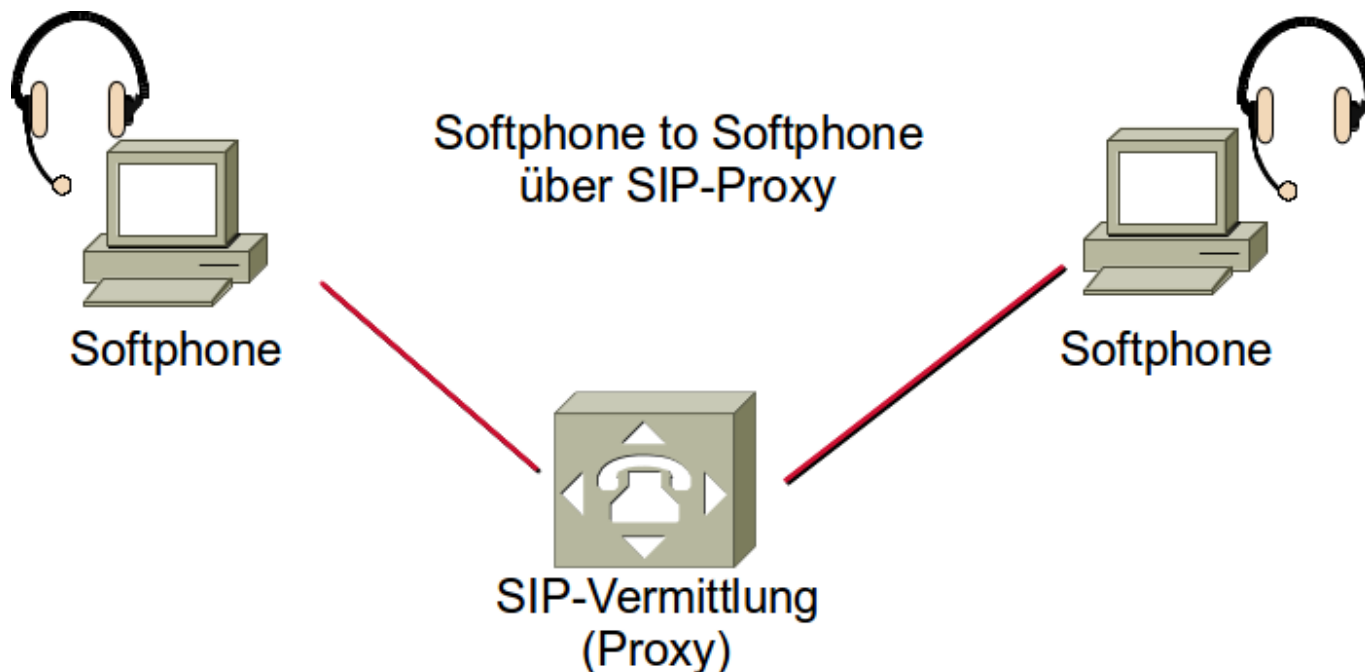


# Laborübung zu VoIP mit sipwitch

Für die Übung zu VoIP wird der SIP-Proxy sipwitch/sipw als vermittelndes Element eingesetzt. Im folgenden ist ein kurzes Howto inklusive einer Beispielkonfiguration, die an die eigenen Bedürfnisse anzupassen ist.



## Vorbereitung alle SIP-fähigen Prozesse stoppen

Falls der Proxy bereits beim Systemstart als Dienst (Daemon) mitgestartet wird, muss der Proxy zunächst gestoppt werden. Mit dem folgenden Befehl lässt sich überprüfen, ob es einen entsprechenden Daemon-Prozess gibt.

```
ps aux | grep sip
```

Der SIP-Proxy kann mit dem folgenden Befehl gestoppt werden

```
sudo sipwitch down
```

Dieser Schritt ist ggf. auch auf den Client-PCs durchzuführen. Bei den Clients ist zusätzlich zu beachten, dass nur EINE VoIP-Software aktiv ist (Server oder Client), da sonst ebenfalls der Port für die zweite Anwendung blockiert ist.

## SIP-Proxy-Konfiguration

Der SIP-Proxy sipwitch wird normalerweise über zwei getrennt Konfigurationsdateien konfiguriert. Die Standard-Konfiguration liegt unter `/etc/sipwitch.conf`<sup>1)</sup> und für die Anlage der Benutzer `/etc/sipwitch.d/lab.xml-sample`<sup>2)</sup>. Diese Konfiguration wird verwendet, wenn der Proxy mit

root-Rechten (sudo) gestartet wird.

**ACHTUNG** Standard-Konfigurationsdateien sind schlecht bzw. fehlerhaft kommentiert. Insbesondere ist der `<realm>`-Tag komplett falsch angegeben. Er darf KEINE Leerzeichen und MUSS mindestens einen Punkt enthalten. Aus diesem Grund sollte man eigene Konfigurationsdateien verwenden. Die unten angeführte Beispielkonfiguration ist entsprechend überarbeitet und kann über den Nutzer keinpassword verwendet werden.

Die unten aufgeführte Datei kann als Beispiel für einen eigenen Aufbau genutzt werden. Dazu muss die Datei in `/home/USER/.sipwitchrc` abgelegt werden und der Proxy mit der folgenden Befehlszeile (ohne sudo !!) gestartet werden:

```
sipw -x9 -f
```

Ein Erklärung der einzelnen XML-Tags findet sich auf der Homepage des [GNU-Telephony-Projekts](#).

HINWEIS: Der Begriff `Extension` steht im Bereich Telefonie für Rufnummern bzw. Nebenstellen. Damit sind also die Rufnummern der einzelnen Teilnehmer gemeint.

Die folgende Beispieldatei basiert auf den offiziellen Dateien, muss aber noch an die örtlichen Gegebenheiten angepasst werden. (Per Klick auf [.sipwitchrc\\_sample](#) kann die Datei direkt heruntergeladen werden.)

[.sipwitchrc\\_sample](#)

```
<?xml version="1.0"?>
<sipwitch>
  <!-- master config file. The default config can be overridden with
a
  runtime one stored in /var/run/sipwitch which can be installed by
  a management system. If one is using a server executed under "user"
  permissions, then this would be ~/.sipwitchrc.
  Erklärung:
  http://www.gnutelephony.org/index.php/GNU_SIP_Witch_configuration

  Konfig: .sipwitchrc in Home-Ordner kopieren.
  Aufruf: sipw -x9 -f
  -->
  <provision>
  <!-- Allows provisioning to be in main config file as well as
  scattered.
  This allows one to produce a single config file that represents
  the
  complete phone system.

  <refer id="x"></refer>
  <alias id="test"><contact>sip:xxx@yyy</contact></alias>
  <user id="y"/>
  <gateway id="z"/>
  -->
  <test id="testing">
```

```

    <secret>EditMe</secret>
    <extension>299</extension>
    <answer>12</answer>
    <duration>120</duration>
    <display>Testing</display>
</test>

<user id="EditMe1">
    <secret>EditMe</secret>
    <extension>201</extension>
    <display>EditMe1</display>
</user>

<user id="EditMe2">
    <secret>1234</secret>
    <extension>202</extension>
    <display>editme 2</display>
</user>

</provision>

<access>
    <!-- Access rules and cidr definitions. By default 127.0.0.1:::1
are in
    a pre-generated "loopback" cidr. Access rule entries are now
    automatically generated by scanning the network interface, so this
    is for special overrides or convenience naming.
<local>172.16.59.0/24</local>

<local>0.0.0.0/01</local>
<local>128.0.0.0/01</local>
    Hier gehört das lokale Netz, in dem der Server betrieben wird
hinein.
-->
    <local>EditMe</local>
</access>

<stack>
    <!-- The effective names this server processes requests for, and an
optional
    list of host or domain names this server will also respond to.
The
    default hostname is always accepted.
    <localnames>sip.gnutelephony.org, server.local, something
somewhere</localnames>
-->

<!-- Stack configuration. Here we restrict all access to the server
under
    the local subnet, and we specify the local subnet is "trusted".
Trusted

```

```
means that challenge digests will be relaxed for devices that are
already registered with the server, and hence reduces the total
sip
traffic needed. We map for 200 calls, set 2 dispatch threads for
sip events, and bind to all interfaces.
<restricted>local</restricted>
<trusted>local</trusted>

-->

<mapped>100</mapped>
<threading>2</threading>
<interface>*</interface>
<dumping>>false</dumping>

<!-- peering entry used for setting "proxy" ip address for external
users
when we are behind a NAT. This is used for determining ip address
for
media proxy in particular. Example entry shown. Can be ip address
or
resolvable hostname.

<peering>www.example.com</peering>
-->

<!-- special user id's. The "system" id is used when the server
creates a
sip message that is not on behalf of any registered "ua", but
rather
from the server itself. For example, when feeding a sms "message"
through the control interface, this is generated as a "system"
message.
Attempts to dial the "system" id will always return SIP FORBIDDEN.

The "anon" id is used when anonymous messages are generated. These
always respond with SIP NOT FOUND if one wishes to contact anon.
-->

<system>
system
</system>
<anon>
anonymous
</anon>
</stack>

<timers>
<!-- ring every 4 seconds -->
<ring>4</ring>
<!-- call forward no answer after x rings -->
<cfna>4</cfna>
```

```
<!-- call reset to clear cid in stack, 6 seconds -->
<reset>6</reset>
</timers>

<!--
  we have 2xx numbers plus space for external users
  Die <extension> aus dem <provision>-Teil müssen im prefix-Bereich
  liegen!
  Beispiel: <extension>=212 -> <prefix>=200
-->
<registry>
  <prefix>200</prefix>
  <range>100</range>
  <keysize>77</keysize>
  <mapped>100</mapped>

  <!-- ACHTUNG: Der Tag <realm> muss mind. einen "." enthalten und
  darf keine
  Leerzeichen beinhalten.!
  s. Hier:
  http://lists.gnu.org/archive/html/sipwitch-devel/2011-01/msg00007.html
-->
  <realm>EditMe.de</realm>
</registry>

<routing>

</routing>
</sipwitch>
```

## SIP-Client Softphone Linphone

Als Softphone wird die Software Linphone verwendet. Bei der Konfiguration muss beachtet werden, dass im gesamten Szenario OHNE Domänen (z.B. [www.irgendwas.de](http://www.irgendwas.de)) gearbeitet wird. Daher muss als Registrar die IP-Adresse des SIP-Proxy angegeben werden. Beispiel:

SIP-Proxy läuft auf 192.168.21.120 → user01@192.168.21.120 D.h. der Benutzer user01 kann sich mit seinem Passwort auf dem Proxy 192.168.21.120 anmelden. Zusätzlich sollte als Registrar ebenfalls der SIP-Proxy angegeben werden.

Wenn man einen anderen Benutzer auf demselben Proxy anrufen möchte, so reicht die Rufnummer (Extension) des Ziels.

## Aufgaben

Protokollieren Sie den folgende Vorgänge mit:

1. Anmeldung der Benutzer (einmal korrekt, einmal mit falschem Passwort, einmal mit falschem Benutzernamen)
2. Verbindungsaufbau (einmal zu korrektem Ziel, einem zu nicht existierendem Ziel)
3. Wiederanmeldung (Reregister) der Benutzer

Achten Sie insbesondere auf die Statusmeldungen.

1)

Basiskonfiguration

2)

Beispiel für Benutzerkonfiguration -> `<provision>`

From:

<http://www.kopfload.de/> - **kopfload - Lad Dein Hirn auf!**

Permanent link:

[http://www.kopfload.de/doku.php?id=lager:oeff\\_netze:voip&rev=1530777838](http://www.kopfload.de/doku.php?id=lager:oeff_netze:voip&rev=1530777838)

Last update: **2025/11/19 16:13**

