

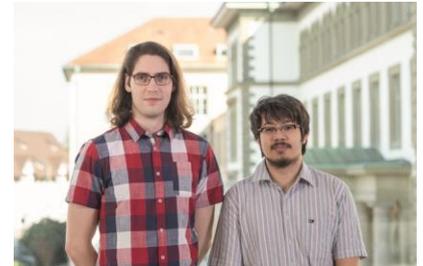
Design und Implementierung einer Alternative zu SSH/Design and Implementation of an Alternative to SSH

Die Secure Shell (SSH) ist eine alte Applikation, die viele Funktionen hat. Jedoch werden nicht alle dieser Funktionen im täglichen Betrieb genutzt. Diese Arbeit hat einen einfacheren Protokoll-Prototyp konzipiert und implementiert. Die entwickelte Lösung ist in der Lage, SSH in ihrer Kernfunktion zu ersetzen: Dem Verbinden zu einer Shell auf einem entfernten System.

Dieses Projekt entwickelte eine Applikation mit dem Namen 'oh-my-gosh', welche die gleiche Kernfunktionalität bietet wie SSH. Dies schliesst die Client-seitige Applikation 'gosh' sowie das Server-Gegenstück 'goshd' ein. Letztere wurde entworfen, um auf Linux-Systemen als Hintergrundprozess zu laufen. Diese Lösung nutzt einen gesicherten Kanal, welcher dem ganzen Prozess mehr Sicherheit gibt. Ein Anwender kann sich selbst auf dem entfernten System authentifizieren; entweder mit einem Passwort oder über Public Key Cryptography. Alle gängigen Arbeitsabläufe sind in der Shell möglich, wie zum Beispiel:

- Durch das Dateisystem navigieren
- Skripte und Applikationen ausführen
- Applikationen nutzen, welche ncurses(3X) nutzen

Die Lösung stützt sich auf einige Unix-spezifische Technologien wie Pseudoterminals und Pluggable Authentication Modules (PAM), welche auch in login(1) genutzt werden, um ihre Anwendungsfälle zu realisieren.



Diplomierende
Raphael Emberger
Kevin Satra Schwarz

Dozent
Stephan Neuhaus

```
test@kepler22b ~$ pud
/home/alan
test@kepler22b ~$ gosh test@localhost
[INFO][0000] Dialing server. host="localhost:222"
2* scheamtcp
[INFO][0000] Connection established. remote="127.0.0.1:222"
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
test@kepler22b test~$ pud
/testp/test
test@kepler22b test~$ echo "I'll be back"
I'll be back
test@kepler22b test~$ logout
test@kepler22b ~$

esberger@kepler22b ~$ sudo goshd
[INFO][0000] Listening on socket. fd=0 macSession=0
[INFO][0014] Accepted connection from peer. remote="127.0.0.1:34540"
[INFO][0014] Started child. pid=2104
[INFO][0000] Connected to client. idiom="0xc0001c3440" pid="0xc0001c2400"
[INFO][0000] Got all the information from the client.
[INFO][0000] Opened pseudo terminal.
2* pty: [12] pty="0xc0001c2400"; ptyFd[0]=12
[INFO][0000] Set up host.
[INFO][0000] Encrypted secret. Sending to client. encryptedSecret="v"
[INFO][0000] client authenticated itself using keys. gid=1001 HOME="/tmp/"
[INFO][0000] Looked up user.
test SHELL=/bin/bash UID=1001 USER=test
[INFO][0000] Started shell.
test@kepler22b test~$ cd /tmp/
test@kepler22b test~$ pwd
/tmp
test@kepler22b test~$ ls -la
total 4
drwxr-xr-x 2 test test 4096 Sep  9 10:59 .
drwxr-xr-x 1 test test 4096 Sep  9 10:59 ..
-rw-r--r-- 1 test test 0 Sep  9 10:59 .bash_history
-rw-r--r-- 1 test test 0 Sep  9 10:59 .bashrc
-rw-r--r-- 1 test test 0 Sep  9 10:59 .profile
test@kepler22b test~$ echo "Failed to write from pty to client."
Failed to write from pty to client.
test@kepler22b test~$ echo "ent"
ent
ent: error="read /dev/ptmx: input/output error"
0
```

Gosh während des Betriebs