

ssh-break.sh

```
#!/bin/bash
```

```
# scannt in diesem Beispiel den Netzwerkbereich 192.168.10.1 bis 192.168.20.255 via SSH mit dem User root  
und dem Passwort <enter password to check for>
```

```
# (<enter password to check for> durch gewuenshtes Passwort ersetzen!)
```

```
# von
```

```
A1="192."
```

```
B1="168."
```

```
C1="10."
```

```
D1="1"
```

```
# bis
```

```
A2="192."
```

```
B2="168."
```

```
C2="20."
```

```
D2="255"
```

```
# Passwort
```

```
PASS="<enter password to check for>"
```

```
PROGDIR=`dirname $0`
```

```
rm -f $PROGDIR/login.expect
```

```
touch $PROGDIR/login.expect
```

```
chmod u+x $PROGDIR/login.expect
```

```
for a in `seq $A1 $A2`
```

```
do
```

```
  for b in `seq $B1 $B2`
```

```
  do
```

```
    for c in `seq $C1 $C2`
```

```
    do
```

```
      for d in `seq $D1 $D2`
```

```
      do
```

```
        trap 'exit 0' 2
```

```
        echo "Teste $a.$b.$c.$d:"
```

```

cat << EOF > $PROGDIR/login.expect
#!/usr/bin/expect

spawn ssh -o PubkeyAuthentication=no -o ConnectTimeout=1 -o NumberOfPasswordPrompts=1
root@$a.$b.$c.$d "uptime"
expect {
    password: {
        sleep 1
        send "$PASS\r"
        exp_continue
    } "connecting (yes/no)?" {
        send "yes\r"
        exp_continue
    } incorrect {
        send_user "invalid password or account\n"
        exit
    } timeout {
        send_user "connection timed out\n"
        exit
    } eof {
        exit
    }
}
EOF

    $PROGDIR/login.expect
    echo -e
    "#####
    #####"
    done
    done
    done
    done

rm -f $PROGDIR/login.expect

```

Revision #1

Created 27 July 2021 10:11:15 by magenbrot

Updated 27 July 2021 10:11:28 by magenbrot