OPENSIPS ALS SIP-SERVER FÜR VIDEOKONFERENZSYSTEME

PDF-Version (druckoptimiert)

ALLGEMEIN

TESTZEITRAUM

Oktober 2016 - Juli 2017

SW-VERSION

Der Test erfolgte mit der Softwareversion 2.2.4 von OpenSIPS. Am Ende des Testzeitraums war eine neuere stabile Version (2.3.0) verfügbar, welche aber aus Gründen der Vergleichbarkeit der Ergebnisse nicht zum Einsatz kam.

GERÄTEKLASSE

OpenSIPS steht für Open SIP Server, d.h. OpenSIPS ist eine Open Source-Implementierung eines SIP-Servers. Es ist somit ein Teil der Client-Server-Architektur des SIP-Protokolls.
OpenSIPS hat seine Webpräsenz unter http://www.opensips.org.

FUNKTIONALITÄTEN

OpenSIPS umfasst zahlreiche Funktionalitäten, z.B. SIP Registrar, SIP Router / Proxy, SIP Redirect Server u.v.m. Eine vollständige Liste ist unter http://www.opensips.org/About/Features zu finden.
OpenSIPS dient dabei als zentrale Struktur, z.B. zum Management von SIP-Endgeräten, zur Vereinfachung von Dialstrings oder auch zum Spamschutz.
Es gibt weitreichende Anpassungsmöglichkeiten durch vorgefertigte Module und eine eigene Skriptsprache.

HARDWARE

OpenSIPS läuft auf Linux-basierten Systemen. Im Test kam Debian 8 ("jessie") zum Einsatz.

INSTALLATION / KONFIGURATION

REPOSITORY VS. MANUELLE INSTALLATION

Zunächst sei angemerkt, dass der erhöhte Aufwand der Installation / Konfiguration bei Open Source immer zu beachten ist. Daher ist die Verwendung der offiziellen Repositorys zum Erhalt der Pakete sehr angeraten. Offiziell verfügbar sind Repositorys für Debian/Ubuntu, Redhat/CentOS/Fedora. Natürlich sind die Repositorys mit Schlüsseln versehen, welche vorher importiert werden sollten um die Authentizität der Pakete sicherzustellen. Ein großer Vorteil bei Nutzung von Repositorys ist die Möglichkeit im Nachgang noch einzelne Module als Pakete einfach installieren zu können, ohne alles neu kompilieren zu müssen.

Alternativ bzw. falls kein Repository genutzt werden kann stehen Tarballs bzw. Checkout über GIT/SVN zur Verfügung. Die Kompilierung ist über das mitgelieferte Tool **menuconfig** (startbar mittels **make menuconfig**) sehr gut möglich. Es können dabei Compile Flags (z.B. Support für IPv6, TCP, TLS) bzw. die zu verwendenden Module (z.B. Datenbank-Anbindung für MySQL, Presence, XMPP-Support) ausgewählt werden. Abhängigkeiten zu anderen (externen) Paketen müssen aber natürlich manuell aufgelöst werden. Allgemein notwendig sind z.B. die Entwicklungsbibliothek von **ncurses** (**libncurses5-dev**) und die Pakete **flex**, **bison** und **m4**.

GRUNDLEGENDE EINSTELLUNGEN

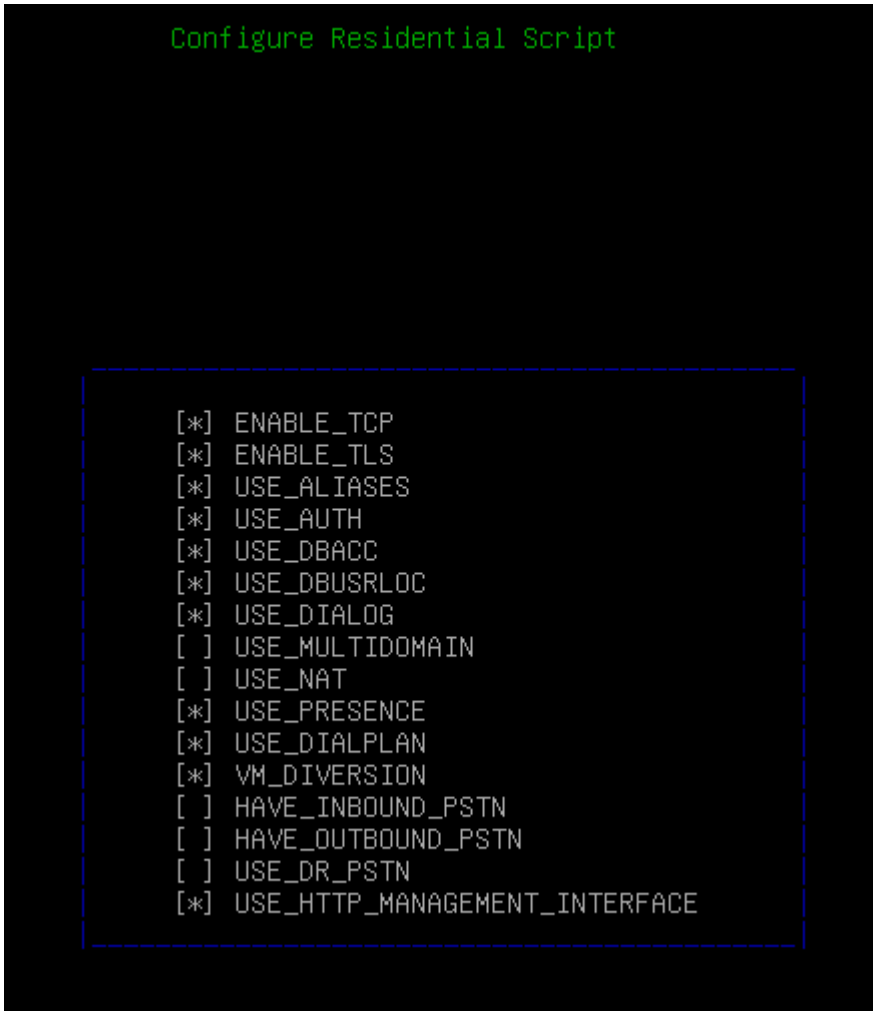
Zunächst ist sicherzustellen, dass das zugrunde liegende Linux-basierte System korrekte Einstellungen für das DNS besitzt. Daher sind die Werte in **/etc/hosts**, **/etc/hostname** und **/etc/resolv.conf** zu überprüfen und gegebenenfalls anzupassen.
Die wichtigsten Pfade zu den Konfigurationsdateien lauten (bei Installation über ein Repository):

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

In den beiden Dateien **/etc/opensips/opensipsctlrc** und **/etc/opensips/osipsconsolerc** sind eine Reihe von grundlegenden Einstellungen mit Standardwerten versehen. Wichtig ist es die gewünschte SIP-Domäne und die Datenbankanbindung mit den entsprechenden Zugangsdaten zu konfigurieren. Die Datenbank und der Datenbanknutzer müssen dabei noch nicht existieren und werden später angelegt (siehe nächster Abschnitt "Datenbank"). Im Test kam dabei als SIP-Domäne **osips.vcc.test.tu-dresden.de** und als Datenbank MySQL auf dem selben Linux-System zum Einsatz. Die weiteren Einstellungen wurden auf den angegebenen Vorgabewerten belassen.

DATENBANK

Mittels des Tools **opensipsdbctl create** werden die Datenbank, der Datenbanknutzer und die entsprechenden Berechtigungen gemäß den Vorgaben aus **/etc/opensips/opensipsctlrc** angelegt.
Zu beachten ist hierbei, dass bei Nutzung des OpenSIPS Control Panels (siehe entsprechender Abschnitt) noch weitere Tabellen in dieser Datenbank angelegt werden müssen.



KONFIGURATIONSDATEI / SKRIPTVORLAGE

Für OpenSIPS existiert eine zentrale Konfigurationsdatei unter **/etc/opensips/opensips.cfg**, die z.B. die zu verwendenden IP-Adressen, eine Auflistung der zu verwendenden Module, Einstellungen für diese Module (u.a. Datenbankanbindung, Dateipfade) und Einstellungen zum Logging enthält. Ein ebenfalls wesentlicher Bestandteil ist das Routing-Skript, dass bei jeder Anfrage (z.B. Register-Request oder Invite-Request) durchlaufen wird und sehr genau auf die benötigten Anforderungen angepasst werden kann.

Zur Erstellung einer solchen Konfigurationsdatei bietet es sich an das Tool **osipsconfig** zu verwenden. Es generiert je nach gewünschtem Einsatzzweck und einer ersten Anforderungsliste (siehe schwarze Box auf der rechten Seite) eine Konfigurations-/Skriptvorlage. Diese Vorlage muss noch an spezifischen Stellen, die mit **# CUSTOMIZE ME** markiert sind, an die lokalen Bedingungen angepasst werden. So sind z.B. IP-Adressen, Datenbankanbindung und Dateipfade betroffen:

```
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
```

Es empfiehlt sich das Modul Domain noch mit anzugeben, um lokale Domains besonders auszeichnen zu können:

```
####### 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

Bis zum Produktivbetrieb ist es sehr nützlich so viele Informationen wie möglich in den Log-Files stehen zu haben, da erfahrungsgemäß einige Fehler und Probleme nicht von vornherein gesehen und umgangen werden können. Daher ist folgender Eintrag in der Datei **/etc/opensips/opensips.cfg** sinnvoll:

```
####### Global Parameters #########

log_level=4
log_stderror=no
log_facility=LOG_LOCAL1
```

Außerdem muss im zugrunde liegenden Linux-basiertem System noch angegeben werden, was mit local1 genau gemeint ist. Daher ist ein Eintrag in **/etc/rsyslog.conf** nötig, um die Ausgabe in eine Datei umzuleiten:

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

Ein Neustart von **rsyslog** im Anschluss ist ebenfalls notwendig.

START ALS SERVICE VS. MANUELLER START

Für einen ersten Test ist ein manueller Start noch praktikabel, allerdings sollte bei häufigerer Verwendung von OpenSIPS der Start als Service präferiert werden.
Der manuelle Start erfolgt mittels **opensips -f /etc/opensips/opensips.cfg**.

Für den Start von OpenSIPS als Service müssen 2 Dateien angepasst werden:

- **/etc/defaults/opensips**: Anpassungen evtl. bei Nutzer (root) und Ausführung (yes) nötig
- **/etc/init.d/opensips**: Anpassungen bei den Dateipfaden evtl. nötig

Die Nutzung erfolgt mittels **/etc/init.d/opensips {start|stop|restart|force-reload|status}**. Insbesondere mittels **/etc/init.d/opensips status** ist evtl. sogar eine erste Problemanalyse im Bedarfsfall möglich.

DNS-EINTRÄGE

Damit OpenSIPS von den SIP-Endgeräten über die SIP-Domäne gefunden werden kann sind DNS-Einträge notwendig. Dabei sind direkte A-Records und auch SRV-Records sinnvoll:

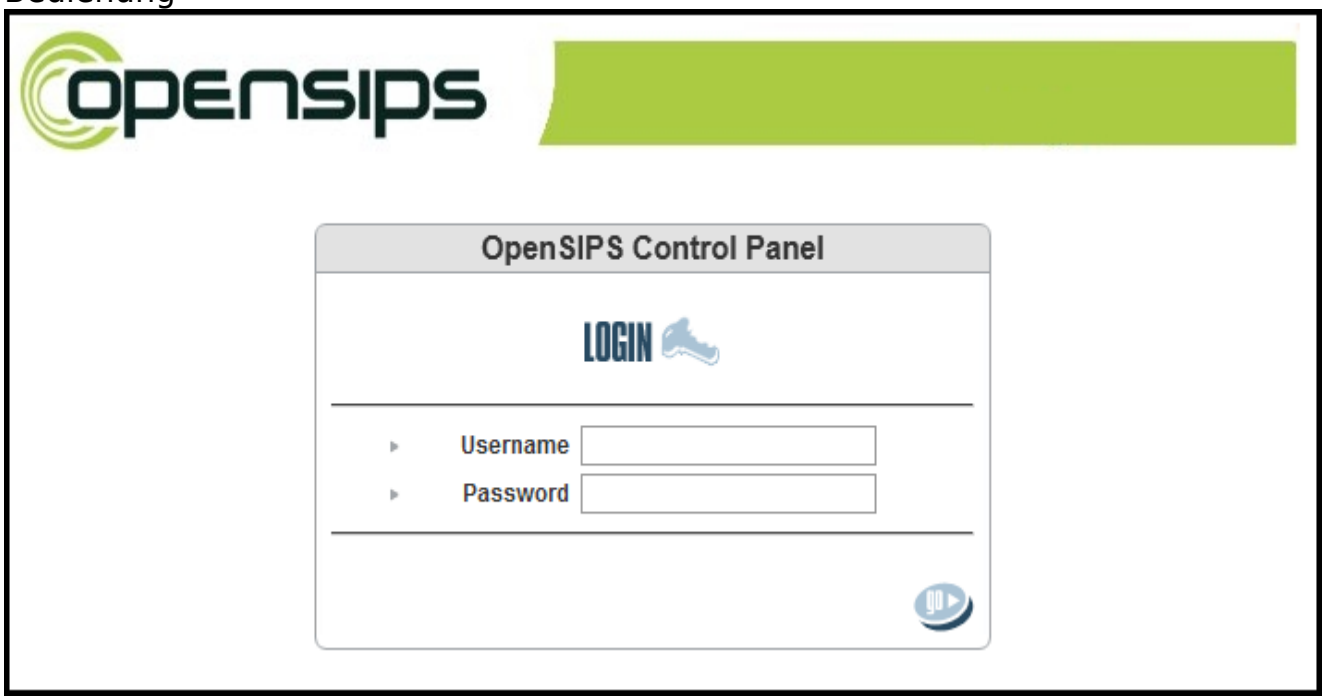
- A-Record: SIP-Domäne => IP-Adresse von OpenSIPS
- SRV-Records: {_sip|_sips}.{_tcp|_udp|_tls}.SIP-Domäne=> DNS-Name von OpenSIPS, Port {5060|5061}

Bei den angegebenen SRV-Records sind natürlich nicht alle Kombinationen sinnvoll. Die Auswahlmöglichkeiten rühren nur von der möglichen Verschlüsselung und des Transportprotokolls her. Sinnvolle Varianten für _tcp sind z.B. nur:

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

Es ist empfehlenswert auch an die entsprechenden Reverse-DNS-Einträge zu denken und die DNS-Einträge für die SIP-Endgeräte gleich mit zu überprüfen.

Bedienung



OPENSIPS CONTROL PANEL

Bei OpenSIPS Control Panel handelt es sich um eine grafische Benutzeroberfläche für OpenSIPS, welche php-basiert ist. OpenSIPS Control Panel wird separat entwickelt und ist unter http://controlpanel.opensips.org/ als Tarball oder SVN Checkout erhältlich.

Einige Paketabhängigkeiten bestehen, so müssen **libapache2-mod-php5, php5, php5-cli, php5-gd, php5-mysql, php-pear, php5-xmlrpc** installiert werden. Außerdem benötigt OpenSIPS Control Panel noch einige weitere Module, die mittels **pear** installiert werden: **pear install MDB2, pear install MDB2#mysql, pear install log**

Die Einbindung in Apache2 erfolgt z.B. mittels Alias- und Directory-Statements in **/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

Mit dieser Einrichtung ist OpenSIPS Control Panel unter host/opensips-cp zu erreichen, wobei host der DNS-Name des Rechners ist, auf dem OpenSIPS Control Panel eingerichtet ist. Im Test war dies der selbe Rechner auf dem auch OpenSIPS läuft.
Dabei ist zu beachten, dass die Dateien von OpenSIPS Control Panel ausreichende Berechtigungen erhalten haben (z.B. www-data) und Apache2 im Anschluss neu geladen werden muss.

Ein weiterer wichtiger Punkt ist die Notwendigkeit der Einstellung

short_open_tag=On

in der Datei **php.ini**, damit die php-Skripte funktionieren.

Außerdem sind diverse Erweiterungen der vorhandenen Datenbank auf dem Rechner mit OpenSIPS nötig. So existieren vorgefertigte Tabellen **ocp_admin_privileges.mysql**, **cdrs.mysql** und **tables.mysql**, die in die Datenbank von OpenSIPS eingefügt werden müssen. Der erste administrative Nutzer von OpenSIPS Control Panel wird in die Tabelle **ocp_admin_privileges** mittels

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

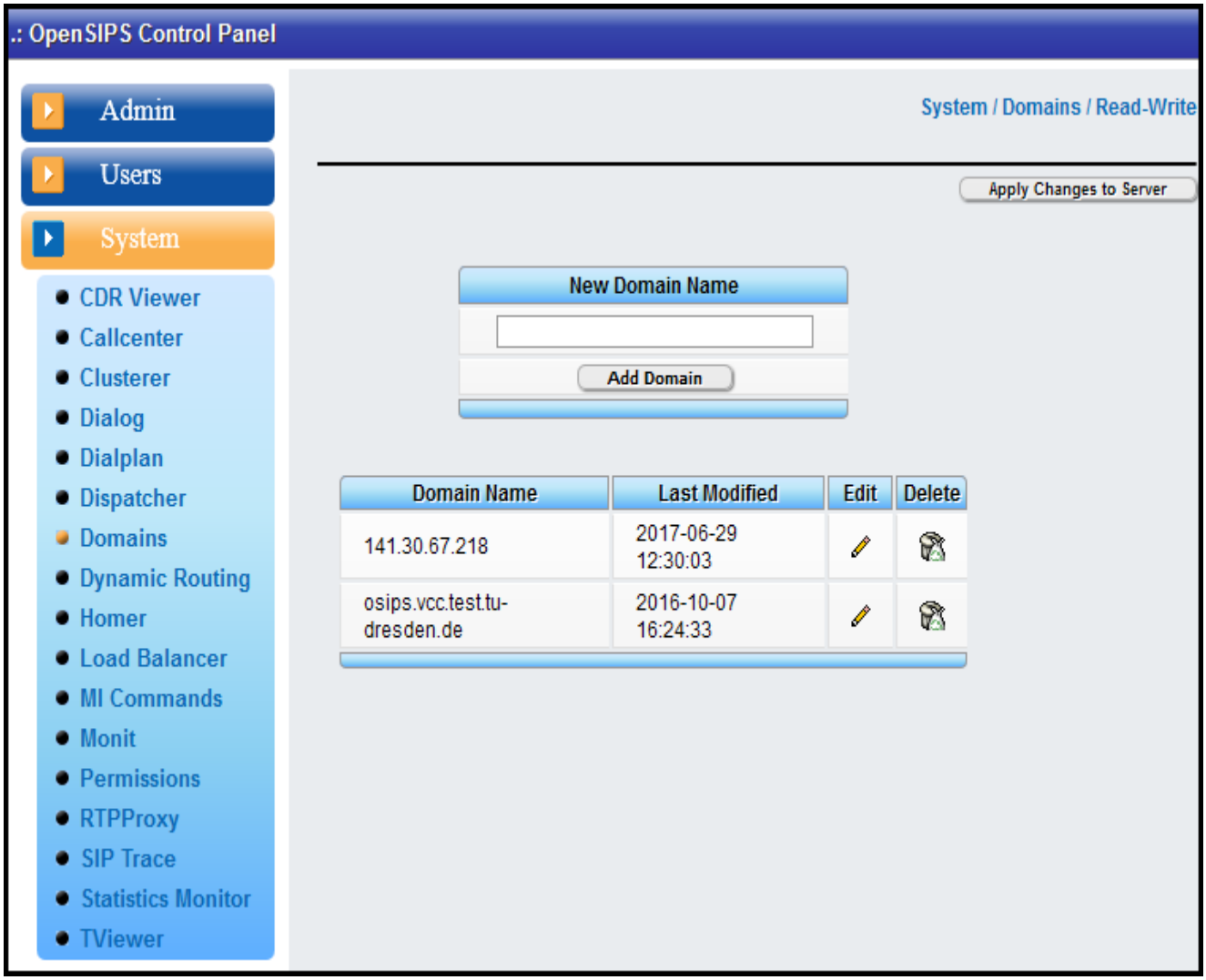
eingetragen. Neben der Anpassung der Datenbank an sich muss beim OpenSIPS Control Panel in **config/db.inc.php** die Datenbankanbindung und in **config/boxes.global.inc.php** die OpenSIPS fifo-Queue angegeben werden. Die OpenSIPS fifo-Queue wird dabei in **/etc/opensips/opensips.cfg** unter FIFO Management Interface definiert.

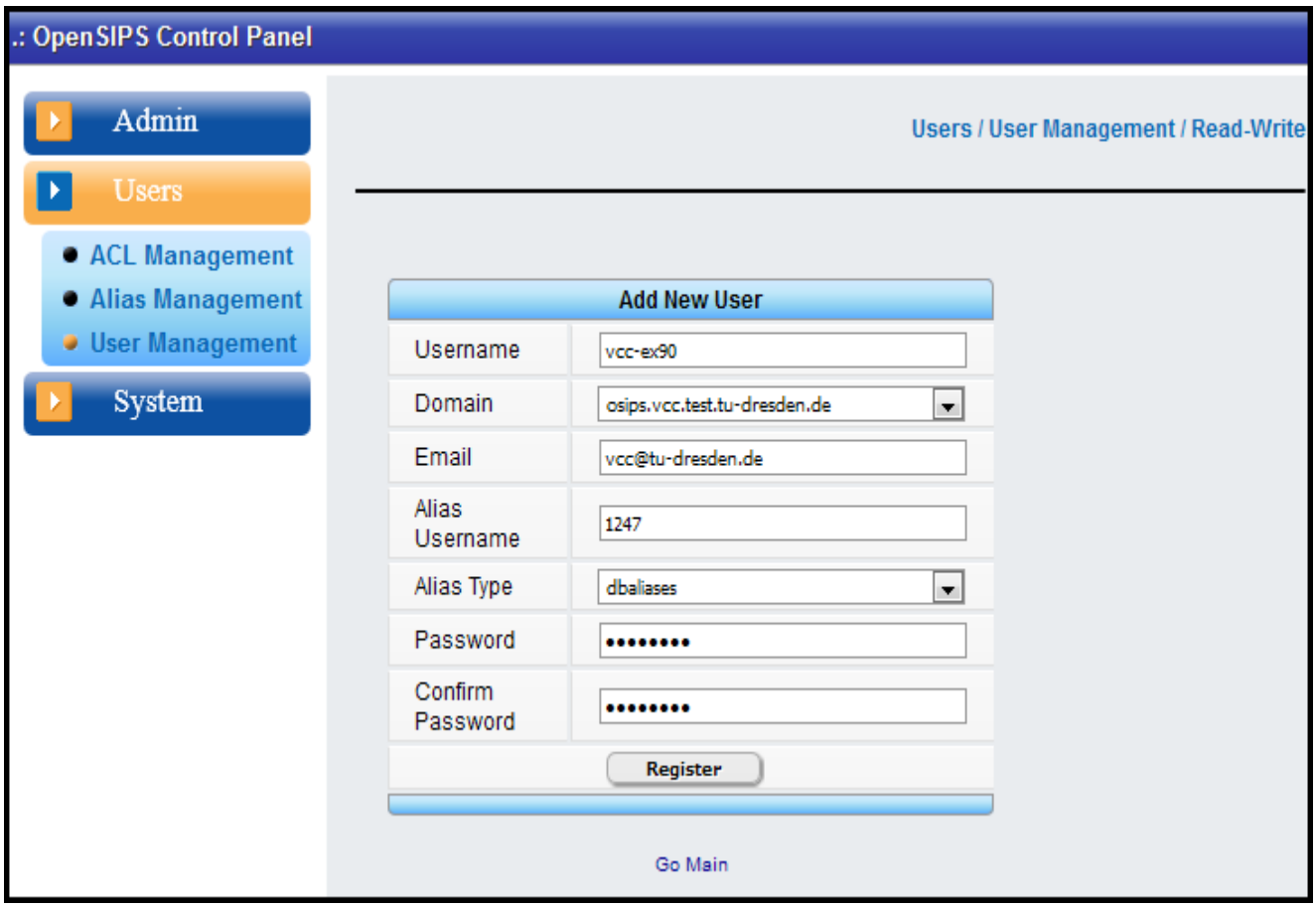
Als letzter Einrichtungspunkt ist noch ein Eintrag in **/etc/crontab** für die Statistiken sinnvoll:

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

Das OpenSIPS Control Panel kann vielfältig genutzt werden. So ist u.a. der Eintrag von SIP-Domänen, das Anlegen von Zugängen / Accounts, die Anzeige aktiver und vergangener Calls und ein Direktzugriff auf verschiedene Module / Tabellen der Datenbank möglich. Als SIP-Domäne sollte immer auch die zugehörige IP-Adresse mit angegeben werden (siehe Screenshot weiter unten).
Dabei gilt zu beachten, dass nicht alle Daten der Datenbank / Module im OpenSIPS Control Panel abgebildet sind, es können aber eigene Menüpunkte programmiert werden.

Ein Debugging ist ohne direkten Datenbankzugang und Zugriff auf die Log-Files von OpenSIPS schwer möglich.


.: OpenSIPS Control Panel
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
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



EINSTELLUNGEN AM SIP-ENDGERÄT

Die notwendigen Einstellungen am SIP-Endgerät halten sich erwartungsgemäß sehr in Grenzen. Zu beachten ist aber, dass bei Geräten die auch andere Protokolle (z.B. H.323) beherrschen zunächst SIP als Protokoll zu aktivieren ist. Auch das "Lauschen" auf Port 5060 bzw. 5061 für eingehende Rufe kann schnell übersehen werden.

Zu den notwendigen Einstellungen am SIP-Endgerät zählen:

- Auswahl des Transportprotokolls (TCP, UDP, TLS)
- Angabe der URI (name@SIP-Domäne), z.B. vcc-ex90@osips.vcc.test.tu-dresden.de
- Angabe Login mit den erzeugten Credentials
- Angabe des SIP-Proxys (DNS-Name von OpenSIPS)

KONTROLLMÖGLICHKEITEN / FEHLERSUCHE

Es gibt zahlreiche Möglichkeiten und Stellen mit denen Fehler / Probleme im Zusammenhang mit OpenSIPS gefunden werden können:

- Am SIP-Endgerät: Je nach Gerät werden z.B. in der Weboberfläche Meldungen zum Registrierungsstatus angezeigt.
- Mittels **netstat –tulpn | grep opensips**: Prüfung, ob OpenSIPS auf den richtigen Ports auf eingehende SIP-Requests wartet.

- In den Logs: In **/var/logs/opensips.log** können bei hohem Log-Level sehr viele Vorgänge nachvollzogen werden. Es stehen jeweils auch die verwendeten Module / Funktionen dabei:

- Im OpenSIPS Control Panel: Es können z.B. laufende Calls im Unterpunkt Dialog und abgeschlossene Calls im Unterpunkt CDR Viewer eingesehen werden:

- In der Datenbank: In der Datenbank von OpenSIPS sind z.B. fehlgeschlagene Rufe in der Tabelle missedCalls und angemeldete SIP-Endgeräte in der Tabelle location zu finden.
- Mittels Kommandozeilen-Tools: Serverstatistiken sind mittels **opensipsctl monitor** und angemeldete SIP-Endgeräte mittels **opensipsctl ul show** bzw. **opensipsctl online** einsehbar:

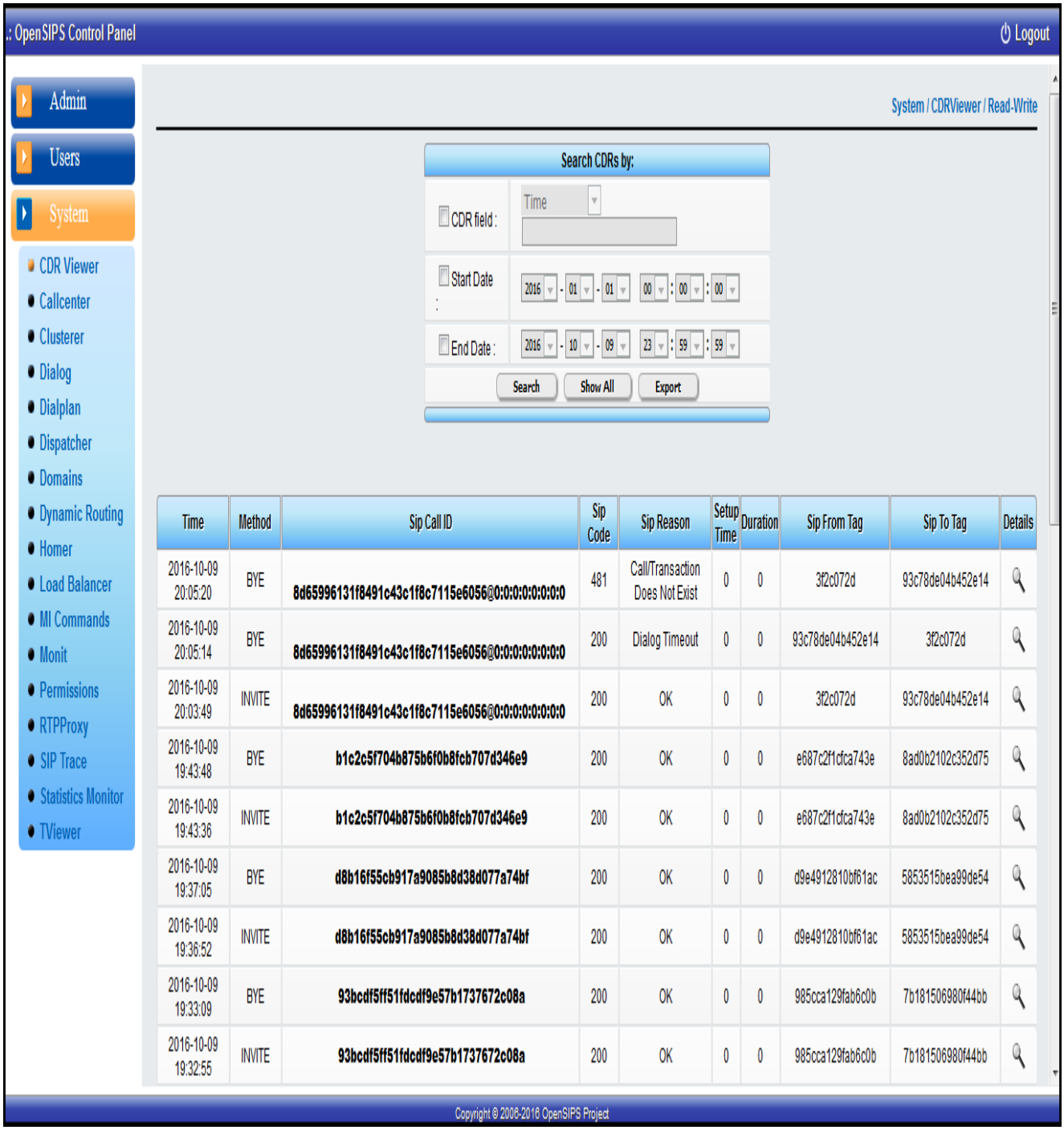
```
root@rncmm0-218:~# opensipsctl ul show
Domain:: location table=512 records=1
        AOR:: vcc-ex90
                Contact:: sip:vcc-ex90@141.30.67.247:5060 Q=
                        Expires:: 3370
                        Callid:: d02faa1d1a33dbff23d0e3671e24d03e
                        Cseq:: 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:~# _
```

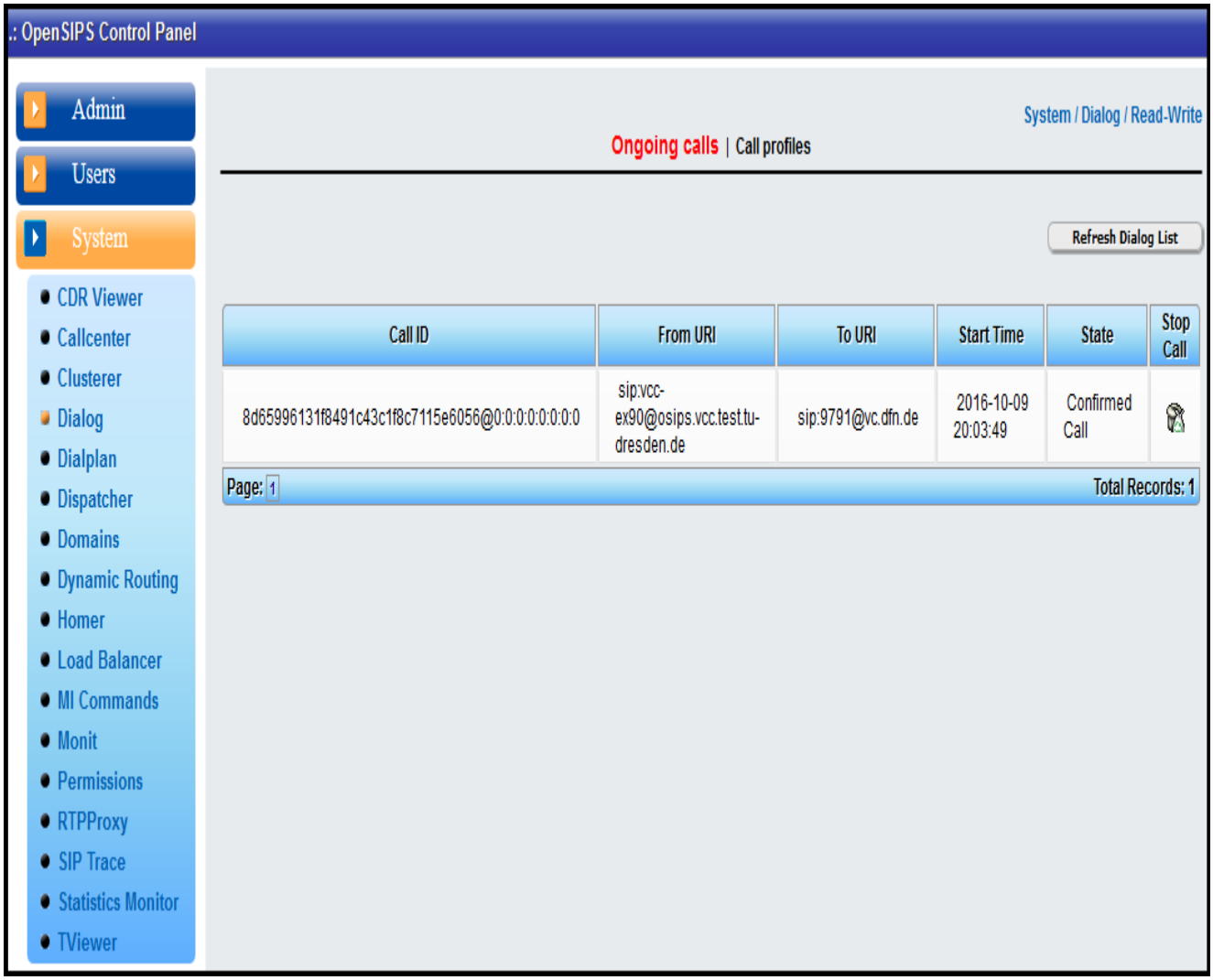
```
[cycle #: 10; if constant make sure server lives]
Server:: OpenSIPS (2.2.1 (x86_64/linux))
Now:: Mon Dec 12 16:17:44 2016
Up since:: Mon Dec 12 16:09:46 2016
Up time:: 478 [sec]

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
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
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 | 8d65996131f8491c43c1f8c7115e6056@0:0:0:0:0:0:0:0 | 481 | Call/Transaction Does Not Exist | 0 | 0 | 3f2c072d | 93c78de04b452e14
2016-10-09 20:05:14 | BYE | 8d65996131f8491c43c1f8c7115e6056@0:0:0:0:0:0:0:0 | 200 | Dialog Timeout | 0 | 0 | 93c78de04b452e14 | 3f2c072d
2016-10-09 20:03:49 | INVITE | 8d65996131f8491c43c1f8c7115e6056@0:0:0:0:0:0:0:0 | 200 | OK | 0 | 0 | 3f2c072d | 93c78de04b452e14
2016-10-09 19:43:48 | BYE | b1c2c5f704b875b6f0b8fcb707d346e9 | 200 | OK | 0 | 0 | e687c2f1cfca743e | 8ad0b2102c352d75
2016-10-09 19:43:36 | INVITE | b1c2c5f704b875b6f0b8fcb707d346e9 | 200 | OK | 0 | 0 | e687c2f1cfca743e | 8ad0b2102c352d75
2016-10-09 19:37:05 | BYE | d8b16f55cb917a9085b8d38d077a74bf | 200 | OK | 0 | 0 | d9e4912810bf61ac | 5853515bea99de54
2016-10-09 19:36:52 | INVITE | d8b16f55cb917a9085b8d38d077a74bf | 200 | OK | 0 | 0 | d9e4912810bf61ac | 5853515bea99de54
2016-10-09 19:33:09 | BYE | 93bcdf5ff51fdcdf9e57b1737672c08a | 200 | OK | 0 | 0 | 985cca129fab6c0b | 7b181506980f44bb
2016-10-09 19:32:55 | INVITE | 93bcdf5ff51fdcdf9e57b1737672c08a | 200 | OK | 0 | 0 | 985cca129fab6c0b | 7b181506980f44bb
Copyright © 2008-2016 OpenSIPS Project




```
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
```

RUFMÖGLICHKEITEN

Prinzipiell sind die Rufe nach dem Registrationsstatus der beteiligten SIP-Endgeräte und nach dem jeweils verwendeten Transportprotokoll zu unterscheiden.

Auf Sender- und Empfangsseite des Rufes kann dabei jeweils TCP / UDP oder TLS zum Einsatz kommen, wenn OpenSIPS entsprechend konfiguriert ist. Daher sind 9 Varianten denkbar, wobei immer OpenSIPS den evtl. notwendigen Protokollwechsel in der Signalisierung übernimmt.

Bzgl. des Registrationsstatus sind 3 Varianten denkbar:

- "intern" zu "intern": Hier sind nur am OpenSIPS registrierte SIP-Endgeräte beteiligt. Rufe sind dann mit Kurzwahl des Username bzw. Alias möglich (SIP-Endgeräte fügen dann die eigene SIP-Domäne hinzu). Natürlich sind auch Rufe mittels **name/alias@SIP-Domäne** möglich.
- "intern" zu "extern": Rufe zu entfernten Gegenstellen erfolgen mittels **uri@other-sip-domain**.

- "extern" zu "intern": Entfernte Gegenstellen haben keine Kurzwahlmöglichkeit und müssen immer **name/alias@SIP-Domäne** rufen.

NUTZUNG VON TLS / ZERTIFIKATSMANAGEMENT

Um verschlüsselte Signalisierung mittels TLS nutzen zu können, müssen bei OpenSIPS die entsprechenden Module konfiguriert und mindestens ein Zertifikat erstellt werden.

Die Zertifikate werden am besten über die DFN-PKI beantragt. Dazu ist es notwendig jeweils einen Zertifikatsrequest mittels **openssl** zu erstellen. Die Anforderungen an die Zertifikate sollten dabei beachtet werden:

Zertifikat für OpenSIPS:

- Zertifikatsprofil: VoIP Server
- eindeutiger Name: DNS-Name OpenSIPS
- alternative Namen: DNS-Name OpenSIPS und SIP-Domäne

Zertifikate für SIP-Endgeräte:

- Zertifikate sind nicht unbedingt notwendig bzw. können auch die meist vorhandenen Build-In-"Standard-Zertifikate" verwendet werden.
- Zertifikate haben die analogen Anforderungen wie bei OpenSIPS, nur dass natürlich jeweils der DNS-Name des SIP-Endgeräts Verwendung findet.

Wichtig ist, dass die DFN-PKI-Stammzertifikate bei OpenSIPS und den SIP-Endgeräten als vertrauenswürdige CA hinterlegt werden, damit auch eine Validierung erfolgen kann.

Die Einbindung der Zertifikate erfolgt in **/etc/opensips/opensips.cfg**. Evtl. vorhandene Zeilen mit modparam("proto_tls", ...) müssen auskommentiert werden, da sich inzwischen alles im Modul tls_mgm befindet:

```
####### 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/")
```

Die Datei mit dem privaten Key sollte ohne Passphrase verfügbar sein, ansonsten ist ein Start als Service nicht möglich, da dort das Passwort nicht eingeben werden kann.
Falls nicht für alle SIP-Endgeräte ein Zertifikat vorhanden ist (require_cert) oder überprüfbar ist (verify_cert), muss der entsprechende Schalter auf "0" gesetzt werden. Die Überprüfung des SIP-Endgeräts ist dann nicht anhand des Zertifikats möglich, aber der Verbindungsaufbau ist verschlüsselt.

Da die DFN-PKI-Zertifikate etwas größer sind, benötigten die Endgeräte im Test mehr als 100ms beim TLS-Handshake mit OpenSIPS. Dabei kam es zu ungewollten Verbindungsabbrüchen beim Rufaufbau. Daher sollten folgende Parameter verändert werden:

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

TLS-Verbindungen werden unterschieden zwischen eingehenden TLS-Verbindungen zu OpenSIPS (server_domain) und ausgehenden TLS-Verbindungen von OpenSIPS (client_domain). Je nach Socket (IP-Adresse:Port) können dabei unterschiedliche Einstellungen vorgenommen werden:

```
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/")
```

Diese Unterscheidung ist insbesondere dann nützlich, wenn OpenSIPS mehrere SIP-Domänen verwalten soll und über verschiedene IP-Adressen kontaktiert werden kann. Es wird dann sichergestellt, dass das verwendete Zertifikat auch zu der SIP-Domäne passt, die gerade benutzt wird.

NUTZUNG WEITERER MODULE / SKRIPTING

Dieser Abschnitt beschreibt beispielhaft die Vorgehensweise wie OpenSIPS weiter an die eigenen Bedarfe angepasst werden kann. Hilfreich für das Skripting sind die Dokumentationen der verfügbaren Variablen, Funktionen und 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-MODUL

Das Modul Domain wurde bereits im Abschnitt Konfigurationsdatei / Skriptvorlage in **/etc/opensips/opensips.cfg** eingefügt:

```
####### 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")
```

Es ermöglicht die Angabe von SIP-Domänen in der Datenbank bzw. über OpenSIPS Control Panel. Die angegebenen SIP-Domänen werden als "lokal" erkannt. Es erfolgt bei jedem Ruf eine Überprüfung, dass mindestens eine Seite (Rufer, Gerufener) lokal ist.

DIALPLAN-MODUL

Das Modul Dialplan ermöglicht die Umschreibung von Strings, typischerweise der Request-URI. Es beherrscht String matching (Matching Operator = 0) und Regex matching (Matching Operator = 1). Die Angabe der Regeln erfolgt über die Datenbank bzw. über OpenSIPS Control Panel:

Dialplan ID	Rule Priority	Matching Operator	Matching Regular Expression	Matching Flags	Substitution Regular Expression	Replacement Expression	Atrributes	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

Die Nutzung im Routing-Skript von **/etc/opensips/opensips.cfg** erfolgt mittels **dp_translate ("0","$ru/$ru");**
Dabei steht $ru für die Request-URI und der erste Wert gibt an, das alle Regeln mit Dialplan ID = 0 überprüft werden sollen.
Ein Beispiel für die Anwendung der obersten Regel ist hier zu sehen (Auszug aus dem Log-File):

```
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
```

SCHUTZ DER SIP-ENDGERÄTE / MAßNAHMEN ZUM SPAMSCHUTZ

Es ist allgemein bekannt, dass ungeschützte SIP-Endgeräte gern SIP-Spamanrufen zum Opfer fallen, die auch stärker als bei H.323 auftreten. Ziel der Angreifer ist meistens die kostenfreie Nutzung von Telefonie oder die Erzeugung hoher Kosten über die infiltrierte VoIP-Plattform. Bei Videokonferenzsystemen ist der Kostenpunkt meist nicht gegeben (außer es ist eine Verbindung in die ISDN-Welt gegeben) aber die SIP-Spamanrufe stören bzw. blockieren die normale Kommunikation erheblich. Zum Schutz kann mit OpenSIPS ein einfacher Spamschutz etabliert werden.

Zunächst sollten die SIP-Endgeräte nach der Registrierung an OpenSIPS nicht mehr von "außen" über Port 5060 / 5061 direkt erreichbar sein, da sonst OpenSIPS bei der Signalisierung umgangen werden kann. Bei einigen Geräten kann dies direkt eingestellt werden, ansonsten muss es mithilfe von Firewalls, ACLs etc. umgesetzt werden. Somit ist die Signalisierung über OpenSIPS erzwungen, die Medienströme (RTP / SRTP) werden aber weiterhin direkt über dynamisch ausgehandelte Ports zwischen den SIP-Endgeräten etabliert.

Der überwiegende Anteil der SIP-Spamanrufe hat die Form numbers@ip-address als To-URI. Diese kommen weiterhin bei OpenSIPS an und könnten auch an entsprechende SIP-Endgeräte durchgestellt werden. Daher ist z.B. ein Block der SIP-Spamanrufe anhand der IP-Adresse als SIP-Domäne möglich. Die notwendigen Zeilen im Routing-Skript von **/etc/opensips/opensips.cfg** lauten:

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

Dabei ist $td die SIP-Domäne in der To-URI ($tu). Falls OpenSIPS über mehrere IP-Adressen erreichbar ist, muss der Ausdruck entsprechend angepasst werden. Natürlich können weitere Aspekte der SIP-Spamcalls berücksichtigt werden, der einfache Spamschutz zeigte im Test aber bereits ausreichende Wirkung:

```
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
```

Reguläre Rufe müssen bei diesem Szenario durch **name/alias@SIP-Domäne** erfolgen, wobei SIP-Domäne ein DNS-Name ist.

FAZIT

Die Installation / Konfiguration von OpenSIPS ist leider nicht selbsterklärend, dafür ist die Dokumentation gut nutzbar. Der Weg bis zur Einsatzfähigkeit von OpenSIPS gestaltet sich trotzdem recht zeitintensiv, insbesondere da die Methode "Trial & Error" öfters verwendet werden muss.
Das separate OpenSIPS Control Panel erleichtert typische Tätigkeiten, ist aber kein vollständiger Ersatz für Datenbankzugriffe oder Konsolenarbeit.
Zahlreiche Stellen an denen Fehler gesucht werden können erzwingen einen gewissen Überblick und daher auch eine entsprechende Einarbeitungszeit.
Das Zertifikatsmanagement lässt sich gut nutzen, insbesondere da Zertifikate über die DFN-PKI einfach, schnell und zuverlässig bezogen werden können.
Eine Einarbeitung in die Skriptsprache, Konsolenarbeit etc. ist absolut unerlässlich.
Zahlreiche Module und die Möglichkeit von Eigenentwicklungen bieten einen großen Spielraum. Z.B. ist ein einfacher Spamschutz für SIP-Endgeräte umsetzbar.
