

# OPENSIPS AS SIP-SERVER FOR VIDEO CONFERENCE SYSTEMS

[PDF-Version \(optimized for print\)](#)

## GENERAL



## PERIOD

October 2016 - July 2017

## SW VERSION

The test was made with the software version 2.2.4 of OpenSIPS. At the end of the testing period was a more stable version (2.3.0) available, which was not used for reasons of comparability.

## DEVICE CLASS

OpenSIPS stands for Open SIP Server, i.e. OpenSIPS is a open source implementation of a SIP-Server. Thus it is a part of the client server architecture of SIP protocols. The online presence of OpenSIPS is available at <http://www.opensips.org>.

## FUNCTIONALITIES

OpenSIPS includes numerous functionalities, for example SIP registrar, SIP router / proxy, SIP redirect server and much more. A complete list can be found at <http://opensips.org/About/Features>. OpenSIPS serves as central structure, for example the management of SIP devices, the simplification of dialstrings or the spam protection. There were wide-ranging adjustment possibilities by prefabricated modules and a own script language.

## HARDWARE

OpenSIPS runs on a Linux based system. Debian 8 ("jessie" ) was used in this test.

## INSTALLATION / CONFIGURATION

### REPOSITORY VS. MANUAL INSTALLATION

It should be noted that the increased effort of the installation / configuration at Open Source always has to be mentioned. Therefore the usage of the official repositories to receive the packages is advisable. The officially available repositories are for Debian/Ubuntu, Redhat/CentOS/Fedora. Of course the repositories are equipped with a key, which should be imported before to ensure the authenticity of the packages. A great benefit at the usage of repositories is the possibility to install individual modules as packages easily in the follow-up without compiling everything again.

Alternatively, or if a repository can not be used, tarballs or checkout via GIT/SVN are available. The compiling via the supplied tool **menuconfig** (launchable by **make menuconfig**) is very possible. It may be both Compile Flags (for example support for IPv6, TCP, TLS) or the to be used modules (for example database linkage for MySQL, Presence, XMPP-support) can be choosed. Of course the reliance to the other (external) packages have to be dissolved manually. In general necessary are for example the development library by **ncurses** (**libncurses5-dev**) and the packages **flex**, **bison** and **m4**.

### BASIC SETTINGS

First of all, it has to be ensured that the Linux-based system possess correct settings for the DNS. Thus, the values **/etc/hosts**, **/etc/hostname** and **/etc/resolv.conf** have to be checked and if necessary, changed. The most important paths to the configuration files are (at installation via a repository):

- **/etc/opensips/**
- **/etc/default/**
- **/etc/init.d/**

In both of the files **/etc/opensips/opensipsctlrc** and **/etc/opensips/osipsconsolerc** are a series of fundamental settings with default values. Important is, to configure the desired SIP domain and the database connection with the relevant login details. The database and the database-user do not have to exist and will be created later (see the next section "database"). In the test, the SIP-domain **osips.vcc.test.tu-dresden.de** and as database MySQL were used on the same Linux-system. Further settings were left on the default values.

### DATABASE

By means of the tool **opensipsdbctl create** the database, the database-user and the appropriate permissions will be created in accordance with requirements of **/etc/opensips/opensipsctlrc**. It should be noted here that at the usage of OpenSIPS Control Panels (see relevant section) further tables in the database have to be added.

### CONFIGURATION FILE / SCRIPT SET

A central configuration file is existing for OpenSIPS under **/etc/opensips/opensips.cfg**, which contains for example to be used IP-addresses, a register of to be used modules, settings for the modules (inter alia, database connection, filepaths) and the setting for the logging. An essential component is the routing-script, which will go through every request for example register-request or invite-request and which can be customized very precisely to the needed requirements.



The preparation of such a configuration file can be done by the tool **osipsconfig**. It generates depending on the desired purpose and a first list of requirements (see the black box on the right side) a configuration-/script template. The template has to be changed to the local conditions by setting up the specific positions, which are marked with **# CUSTOMIZE ME**. Affected are, for example the IP-address, the database connection and the filepaths.

```
listen=udp:141.30.abc.xyz:5060 # CUSTOMIZE ME
listen=tcp:141.30.abc.xyz:5060 # CUSTOMIZE ME
listen=tls:141.30.abc.xyz:5061 # CUSTOMIZE ME
```

It is recommended to indicate the module domain, to distinguish particularly the local domains:

```
##### Modules Section #####
loadmodule "domain.so"
modparam("domain", "db_url", "mysql://user:pw@host/db-name")
modparam("domain", "db_mode", 0) # No caching for debug
modparam("domain", "domain_table", "domain")
modparam("domain", "domain_col", "domain")
```

## LOG FILES

Until the productive operation it is very valuable to have informations in the log files as many as possible, because regularly the few bugs and problems can not be seen and be bypassed in advance. It is therefore sensible to do the following entry to the file **/etc/opensips/opensips.cfg**:

```
##### Global Parameters #####
log_level=4
log_stderr=no
log_facility=LOG_LOCAL1
```

Furthermore there has to be indicated on the Linux-based system what local1 exactly means. This is why the entry in **/etc/rsyslog.conf** is needed, to redirect the issues into the file:

```
local1.* -/var/log/opensips.log
```

**Rsyslog** has to be restarted afterwards.

## START AS SERVICE VS. MANUAL START

For a first testing, a manual start is still practicable, but the start as service should be preferred at a frequent usage of OpenSIPS. For the manual start of OpenSIPS as service, two files have to be changed:

- **/etc/defaults/opensips**: Perhaps adjustments at user (root) and execution (yes) necessary
- **/etc/init.d/opensips**: Adjustments at the filepaths may be necessary

The usage take place by **/etc/init.d/opensips {start|stop|restart|force-reload|status}**. Especially with **/etc/init.d/opensips status** a first analysis of problems is may be possible.

## DNS RECORDS

To be found by SIP-devices via the SIP-domain, DNS-records are necessary for OpenSIPS. In this case, direct A-records and also SRV-records make sense:

- A-record: SIP-domain;ne => IP-address from OpenSIPS
- SRV-records: **{\_sip|\_sips}.\_{tcp|udp|tls}.SIP-domain;ne=> DNS-name from OpenSIPS, Port {5060|5061}**

Not all combinations are reasonable at the indicated SRV-records. The choosing options are coming from the possible encoding and the transport protocol. Useful variants for `_tcp` are for example just:

- `_sip._tcp.SIP-Domäne => DNS-Name von OpenSIPS, Port 5060`
- `_sips._tcp.SIP-Domäne => DNS-Name von OpenSIPS, Port 5061`

It is recommendable to think about the appropriate reserve DNS-entries and to check the DNS-entries for the SIP-devices.

## Operation

### OPENSIPS CONTROL PANEL

The OpenSIPS Control Panel is a php based graphical user interface. OpenSIPS Control Panel is developed separately and is available under <http://controlpanel.opensips.org/> as Tarball or SVN Checkout.

There are some package dependencies, so **libapache2-mod-php5**, **php5**, **php5-cli**, **php5-gd**, **php5-mysql**, **php-pear**, **php5-xmllrpc** have to be installed. The OpenSIPS Control Panel needs also some other modules which can be installed via **pear**: **pear install MDB2**, **pear install MDB2#mysql**, **pear install log**

The integration into Apache2 happens for example via Alias- and Directory-statements at `/etc/apache2/apache2.conf`:

```
### OpenSips Control Panel ###
<Directory /somewhere/opensips-cp/web>
Options Indexes FollowSymLinks MultiViews
AllowOverride None
</Directory>

<Directory /somewhere/opensips-cp>
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Require ip 141.30.abc.xyz/32
</Directory>

Alias /opensips-cp /somewhere/opensips-cp/web
```

With these settings OpenSIPS Control Panel can be achieved under `host/opensips-cp`, in which `host` the DNS name of the computer is, where OpenSIPS is setted up. During testing it was the same computer on which OpenSIPS runs. It should be noted that, the files of OpenSIPS Control Panel got enough rights (for example `www-data`) and Apache2 has to be reloaded at the end.

A further important point is the necessity of the setting

```
short_open_tag=On
```

into the file **php.ini**, so that the php-scripts work.

Moreover, diverse extensions of the existing database on the computer with OpenSIPS are required. There are therefore prefabricated tables **ocp\_admin\_privileges.mysql**, **cdrs.mysql** and **tables.mysql**, which have to be inserted into the data base of OpenSIPS. The first administrativ user of OpenSIPS Control Panel will be registered at the table **ocp\_admin\_privileges** via

```
INSERT INTO ocp_admin_privileges values('admin','password',md5('admin:password'),'all','all')
```

Besides the adjustments of the database, the database connection at **config/db.inc.php** and **config/boxes.global.inc.php** at the OpenSIPS fifo-queue have to be specified at the OpenSIPS Control Panel. The OpenSIPS fifo-queue will be defined at **/etc/opensips/opensips.cfg** under the fifo management.

As the last set-up point, the entry at **/etc/crontab** is senseful for the statics:

```
***** root cd /somewhere/opensips-cp/cron_job/; php get_opensips_stats.php > /dev/  
null
```

The OpenSIPS Control Panel can be used in many different ways. So is, inter alia, the entry of SIP domains, the creation of accesses / accounts, the notification of active and past calls and the direct access on various modules / tables of the data base possible. As the SIP domain, the related ip-address should be also indicated every time (see the screenshot below). Here it should be noted, that not all files of the database / modules are depicted in the OpenSIPS Control Panel, but a few menu items can be programmed.

A debugging is without a direct data base access and access on the log-files from OpenSIPS hardly possible.

## SETTINGS AT THE SIP DEVICE

The necessary settings at the SIP device remain within limits as expected. It is to be noted, that the devices which control another protocols (for example H.323), first of all activate SIP as protocol. Also the "eavesdropping" on port 5060 or 5061 for incoming calls can be missed quickly.

The necessary settings at the SIP device are including:

- Selection of the transport-protocols (TCP, UDP, TLS)
- Specification of URI (name@SIP domain), for example vcc-ex90@osips.vcc.test.tu-dresden.de
- Specification of the Login with the generate credentials
- Specification of the SIP proxy (DNS name of OpenSIPS)

## CONTROL OPTIONS / TROUBLESHOOTING

There are a lot of possibilities and places where errors / problems can be found in connection with OpenSIPS:

- At the SIP device: Depending on the device, it will for example show notifications to the registration status on the web interface.
- Via **netstat -tulpn | grep opensips**: Testing, if OpenSIPS is waiting at the correct port for incoming SIP Requests.
- At the logs: In **/var/logs/opensips.log** a lot of processes can be traced at a high log level. There are also the used modules / functions for each case:
  
- At the OpenSIPS Control Panel: There is the possibility for example, that running calls in the subitem dialogue and finished calls in the subitem CDR Viewer may be reviewed:



- At the database: At the database of OpenSIPS are for example failed calls in the table missedCalls and registered SIP devices can be found in the table location.
- Via the command-line tool: Serverstatics are accessible via **opensipsctl monitor** and registered SIP devices via **opensipsctl ul show** or **opensipsctl online**:

## CALL POSSIBILITIES

In principle, the calls have to be devided in accordance with the registration status of the involved SIP devices and in accordance with used transport protocol.

On the sending and on the receiving site of the call, TCP / UDP or TLS can be used in each case, if OpenSIPS is configured accordingly. In this case 9 variants are conceivable, whereby OpenSIPS is assuming every time the possible necessary log switching in the signalling.

Regarding the registration status 3 variants are conceivable:

- "internal" to "internal": In this case, only the at OpenSIPS registered SIP devices are involved. Calls are possible with the speed dial of the username or the alias (SIP devices are adding the own SIP domain). Of course calls are also possible via **name/alias@SIP-domain**.
- "internal" to "external": Calls to distant remote stations can be done via **uri@other-sip-domain**.
- "external" to "internal": Distant remote stations don` t have a speed dial option, so they have call every time **name/alias@SIP-domain**.

---

## USAGE OF TLS / CERTIFICATE MANAGEMENT

To use encoded signalling via TLS, a according module has to be configured at OpenSIPS and there has to be created at least one certificate.

The certificates will be applied at DFN-PKI. To that, it is necessary to create a certifacte request for each case via **openssl**. The requirements to the certificate should be taken account:

Certifcate for OpenSIPS:

- Certificate profile: VoIP server
- Clear name: DNS name OpenSIPS
- Alternative DNS name OpenSIPS and SIP domain

Certificates for SIP devices:

- Certificates are not absolutely necessary or it is also possible to use the mostly available build-in "standard certificates".
- Certificates have the analogue requirements like at OpenSIPS, except that only the DNS name of the SIP device will be used.

It is important, that the DFN-PKI master certificate at OpenSIPS and the SIP devices will be lodged as trustworthy CA so that a validation can be performed.



The involvement of the certificates takes place at **/etc/opensips/opensips.cfg**. Potentially available lines with `modparam("proto_tls", ...)` have to be commented out, because meanwhile everything is located at the module `tls_mgm`:

```
##### Modules Section #####
loadmodule "proto_tls.so"
## manuell auskommentiert, da Parameter nicht verfuegbar:
## modparam("proto_tls","verify_cert", "1")
## modparam("proto_tls","require_cert", "0")
## modparam("proto_tls","tls_method", "TLSv1")
## modparam("proto_tls","certificate", "/etc/opensips/tls/user/user-cert.pem")
## modparam("proto_tls","private_key", "/etc/opensips/tls/user/user-privkey.pem")
## modparam("proto_tls","ca_list", "/etc/opensips/tls/user/user-calist.pem")

## manuell hinzugefuegt:
loadmodule "tls_mgm.so"

modparam("tls_mgm","db_url", "mysql://user:pw@host/db-name")

modparam("tls_mgm","verify_cert", "1")
modparam("tls_mgm","require_cert", "1")
modparam("tls_mgm","tls_method", "TLSv1")
modparam("tls_mgm","certificate", "/etc/opensips/tls/user-dfnpki/user-cert.pem")
modparam("tls_mgm","private_key", "/etc/opensips/tls/user-dfnpki/user-privkey-
new.pem")
modparam("tls_mgm","ca_list", "/etc/opensips/tls/user-dfnpki/user-calist.pem")
modparam("tls_mgm","ca_dir", "/etc/opensips/tls/user-dfnpki/")
```

The file with the private key should be available without passphrase otherwise a start as service is not possible because there can't be entered a password.

If there are not enough certificates for all SIP devices (`require_cert`) or they are not verifiable (`verify_cert`), the corresponding switch has to be set on "0". The review of the SIP devices is not possible via the certificate but the connection establishment is encoded.

Because the DFN-PKI certificates are slightly bigger, the devices require during testing more than 100 ms at the TLS handshake with OpenSIPS. Within this, there were accidental disconnections at the call establishment. Thus, the following parameters have to be changed:

```
modparam("tls_mgm","tls_send_timeout", zeitwert)
modparam("tls_mgm","tls_handshake_timeout", zeitwert)
```

TLS connection will be distinguished between incoming TLS connections to OpenSIPS (`server_domain`) and outgoing TLS connections from OpenSIPS (`client_domain`). Depending on the socket (IP address: Port) different settings can be made:

```
modparam("tls_mgm","server_domain","1=141.30.67.218:5061")
modparam("tls_mgm","verify_cert", "1:1")
modparam("tls_mgm","require_cert", "1:1")
modparam("tls_mgm","tls_method", "1:TLSv1")
modparam("tls_mgm","certificate", "1:/etc/opensips/tls/user-dfnpki/user-cert.pem")
modparam("tls_mgm","private_key", "1:/etc/opensips/tls/user-dfnpki/user-privkey-
new.pem")
modparam("tls_mgm","ca_list", "1:/etc/opensips/tls/user-dfnpki/user-calist.pem")
modparam("tls_mgm","ca_dir", "1:/etc/opensips/tls/user-dfnpki/")

modparam("tls_mgm","client_domain","2=141.30.67.247:5061")
modparam("tls_mgm","verify_cert", "2:1")
modparam("tls_mgm","require_cert", "2:1")
modparam("tls_mgm","tls_method", "2:TLSv1")
modparam("tls_mgm","certificate", "2:/etc/opensips/tls/user-dfnpki/user-cert.pem")
modparam("tls_mgm","private_key", "2:/etc/opensips/tls/user-dfnpki/user-privkey-
new.pem")
modparam("tls_mgm","ca_list", "2:/etc/opensips/tls/user-dfnpki/user-calist.pem")
modparam("tls_mgm","ca_dir", "2:/etc/opensips/tls/user-dfnpki/")
```

These distinctions are particularly useful, if OpenSIPS has to manage multiple SIP domains and can be contacted via different IP addresses. Then it will be ensured, that the used certificate also fits to the SIP domain, which is currently used.

## USAGE OF OTHER MODULES / SCRIPTING

This section describes in an exemplary the procedure how OpenSIPS can be adapted to the own needs. Helpful for the scripting are the documentations of the available variables, functions and statements:

- <https://www.opensips.org/Documentation/Script-CoreVar-2-2>
- <https://www.opensips.org/Documentation/Script-CoreFunctions-2-2>
- <https://www.opensips.org/Documentation/Script-Statements-2-2>

## DOMAIN MODULE

The domain module was already inserted in the chapter configuration file / script set at **/etc/opensips/opensips.cfg**:

```
##### Modules Section #####  
loadmodule "domain.so"  
modparam("domain", "db_url", "mysql://user:pw@host/db-name")  
modparam("domain", "db_mode", 0) # No caching for debug  
modparam("domain", "domain_table", "domain")  
modparam("domain", "domain_col", "domain")
```

It allows to enter SIP domains in the data base or via the OpenSIPS Control Panel. The indicated SIP domains will be recognised as "local". Each call will be tested, that at least one site is local (caller, callee).

## DIALPLAN MODULE

The module dialplan enables the transcription of strings, typically the request URI. It controls the string matching (matching operator = 0) and the regex matching (matching operator = 1). The indication of the rules are made via the database or the OpenSIPS Control Panel:

The usage of **/etc/opensips/opensips.cfg** at the routing script is made via **dp\_translate ("0","\$ru/\$ru");**

In that connection, \$ru stands for request URI and indicates the first value that all rules have to be controlled with dialplan ID = 0.

For example for the usage of the first rule is here to see (extract from the log file):

## PROTECTION OF THE SIP DEVICES/ MEASURES FOR THE SPAM PROTECTION

It is common ground that unprotected SIP devices fall victim to SIP spam calls which also become more as at H.323. Targets of the attackers are mostly the free cost usage of the telephone or the generating of high costs via the infiltrated VoIP platform. At video conference systems the expense is mostly not given (except there is a connection into the ISDN world), but the SIP spam calls are disturbing or blocking the normal communication substantially. To be protected, a simple spam protection could be established with OpenSIPS.

To begin with, the SIP devices should be not longer directly reachable from "outside" via the Port 5060/ 5061 after the registration at OpenSIPS, as otherwise OpenSIPS could be bypassed at the signalling. At some devices it can be directly stopped, otherwise it has to be realized by the help of firewalls, ACL`s etc. Therefore the signalling via OpenSIPS is enforced, the media streams (RTP / SRTP) will continue to be established directly via dynamically negotiated ports between the SIP devices.



The majority of the SIP spam calls have the form numbers@ip-address as To-URI. These will continue to come at OpenSIPS and could be transmitted to the appropriate device. That is why for example a block of the SIP spam calls with the IP address as SIP domain is possible. The necessary lines in the routing script from `/etc/opensips/opensips.cfg` were as follows:

```
if ($td == "141.30.67.218") {  
  xlog("Dropped Spam-Call to $tu\n");  
  exit;  
}
```

At this point \$td is the SIP domain at the To-URI (\$tu). If OpenSIPS is reachable via multiple IP addresses, the term has to be correspondingly adjusted. Of course, more aspects of the SIP spam calls can be considered, but during the test the simple spam protection showed already a sufficient effect:

In this scenario the regular calls have to be made by **name/alias@SIP-Domäne**, whereby the SIP domain is a DNS name.

---

## CONCLUSION

The installation / configuration of OpenSIPS is unfortunately not self-explanatory, but the documentation is well suited. The way to the operational capability of OpenSIPS requires still a lot of time, especially because the method "trial & error" has to be used many times.

The separate OpenSIPS Control Panel facilitates the typical operations, but it is not a complete replacement for the database access or the work with the console.

Many entities, where the mistakes can be found require a certain overview and thus appropriate training period.

The certificate management can be used effectively, especially because the certificates can be taken up via the DFN-PKI easy, fast and reliable.

The induction into the script language, work with the console etc. is absolutely essential. Numerous modules and the possibility of self-development offers a great scope. For example is a simple spam protection for SIP devices viable .

```
DBG:core:parse_msg: SIP Request:
DBG:core:parse_msg: method: <INVITE>
DBG:core:parse_msg: uri: <sip:vcc-c40@141.30.67.218>
DBG:core:parse_msg: version: <SIP/2.0>
DBG:core:parse_headers: flags=2
DBG:core:parse_via_param: found param type 232, <branch> = <z9hG4bKa1959b67
DBG:core:parse_via_param: found param type 235, <rport> = <n/a>; state=17
DBG:core:parse_via: end of header reached, state=5
DBG:core:parse_headers: via found, flags=2
DBG:core:parse_headers: this is the first via
DBG:core:receive_msg: After parse_msg...
DBG:core:receive_msg: preparing to run routing scripts...
DBG:core:parse_headers: flags=8
DBG:core:get_hdr_field: cseq <CSeq>: <100> <INVITE>
DBG:core:parse_to: end of header reached, state=10
DBG:core:parse_to: display={}, ruri={sip:vcc-c40@141.30.67.218}
DBG:core:get_hdr_field: <To> [29]; uri=[sip:vcc-c40@141.30.67.218]
DBG:core:get_hdr_field: to body [<sip:vcc-c40@141.30.67.218>#015#012]
DBG:core:comp_scriptvar: str 20 : 141.30.67.218
Dropped Spam-Call to sip:vcc-c40@141.30.67.218
```

```
DBG:dialplan:dp_translate_f: dpid is 0 partition is default
DBG:dialplan:dp_get_svalue: searching 14
DBG:dialplan:dp_translate_f: input is sip:97924988@osips.vcc.test.tu-dresden.de
DBG:dialplan:dp_translate_f: Checking with dpid 0
DBG:dialplan:test_match: test_match:[0] sip:97924988@osips.vcc.test.tu-dresden.de
DBG:dialplan:test_match: test_match:[1] @osips.vcc.test.tu-dresden.de
DBG:dialplan:translate: Regex operator testing. Got result: 0
DBG:dialplan:translate: Found a matching rule 0x7f479f9b1818: pr 5, match_exp sip:979[0-9]+(@osips.vcc.test.tu-dresden.de){0,1}
DBG:dialplan:test_match: test_match:[0] sip:97924988@osips.vcc.test.tu-dresden.de
DBG:dialplan:test_match: test_match:[1] 24988
DBG:dialplan:test_match: test_match:[2] @osips.vcc.test.tu-dresden.de
DBG:dialplan:dp_translate_f: input sip:97924988@osips.vcc.test.tu-dresden.de with dpid 0 => output sip:97924988@vc.dfn.de
```

Dialplan ID	Rule Priority	Matching Operator	Matching Regular Expression	Matching Flags	Substitution Regular Expression	Replacement Expression	Attributes	Edit	Delete	Clone
0	5	1	sip:979[0-9]+(@osips.vcc.test.tu-dresden.de){0,1}	0	sip:979([0-9]+(@osips.vcc.test.tu-dresden.de){0,1})	sip:979\1@vc.dfn.de				
0	6	1	sip:[0-9]{1,5}(@vc.dfn.de)	0	sip:([0-9]{1,5})(@vc.dfn.de)	sip:979\1@vc.dfn.de				

Page: 1 Total Records: 2



```
root@rncmm0-218:~# opensipsctl u1 show
Domain:: location table=512 records=1
AOR:: vcc-ex90
    Contact:: sip:vcc-ex90@141.30.67.247:5060 Q=
        Expires:: 3370
        Callid:: d02faa1d1a93dbff23d0e3671e24d03e
        Cseq:: 52570
        User-agent:: TANDBERG/520 (TC7.1.1.168aadf)
        State:: CS_SYNC
        Flags:: 0
        Cflags::
        Socket:: udp:141.30.67.218:5060
        Methods:: 5247
        SIP_instance:: <urn:uuid:ba218525-db0b-51a9-aaf1-fb57167f8e9e>
root@rncmm0-218:~# _
```

```
[cycle #: 10; if constant make sure server lives]
Server:: OpenSIPS (2.2.1 (x86_64/linux))
Now:: Mon Dec 12 16:17:44 2016
Up since:: Mon Dec 12 16:09:46 2016
Up time:: 478 [sec]

Transaction Statistics:
tm:UAS_transactions:: 0
tm:UAC_transactions:: 0
tm:inuse_transactions:: 0

Stateless Server Statistics:
sl:sent_replies:: 7
sl:sent_err_replies:: 0
sl:received_ACKs:: 0

UsrLoc Stats:
usrloc:registered_users:: 1
usrloc:location-users:: 1
usrloc:location-contacts:: 1
usrloc:location-expires:: 0
```

OpenSIPS Control Panel Logout

System / CDRViewer / Read-Write

Search CDRs by:

CDR field: Time

Start Date: 2016 - 01 - 01 00 : 00 : 00

End Date: 2016 - 10 - 09 23 : 59 : 59

Search
Show All
Export

Time	Method	Sip Call ID	Sip Code	Sip Reason	Setup Time	Duration	Sip From Tag	Sip To Tag	Details
2016-10-09 20:05:20	BYE	<b>8d65996131f8491c43c1f8c7115e6056@0:0:0:0:0:0:0</b>	481	Call/Transaction Does Not Exist	0	0	3f2c072d	93c78de04b452e14	
2016-10-09 20:05:14	BYE	<b>8d65996131f8491c43c1f8c7115e6056@0:0:0:0:0:0:0</b>	200	Dialog Timeout	0	0	93c78de04b452e14	3f2c072d	
2016-10-09 20:03:49	INVITE	<b>8d65996131f8491c43c1f8c7115e6056@0:0:0:0:0:0:0</b>	200	OK	0	0	3f2c072d	93c78de04b452e14	
2016-10-09 19:43:48	BYE	<b>b1c2c5f704b875b6f0b8fcb707d346e9</b>	200	OK	0	0	e687c2f1cfa743e	8ad0b2102c352d75	
2016-10-09 19:43:36	INVITE	<b>b1c2c5f704b875b6f0b8fcb707d346e9</b>	200	OK	0	0	e687c2f1cfa743e	8ad0b2102c352d75	
2016-10-09 19:37:05	BYE	<b>d8b16f55cb917a9085b8d38d077a74bf</b>	200	OK	0	0	d9e4912810bf61ac	5853515bea99de54	
2016-10-09 19:36:52	INVITE	<b>d8b16f55cb917a9085b8d38d077a74bf</b>	200	OK	0	0	d9e4912810bf61ac	5853515bea99de54	
2016-10-09 19:33:09	BYE	<b>93bcd5ff51fdcdf9e57b1737672c08a</b>	200	OK	0	0	985cca129fab6c0b	7b181506980f44bb	
2016-10-09 19:32:55	INVITE	<b>93bcd5ff51fdcdf9e57b1737672c08a</b>	200	OK	0	0	985cca129fab6c0b	7b181506980f44bb	

Copyright © 2008-2018 OpenSIPS Project

OpenSIPS Control Panel

System / Dialog / Read-Write

Ongoing calls | Call profiles

Refresh Dialog List

Call ID	From URI	To URI	Start Time	State	Stop Call
8d65996131f8491c43c1f8c7115e6056@0:0:0:0:0:0:0	sip:vcc-ex90@osips.vcc.test.tu-dresden.de	sip:9791@vc.dfn.de	2016-10-09 20:03:49	Confirmed Call	

Page: 1 Total Records: 1

- Admin
- Users
- System
  - CDR Viewer
  - Callcenter
  - Clusterer
  - Dialog
  - Dialplan
  - Dispatcher
  - Domains
  - Dynamic Routing
  - Homer
  - Load Balancer
  - MI Commands
  - Monit
  - Permissions
  - RTPProxy
  - SIP Trace
  - Statistics Monitor
  - TViewer

```
DBG:core:parse_msg: SIP Request:
DBG:core:parse_msg: method: <INVITE>
DBG:core:parse_msg: uri: <sip:97924988@osips.vcc.test.tu-dresden.de>
DBG:core:parse_msg: version: <SIP/2.0>
DBG:core:parse_headers: flags=2
```

.: OpenSIPS Control Panel

Admin

Users

- ACL Management
- Alias Management
- User Management

System

Users / User Management / Read-Write

### Add New User

Username	<input type="text" value="vcc-ex90"/>
Domain	<input type="text" value="osips.vcc.test.tu-dresden.de"/>
Email	<input type="text" value="vcc@tu-dresden.de"/>
Alias Username	<input type="text" value="1247"/>
Alias Type	<input type="text" value="dbaliases"/>
Password	<input type="password" value="....."/>
Confirm Password	<input type="password" value="....."/>

[Go Main](#)

OpenSIPS Control Panel

System / Domains / Read-Write

Apply Changes to Server

**New Domain Name**

Add Domain

Domain Name	Last Modified	Edit	Delete
141.30.67.218	2017-06-29 12:30:03		
osips.vcc.test.tu-dresden.de	2016-10-07 16:24:33		

- Admin
- Users
- System**
  - CDR Viewer
  - Callcenter
  - Clusterer
  - Dialog
  - Dialplan
  - Dispatcher
  - Domains**
  - Dynamic Routing
  - Homer
  - Load Balancer
  - MI Commands
  - Monit
  - Permissions
  - RTPProxy
  - SIP Trace
  - Statistics Monitor
  - TViewer

**opensips**

OpenSIPS Control Panel

**LOGIN**

▶ Username

▶ Password



## Configure Residential Script

```
[*] ENABLE_TCP  
[*] ENABLE_TLS  
[*] USE_ALIASES  
[*] USE_AUTH  
[*] USE_DBACC  
[*] USE_DBUSRLOC  
[*] USE_DIALOG  
[ ] USE_MULTIDOMAIN  
[ ] USE_NAT  
[*] USE_PRESENCE  
[*] USE_DIALPLAN  
[*] VM_DIVERSION  
[ ] HAVE_INBOUND_PSTN  
[ ] HAVE_OUTBOUND_PSTN  
[ ] USE_DR_PSTN  
[*] USE_HTTP_MANAGEMENT_INTERFACE
```