3:off   4:off   5:off   6:off
capi                0:off   1:off   2:off   3:off   4:off   5:off   6:off
cpuspeed            0:off   1:on    2:on    3:on    4:on    5:on    6:off
crond               0:off   1:off   2:on    3:on    4:on    5:on    6:off
...
```

If we want to turn a certain service off for one of the runlevels the command **chkconfig --level 5 atd off** changes the configuration so that the service atd wont be started the next start-up.

4 SSH Connection

To connect a SSH server we use the command **ssh server_IP**. If we want to have support of X traffic we use the additional option **-X**.

```
root@LeNotre:~# ssh 160.98.31.6 -v
OpenSSH_4.7p1 Debian-8ubuntu1.2, OpenSSL 0.9.8g 19 Oct 2007
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Applying options for *
debug1: Connecting to 160.98.31.6 [160.98.31.6] port 22.
debug1: Connection established.
debug1: permanently_set_uid: 0/0
debug1: identity file /root/.ssh/identity type -1
debug1: identity file /root/.ssh/id_rsa type -1
debug1: identity file /root/.ssh/id_dsa type -1
debug1: Remote protocol version 2.0, remote software version OpenSSH_5.0
debug1: match: OpenSSH_5.0 pat OpenSSH*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1.2
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-cbc hmac-md5 none
debug1: kex: client->server aes128-cbc hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST(1024<1024<8192) sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: Host '160.98.31.6' is known and matches the RSA host key.
debug1: Found key in /root/.ssh/known_hosts:1
debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,gssapi-with-mic,password
debug1: Next authentication method: gssapi-with-mic
debug1: Next authentication method: publickey
debug1: Trying private key: /root/.ssh/identity
debug1: Trying private key: /root/.ssh/id_dsa
debug1: Next authentication method: password
root@160.98.31.6's password:
debug1: Authentication succeeded (password).
debug1: channel 0: new [client-session]
debug1: Entering interactive session.
debug1: Sending environment.
debug1: Sending env LANG = fr_CH.UTF-8
Last login: Fri Sep 26 11:29:45 2008 from 160.98.158.96
```

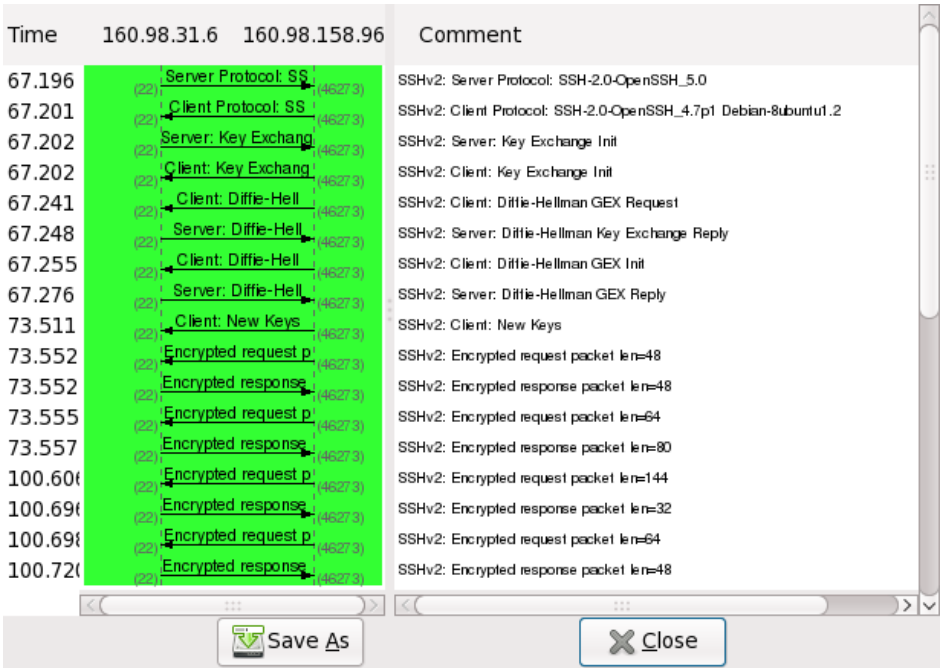
The servers public key is saved on the client machine in the file **$HOME/.ssh/known_hosts**.

```
|1|6XlB8F4dKiDUSRY8fBuF5sf/6Js=|wvyXi7qk+/uhUoAT66Q9lVio6oA= ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA0TcDLfT1XU8Yj27TKR3pZ0OXA6BVBehwSQtg+SEx6r8rPowc8
ZmtJw6u/s/mCG0VCjEiVbO1dsI5vGARFzbA3ooZDD07hx+by4Qd1JFgFU7oAahNnmTG/swHLH0G4V
rSef8DJadb3v7VljfM3cG8GqV6iuT1NsV1QtZ2XKEvpYPDY+oITLyvMbNwy8edsQ+z3m0hjr7aZFryKX
UVp1htg4WfP/lvcpve2cx0aulYWlWRJQvCekQt2j7n2ogha0pmff3zmjZBMyzJ6OgBGJBKW7PUELDYCg
6aGdbA2pNGw372p5kXA3jQVd93AqoiYgWygTLtms25mMhBaP4ERj3D5Q==
```

The server himself has its key in the directory **/etc/ssh/**.
For example in the file ssh_host_rsa_key we find the servers private RSA key.

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEA0TcDLfT1XU8Yj27TKR3pZ0OXA6BVBehwSQtg+SEx6r8rPowc
8ZmtJw6u/s/mCG0VCjEiVbO1dsI5vGARFzbA3ooZDD07hx+by4Qd1JFgFU7oAahN
nmTG/swHLH0G4VrSef8DJadb3v7VljfM3cG8GqV6iuT1NsV1QtZ2XKEvpYPDY+oI
TLyvMbNwy8edsQ+z3m0hjr7aZFryKXUVp1htg4WfP/lvcpve2cx0aulYWlWRJQvC
ekQt2j7n2ogha0pmff3zmjZBMyzJ6OgBGJBKW7PUELDYCg6aGdbA2pNGw372p5kX
A3jQVd93AqoiYgWygTLtms25mMhBaP4ERj3D5QIBIwKCAQEAj3YuEOJ8XT2NLyds
OXOYu9aTcDNtgGTfSAfNd6G71CtQ2m60wu0JBNbeZZXfkL/T6btKvmyZsIUu55JU
3LfUtdsJ3IHCa0jf3ALSoGOvmZUxYDjkxGJckXX2SmRcfUWXpByUcZdU8K6+WGDG
/nY318HtoRIHO4AV5LBRKZMZW4xPnnR7Nm8YIDF2q3fMDiLBpDw8GJmO0U5VGzr2
AHSUlubrP2GyNKKV4KDBGnxlOvPUNKgcuy0lNafW+gGkZ7x0qE1mcYCyNu3ktDUg
XwihDKWxclw1UL96IbJnNlQdkmyL7sMucnUDQpRk766nJBuDxgiFxJAttEEhLavi
OPWWiwKBgQDxiOuXrQCcpRT3V0hurmYkG16Eco7U/5z7oPAF1ciHCJwHLxmIBh0J
YogP0eK7Ek5/leD4UdjGk7CJdZ1nFQxaAlr9oxKfY1/UNgLloZPeKON05bnTlVJO
Lt4PpQe21K/cJH4nm+lP1UROeZrsO7w7uDTS1vah5vXoEbQ28rf/SQKBgQDdvpSS
Ra9kswEBunB6w2GKs4wazaVqc4EPwM8Tu5CNyYOA/xwY37xlj69K/RE0Ug4mwmBK
7MmRAvmaOFL/6aCNBjIV5vKd6lyy1sJMQbqBVbQRuYXhXvpkedSFW6Y6I/DOcZOh
AKTbv4LAANstEaFWnYcCWoTVCvNeSgR4wHTzvQKBgHVRIfk2xchtcJVk6KrJvJUx
3XOIGX1mNk5Ve+WTuSulGJXGa3yVQU2zg+ptQj2bLXErioc9sm8UiPJPEfA9bGY7
qInS3SjYfwCshRBzDU6mJVYJLl90ac40BXVQKFF9P3mOEWOyIN2iGdz5PJ6gq+J+
CwdSd8r7Jvuw0+Asv8UjAoGAUly61z50WJpCM9eI3SP/qIvU9AM9cK6PBdnmg6wR
HriP7hWcuMghZ40j2gY5lxcp0+HL/pJ2wNU4I1a/viOTZ5SWQqY825GQJSsyOZTB
ncC/OciQ0A1VsEqCIueG4mUe7ZfmYF9TLQyeR1DGb9NZLszLvwutrjdEc3qT8lYc
z48CgYEApCcW1RNYroqWp8qzax42gh1I/A1Daoium7pQo9D1EXWEUhGBQA1RRWg7
FGuUdjA7f+X0jA7IFyFU84YIgp9y2sMsl392mAyDPxucBFkSqG5TTV17kINAPVed
9MgLwvL3rRiTbdfklI+EZ8GFWc1zXFvUpC4TIPZr8joFKsuzZOc=
-----END RSA PRIVATE KEY-----
```

The traffic between the server and the client is the following:



Generate DSA key

To generate a DSA key we use **ssh-keygen -t dsa -b 1024** which creates a 1024 bit DSA key and saves it at **/root/.ssh/id_dsa/** using a passphrase to protect the file against usage of others.

The observed traffic is the same, because the exchange of the key happens as encrypted traffic which couldn't be observed.