



Senior Voice Engineer for QSC AG, one of the major German voice and data providers. Alexandr holds a diploma in physics of Odessa State University. He has 20 years of experience in telecommunication techniques and has contributed to many OpenSource projects like FeeSwitch, SER, Kamailio, SEMS, Asterisk, SIPP, Wireshark. Alexandr is the main developer of Homer SIP Capture project. He is also founder of IRC RusNet Network, one of the biggest national IRC networks in the world.

QSC AG is digitising the German SME sector. With decades of experience and expertise in the areas of Cloud, Consulting, Outsourcing, and Telecommunications, QSC accompanies its customers securely into the digital age. Today already, cloud-based procurement models offer increased speed, flexibility, and full service availability. The Company's TÜV and ISO-certified data centres in Germany and its nationwide All-IP network form the basis for maximum end-to-end quality and security. QSC's customers benefit from one-stop innovative products and services that are marketed both directly and via partners. For details visit: <http://qsc.de>

QXIP BV {QuickSIP} is an Amsterdam based R&D Company specializing in Open-Source and Commercial Voice Technology Development - Our flagship projects are SIPCAPTURE **HOMER** and **PCAPTURE** based on our mature and open encapsulation protocol **HEP/EEP** (*Extensible Encapsulation Protocol*) Our Open-Source solutions are deployed and trusted by thousands of businesses worldwide. Our Customers include large telephony and network operators, voice service carriers, voip service providers, cloud service providers, call center operators and voice equipment vendors. Our Capture Technologies are natively implemented in all major OSS voip platforms such as *Kamailio*, *OpenSIPS*, *FreeSWITCH*, *Asterisk*, *RTPEngine* and many tools such as *sipgrep*, *sngrep* and more. For full details please visit our website at <http://qxip.net>

Things you already know about #HOMER

If you missed our Workshop, make sure you grab the PDF!

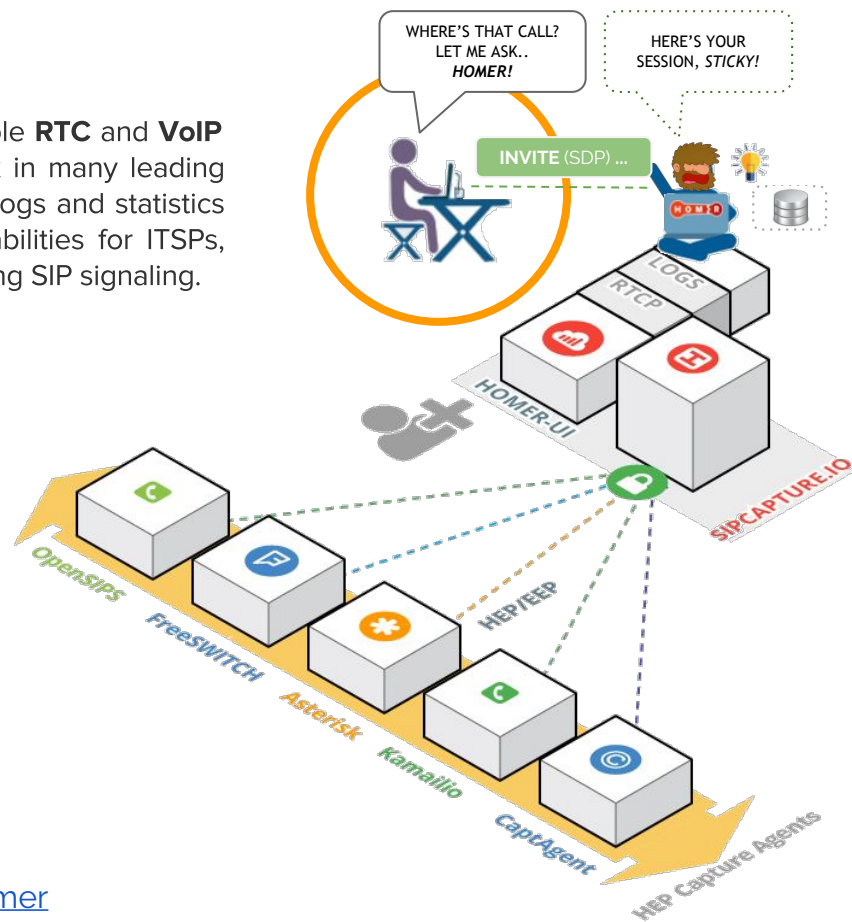
HOMER is part of the **SIPCAPTURE** stack, a robust, carrier-grade, scalable **RTC** and **VoIP** Capture and Monitoring application with built in support out of the box in many leading platforms, ready to process, index & store insane amounts of signaling, logs and statistics and providing instant search, end-to-end analysis and drill-down capabilities for ITSPs, VoIP Providers Trunk Suppliers as well as Enterprises and Developers using SIP signaling.

HOMER provides many features and advantages, including:

- Instant centralized access to present and past signaling & stats
- Full SIP/SDP payload retention with precise timestamping
- Automatic correlation of sessions, logs and reports
- Support for RTP and RTCP Media statistics and analytics
- Visual representation of multi protocol session call-flows
- Fast detection of usage and system anomalies
- System agnostic view of VoIP and RTC traffic flows
- Unlimited plug & play capture agents and HEP custom data feeds
- Multi-User and Customizable UI based on JS/Angular/D3
- PCAP Exporting and Sharing functionality with 3rd parties

... and much more!

FIND ALL ABOUT HOMER: <http://github.com/sipcapture/homer>



What is HEP/EEP ?

HEP = Homer Encapsulation Protocol

EEP = Extensible Encapsulation Protocol

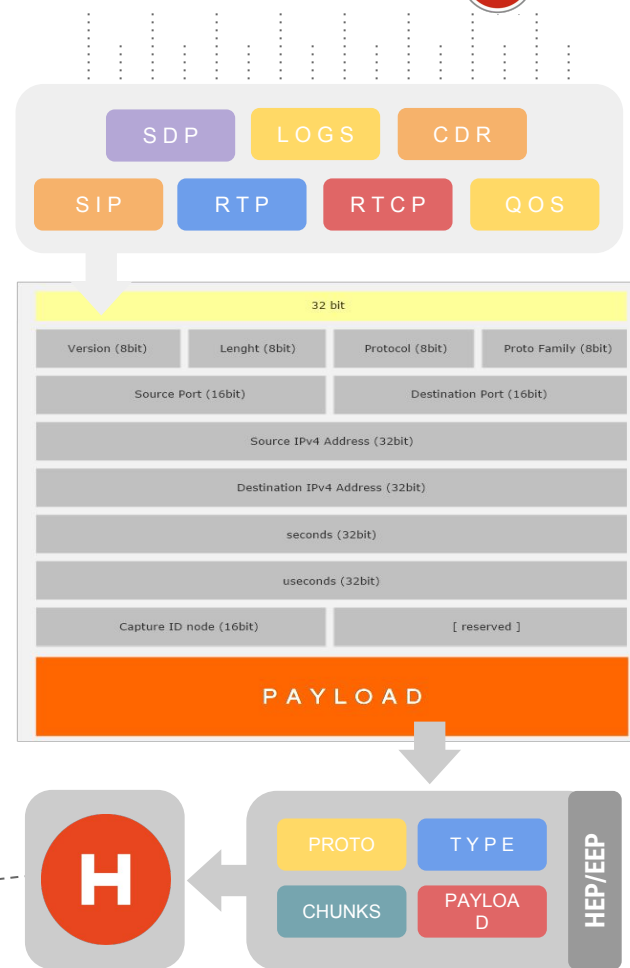
HOMER's own Encapsulation protocol (*HEP/EEP*) is used to wrap, argument and transfer captured packets between a capture *Agent* and capture *Server*.

The **HEP/EEP** Extensible Encapsulation protocol was designed to provide an efficient, modular and low-level framework to accurately duplicate passively obtained IP datagrams for remote collection over *UDP/TCP/SCTP* connections, where full retention of original datagram headers, timestamp, correlation pointers and original payload MUST be provided back to the collectors without any alterations or data loss.

The **HEP3/EEP** definition includes both generic (*internal*) and vendor- specific custom defined **chunk types** providing ground for implementors to extend the spectrum of the deliverable data within the HEP protocol alongside the encapsulated IP datagrams with custom data without protocol modifications.

HOMER currently supports HEP encoding/decoding for *SIP, XMPP, RTCP, RTCP-XR, HSP-MOS* and *Custom Logs* or *CDRs* in plain text or JSON format.

Find the full HEP/EEP specs at: <http://github.com/sipcapture/hep>



HEP/EEP - Native Capture Agents

Integrated Agents in OSS Platforms

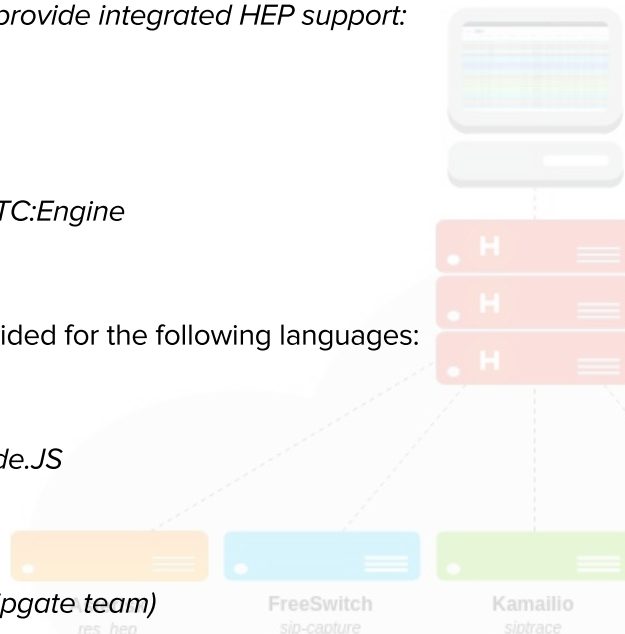
HEP agents have been consistently integrated across leading OSS solutions - chances are you have one in your fleet *already!*

The following projects provide integrated HEP support:

- Kamailio
- OpenSIPS
- FreeSWITCH
- Asterisk / PJSIP
- RTP:Engine + RTC:Engine
- sipgrep
- sngrep

Examples are also provided for the following languages:

- C/C++
- Java
- Javascript / Node.JS
- Erlang
- Python
- Go
- Perl (thanks @sipgate team)



Kamailio Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-Kamailio>

OpenSIPS Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-OpenSIPS>

FreeSWITCH Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-FreeSwitch>

CaptAgent Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-Captagent4>

nProbe VoIP Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-nProbe>

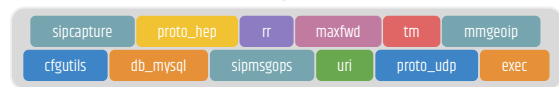
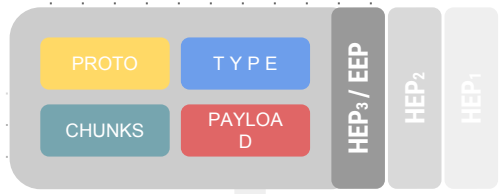
ACME SBC Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-ACME-Packet>

Find more on our wiki: <http://github.com/sipcapture/homer/wiki>

Inside the CAPTURE SERVER

Nuts and Bolts behind the HEP Sockets



sipcapture.opensips.cfg



HOMER 5 capture servers can be based on either **Kamailio 4.4+** or **OpenSIPS 2.2+** using the **SIPCAPTURE** module supporting **HEP / EEP** functionality in combination with any other available module to provide a programmable and modular **RTC** packet capture framework with no limitations and no presets, ready to extend and customize

Who's best? Only YOU decide!



sipcapture.kamailio.cfg



What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

Recent Developments:

- ★ HEP3 support in FreeSWITCH 1.6.8
- ★ HEP3 support in Kamailio (siptrace)
- ★ HEP3 support in RTPEngine mr4.4.1
- ★ CaptAgent 6.1 with new modules
- ★ New protocols support in HOMER 5

What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

During our previous **workshop** a few questions and ideas from the audience were shared about **HOMER**:

- ★ HEP/EEP Event Socket
- ★ Elasticsearch support
- ★ Postgres SQL support

Now, clearly neither one of those features and/or integrations would be possible to achieve in a single day.

This would require a modular and extensible design upfront, an incredibly flexible set of tools and skills to perform everything on the road - not to mention much more time available and a comfortable desk.

What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

During our previous **workshop** a few questions and ideas from the audience were shared about **HOMER**:

- ★ HEP/EEP Event Socket
- ★ Elasticsearch support
- ★ Postgres SQL support

Now, clearly neither one of those features and/or integrations would be possible to achieve in a single day.

This would require a modular and extensible design upfront, an incredibly flexible set of tools and skills to perform everything on the road - not to mention much more time available and a comfortable desk.

Unless you are like me.....

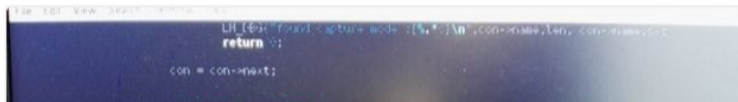


Alexandr Dubovikov
@adubovikov



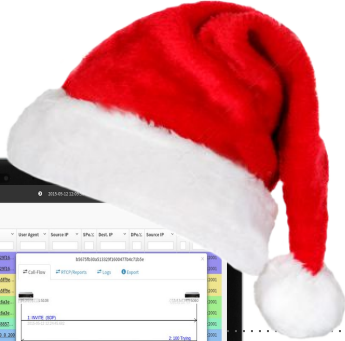
Following

Compiling kamailio and the new **@sipcapture** module at 300 km/h. On the way to **@KamailioWorld** with DB ICE



Mele Kalikimaka with HOMER 5

AKA how quickly things can happen when love & magic are involved



ID	Date	Method	Account	From User	From SIP	User Agent	DNIS	DNIS	DNIS
3447	2015-11-01 12:04:45.462	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3448	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3449	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3450	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3451	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3452	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3453	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3454	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3455	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3456	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3457	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3458	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3459	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3460	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3461	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3462	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3463	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3464	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3465	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3466	2015-11-01 12:04:45.463	INVITE	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000



Mele Kalikimaka with **HOMER 5**

AKA how quickly things can happen when love & magic are involved



WARNING: Beer consumption might have affected the rest of the presentation @2AM

What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap



History of HEP/EEP Support in Kamailio:

As the HEP encapsulation protocol evolved, so did the features of sipcapture module and Kamailio:

- Generation 1: No handling, all HEP messages directly sent to default DB table
Kamailio 3.x
- Generation 2: No handling, HEP messages can be sent to arbitrary DB table from config
Kamailio 4.x
- Generation 3: Full handling, messages can be manipulated and sent according to any parameter
Kamailio 5.x

What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap



Kamailio 5.x: HEP/EEP Event Socket

What could make HEP/EEP even more useful? An Event Socket of course! Some awesome things you can do:

- Parse, Process and Extract HEP/EEP header values and types
- Handle and Process non-SIP Messages (no parsing) directly and dynamically from the config script
- Generate custom statistics and Metrics based on the Capture Agent ID, Source, etc
- Forge and handle your very own HEP/EEP type in seconds for development and testing

Some of our common Types:

0x01	SIP
0x02	XMPP
0x03	SDP
0x05	RTCP (json serialized)
0x06	MGCP
0x29	WSS (webRTC)
0x64	LOGS (text or JSON)

KAMAILIO EVENT SOCKET

Example Usage of the Integrated Capture Agent (sipcapture module)



```

event_route[sipcapture:request] {

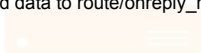
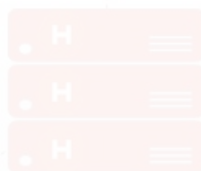
    xlog("received sipcapture request from $si:$sp\n");
    xlog("non-sip packet received - content [[${mb}] from [${si}:${sp}]\n");
    /* get proto type id from HEP header */
    hep_get("0x0B", "0x0000", "$var(data)");
    hep_get("0x0F", "0x0000", "$var(payload)");
    hep_get("0x11", "0x0000", "$var(correlation_id)");
    hep_get("0x0C", "0x0000", "$var(capture_id)");

    /* Statistics for capture id */
    if($sht(a=>captureagent::$var(capture_id)) == $null) $sht(a=>captureagent::$var(capture_id)) = 0;
    $sht(a=>captureagent::$var(capture_id)) = $sht(a=>captureagent::$var(capture_id)) + 1;
    /* you can make statistics here */

    $var(proto) = $(var(data){s.int});

    if($var(proto) == 100) {
        /* send this data to logs_capture */
        report_capture("logs_capture", "$var(callid)");
        //dont send data to route/onreply_route
        drop;
    }
    else if($var(proto) == 50) {
        /* send this data to RTCP */
        report_capture("rtcp_capture", "$var(callid)");
        //dont send data to route/onreply_route
        drop;
    }
    else if($var(proto) == 1) {
        /* SIP */
        /* send this data to reply or onreply route, sipcapture will call parse_sip after payload will be extracted */
        return;
    }
}

```



ANY VoIP System

What's NEXT with HOMER 5

A brief look into the future development and roadmap



Elasticsearch (deeper) Integration

HOMER 5 natively supports querying *Elasticsearch* data from *Dashboard* widgets/charts, but not for session data... until now! Thanks to the power of our **HEP Event Socket** you can now ship selected packets to your powerful Elasticsearch Cluster or any other JSON store. Here's a fully working example using **sipcapture + jansson + http_client_async** modules

```
event_route[sipcapture:request] {

    xlog("received sipcapture request from $si:$sp\r\n");

    /* get proto type id from HEP header */
    hep_get("0x0B", "0x0000", "$var(data)");
    hep_get("0x0F", "0x0000", "$var(payload)");
    hep_get("0x11", "0x0000", "$var(correlation_id)");

    $var(proto) = $(var(data)[s_int]);

    if($var(proto) == 100) {

        # create a transaction to be paused, and resumed in route[HTTP_REPLY]
        t_newtran();
        #Create json object for ElasticSearch
        jansson_set("string", "body", "$var(payload)", "$var(json)");
        jansson_set("string", "postDate", "2016-05-20", "$var(json)");
        jansson_set("string", "title", "Data from Homer", "$var(json)");
        #Send to ES
        http_async_query("http://elasticsearch:9200/homer/logs", "$var(json)", "HTTP_REPLY");
        drop;
    }
}
```

```
else if($var(proto) == 1) {
    /* SIP */
    # create a transaction to be paused, and resumed in route[HTTP_REPLY]
    t_newtran();
    #Create json object for ElasticSearch
    jansson_set("string", "body", "$var(payload)", "$var(json)");
    jansson_set("string", "postDate", "2016-05-20", "$var(json)");
    jansson_set("string", "title", "SIP from Homer", "$var(json)");
    #Send to ES
    http_async_query("http://elasticsearch:9200/homer/sip", "$var(json)", "HTTP_REPLY");
    drop;
}

drop;
}

route[HTTP_REPLY] {
    if ($http_ok) {
        xlog("L_INFO", "route[HTTP_REPLY]: status $http_rs\n");
        xlog("L_INFO", "route[HTTP_REPLY]: body $http_rb\n");
    } else {
        xlog("L_INFO", "route[HTTP_REPLY]: error $http_err\n");
    }
}
```




What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

Here's a quick example we created around 2AM last night - make it your own and start pairing data in your **SIPCAPTURE** stack to **Elasticsearch**!

Have a good Idea or good business case?
Send us a mail and help us shape the future of this feature to be as flexible as we can make it! ;)

#opensource

Table JSON

```

1  {
2  " _index": "homer",
3  " _type": "logs",
4  " _id": "AVTLetqotasK96kVfXPa",
5  " _score": null,
6  " _source": {
7  " @timestamp": "2016-05-20T01:00+0000",
8  " ts": 1463700600000,
9  " value": 1,
10 " string": "Hello Elasticsearch, this is HOMER speaking!",
11 " group": "callid"
12 },

```

The screenshot shows the HOMER 5 interface with a table of SIP signaling events. A call ID is selected, and an Elasticsearch result window is open, showing the JSON log entry for that call. A large arrow points from the JSON example above to the Elasticsearch result window.

Id	Date	Method
23250	2016-05-20 00:21:44.023 +...	REGISTER
23249	2016-05-20 00:21:44.023 +...	401
23251	2016-05-20 00:21:44.079 +...	REGISTER
23248	2016-05-20 00:21:44.080 +...	403
23253	2016-05-20 00:21:56.318 +...	REGISTER
23252	2016-05-20 00:21:56.319 +...	401
23255	2016-05-20 00:21:56.434 +...	REGISTER

Call-ID: 16b50d4f-5f8...-573e3c7d@109.69.67.241:5080

Elasticsearch Result from :

```

{"_index":"homer","_type":"logs","_id":"AVTLetqotasK96kVfXPa","_score":1,"_source":{"@timestamp":"2016-05-20T01:00+0000","ts":1463700600000,"value":1,"string":"Hello Elasticsearch, this is HOMER speaking!","group":"callid"}}

```

What's **NEXT** with **HOMER 5**

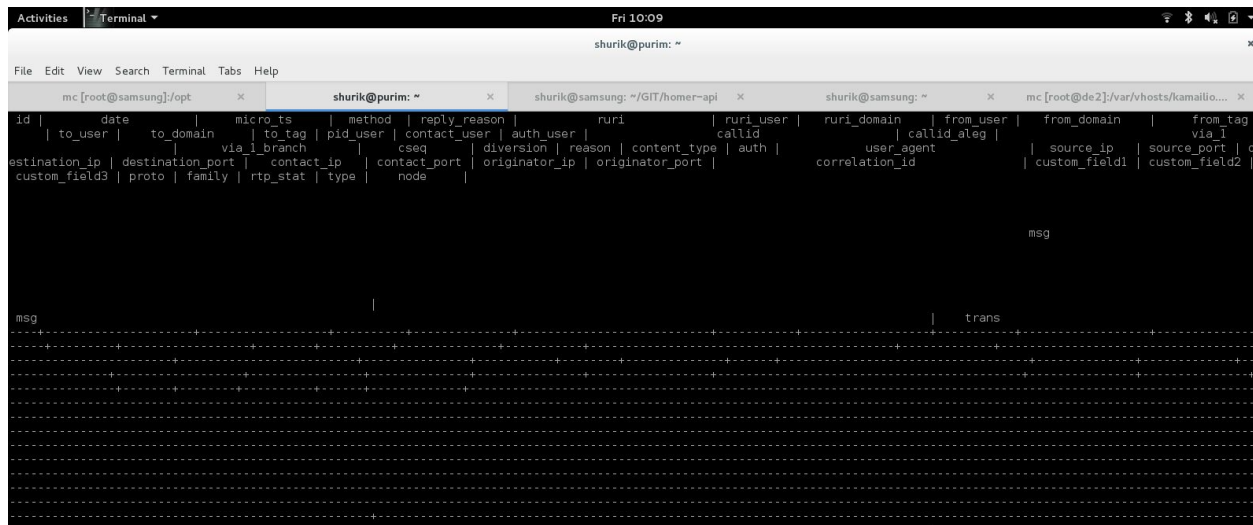
A brief look into the future development and roadmap



Postgres SQL

HOMER 5 natively ships with an “agnostic” database module virtually supporting multiple database backends, but in reality only the **MySQL** connector was fully developed.... Until last night!

While you were enjoying the drinks kindly offered by the conference, we were busy making this happen!
(Thanks William King for the onsite assistance fine tuning!)



```

Activities | Terminal
Fri 10:09
shurik@purim: ~

File Edit View Search Terminal Tabs Help

mc [root@samsung]/opt x shurik@purim: ~ x shurik@samsung: ~/GIT/homer-api x shurik@samsung: ~ x mc [root@de2]:/var/vhosts/kamailio... x

id | date | micro_ts | method | reply_reason | ruri | ruri_user | ruri_domain | from_user | from_domain | from_tag |
| to_user | to_domain | to_tag | pid_user | contact_user | auth_user | callid | callid_aleg | | | via_1 |
via_1_branch | cseq | diversion | reason | content_type | auth | user_agent | source_ip | source_port | d
estination_ip | destination_port | contact_ip | contact_port | originator_ip | originator_port | correlation_id | custom_field1 | custom_field2 |
custom_field3 | proto | family | rtp_stat | type | node |

msg

msg | trans
  
```

What's NEXT with HOMER 5

A brief look into the future development and roadmap



Postgres SQL

HOMER 5 natively ships with an “agnostic” database module virtually supporting multiple database backends, but in reality only the **MySQL** connector was fully developed... Until last night!

While you were enjoying the drinks kindly offered by the conference (Thanks William King for the onsite assistance fine tuning!)

Activities Terminal Fri 10:09 shurik@purim: ~

File Edit View Search Terminal Tabs Help

```
mc [root@samsung]/opt x shurik@purim: ~ x shurik@samsung: ~/GIT/homer-api x shurik@samsung: ~
```

```
May 20 09:58:42 purim kernel: [4273445.006037] device eth0 entered promiscuous mode
```

```
May 20 10:00:12 purim kernel: [4273535.407054] device eth0 left promiscuous mode
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

```
May 20 10:00:58 purim /usr/local/kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query (PGRES_FATAL_ERROR)
```

Activities Terminal Fri 10:09 shurik@purim: ~

File Edit View Search Terminal Tabs Help

```
mc [root@samsung]/opt x shurik@purim: ~ x shurik@samsung: ~/GIT/homer-api x shurik@samsung: ~
```

id	date	micro_ts	method	reply_reason	ruri	ruri_user	ruri_domain
	to_user	to_domain	pid_user	contact_user	auth_user		
		via_i_branch	cseq	diversion	reason	content_type	auth
	destination_ip	destination_port	contact_ip	contact_port	originator_ip	originator_port	cal
	custom_field3	proto	family	rtp_stat	type	node	correlation_id

```
msg
```

```
-----
```

```
-----
```

```
-----
```

```
-----
```

```
-----
```

```
-----
```

```
-----
```

What's NEXT with HOMER 5

A brief look into the future development and roadmap



Postgres SQL

```

shurik@purim: ~
shurik@samsung: ~/GIT/homer-api x
shurik@samsung: ~ x
mc [root@de2]: /var/vhosts/kamailio... x

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...to_date, method, country, lat, lon
^#012HINT: No function matches the given name and argument types. You might need
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...to_date, method, country, lat, lon
^#012HINT: No function matches the given name and argument types. You might need
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...ats_data (from_date, to_date, type
^#012HINT: No function matches the given name and argument types. You might need
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...ats_data (from_date, to_date, type
^#012HINT: No function matches the given name and argument types. You might need
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...ats_data (from_date, to_date, type
^#012HINT: No function matches the given name and argument types. You might need
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...ats_data (from_date, to_date, type
^#012HINT: No function matches the given name and argument types. You might need
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...ats_data (from_date, to_date, type
^#012HINT: No function matches the given name and argument types. You might need
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...to_date, method, source_ip,
^#012HINT: No function matches the given name and argument types. You might need to
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...to_date, method, source_ip,
^#012HINT: No function matches the given name and argument types. You might need to
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...to_date, method, source_ip,
^#012HINT: No function matches the given name and argument types. You might need to
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...to_date, method, source_ip,
^#012HINT: No function matches the given name and argument types. You might need to
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:244]: db_postgres_submit_query()
from unixtime(integer, unknown) does not exist#012LINE 1: ...to_date, method, country, lat, lon,
^#012HINT: No function matches the given name and argument types. You might need to
kamailio-dev/sbin/kamailio[29296]: WARNING: db_postgres [km_dbase.c:240]: db_postgres_submit_query()

```

backends,



What's NEXT with HOMER 5

A brief look into the future development and roadmap

Postgres SQL

Activities Terminal Thu 20:50

dev.sipcapture.org/#/result

HOMER Search Panels Last 6 Hours

SIP Signaling Search

Id	Date	Method	Reason	RURI user	From User	To User	CallID	CallID_AL	User Agent	Source Host	SPort	Destination Ho..	DPort	Pro..	Node
2	2016-05-19 20:39:19.141 +...	REGISTER			104	104	ff4df6e1-cf634572-2328		PolycomVX-V	92.204.1.151	5060	109.69.67.241	5060	udp	homer01:301
1	2016-05-19 20:39:19.143 +...	200	OK												
4	2016-05-19 20:39:30.649 +...	REGISTER													
3	2016-05-19 20:39:30.650 +...	401	Una												
6	2016-05-19 20:39:30.764 +...	REGISTER													
5	2016-05-19 20:39:30.765 +...	403	Forb												
9	2016-05-19 20:39:44.139 +...	REGISTER													
8	2016-05-19 20:39:44.140 +...	401	Una												
10	2016-05-19 20:39:44.188 +...	REGISTER													
7	2016-05-19 20:39:44.189 +...	403	Forb												
2	2016-05-19 20:39:56.885 +...	OPTIONS													
1	2016-05-19 20:39:56.885 +...	OPTIONS													
3	2016-05-19 20:39:56.929 +...	200	OK												

Call-ID: #4df6e1-cf634572-23287b87@192.168.178.32

shurik@purim: ~

File Edit View Search Terminal Tabs Help

```

id | date | micro_ts | method | reply_reason |
ruri | ruri_user | ruri_domain | from_user | from domain |
from_tag | to_user | to_domain | to_tag | pid_user | cont |
user | auth_user | callid | | callid_alleg |
via_1 | | via_1_branch |
cseq | diversion | reason | content_type | auth | use |
ent | source_ip | source_port | destination_ip | destination |
contact_ip | contact_port | originator_ip | originator_port |
correlation_id | | custom_field1 | custom_field2 | custom_fie |
proto | family | rtp_stat | type | node |

```

MSG ID: 2

Message Details

2016-05-19 20:39:19 +0200 : 92.204.1.151:5060

92.204.1.151:5060

sip:1:REGISTER (AUTH)

2016-05-19 20:39:19.141+0200

REGISTER sip:sip2.botauro.com:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.178.32;branch=z9hG4bK
From: "104" <sip:104@sip2.botauro.com>;tag=80
To: <sip:104@sip2.botauro.com>
CSeq: 3053 REGISTER
Call-ID: ff4df6e1-cf634572-23287b87@192.168.178.32
Contact: <sip:104@192.168.178.32>;methods="MESSAGE, SUBSCRIBE, NOTIFY, PRACK, UPD
User-Agent: PolycomVX-VX_500-UA/4.0.1.1503
Accept-Language: en
Authorization: Digest username="104", realm="sip:192.168.178.32", digest-uri="sip:192.168.178.32", response="937f

What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap



HOMER ALARMS HOW-TO

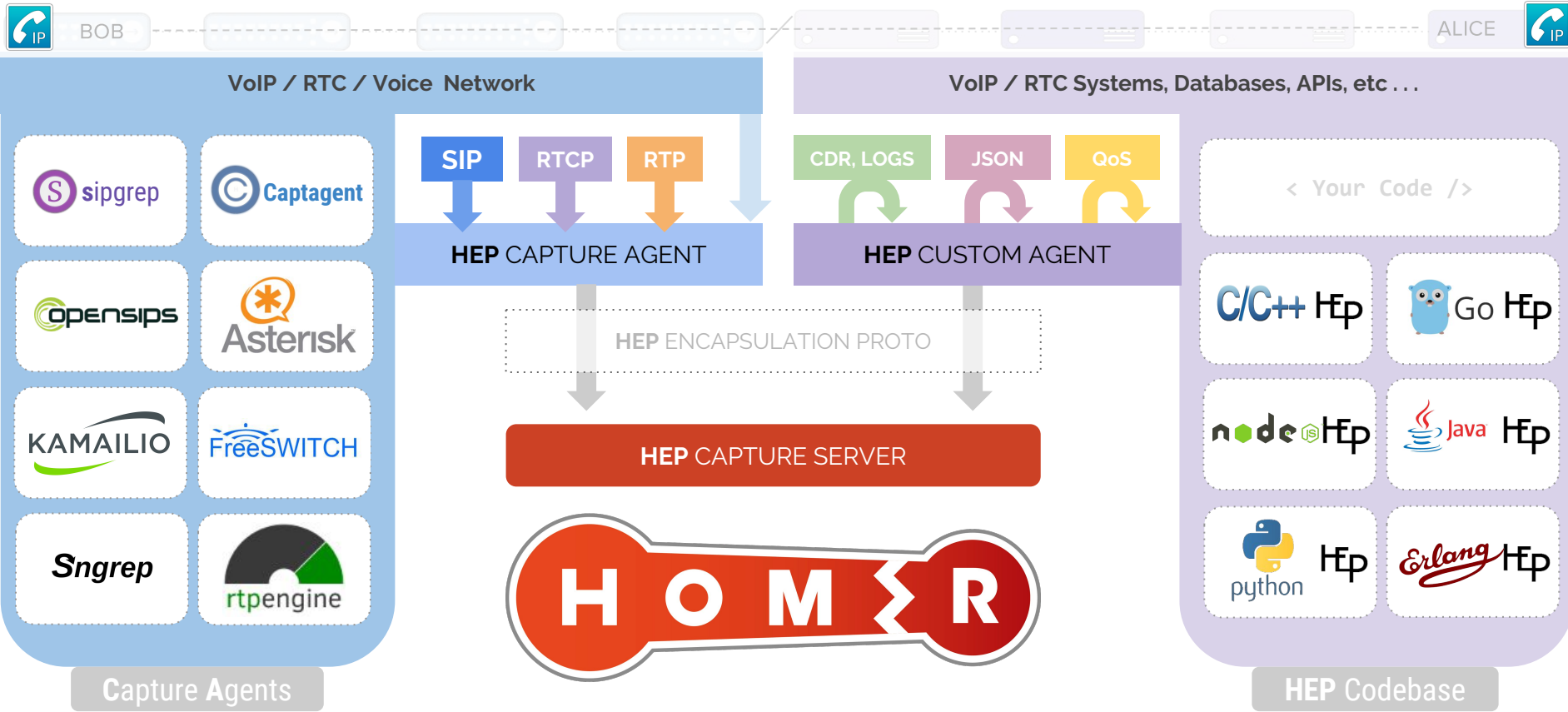
Members of the audience asked us to add a few examples for Alarms creation and management - in a nutshell:

- 1) Create alarms in kamailio.cfg
- 2) Create notification block in kamailio.cfg
- 3) Create UI trigger in H5 user-interface admin

```
if($ua =~ "(friendly-scanner|sipvicious|sipcli)") {
    $var(atype) = 'scanner';
    sql_query("cb", "INSERT INTO alarm_data_mem (create_date, type, total, source_ip, description) VALUES(NOW(), '$var(atype)', 1, '$si', 'Friendly scanner alarm!') ON DUPLICATE KEY UPDATE total=total+1");
    route(KILL_VICIOUS);
}
#Alarm for Scanner;
if($var(atype) == "scanner") {
    sql_query("cb", "DELETE FROM alarm_data_mem WHERE type='scanner' AND total < $var(avalue)");
    if($var(anotify) == 1) {
        sql_query("cb", "SELECT * FROM alarm_data_mem WHERE type='scanner' AND total >= $var(avalue) LIMIT 2", "rd");
        if($dbr(rd=>rows) > 0) {
            route(SEND_ALARM);
        } sql_result_free("rd");
    }
}
}
```

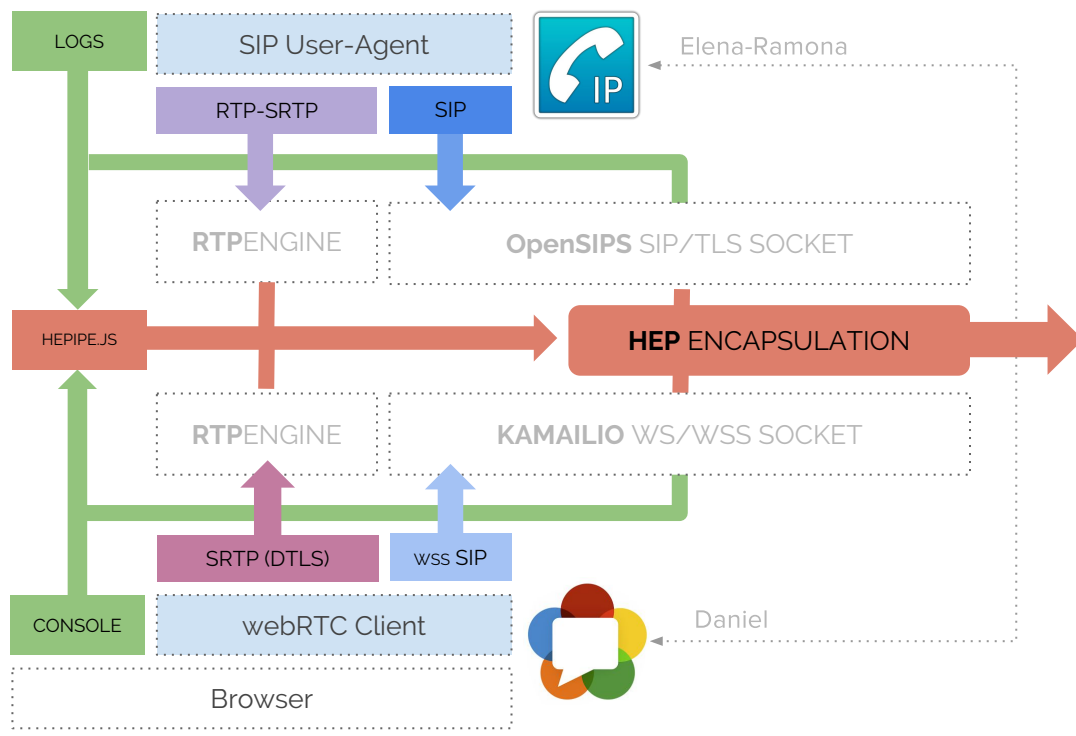
```
route[SEND_ALARM] {
    exec_msg('echo "Value: $var(thvalue), Type: $var(atype), Desc: $var(aname)" | mail -s "HOMER ALERT $var(atype) - $var(thvalue)" $var(aemail)');
}
```

SIPCAPTURE HEP/EEP Stack



Kamailio/SER Monitoring

Example Illustration SIP to SIP/WSS via SER Proxies and RTP Relays



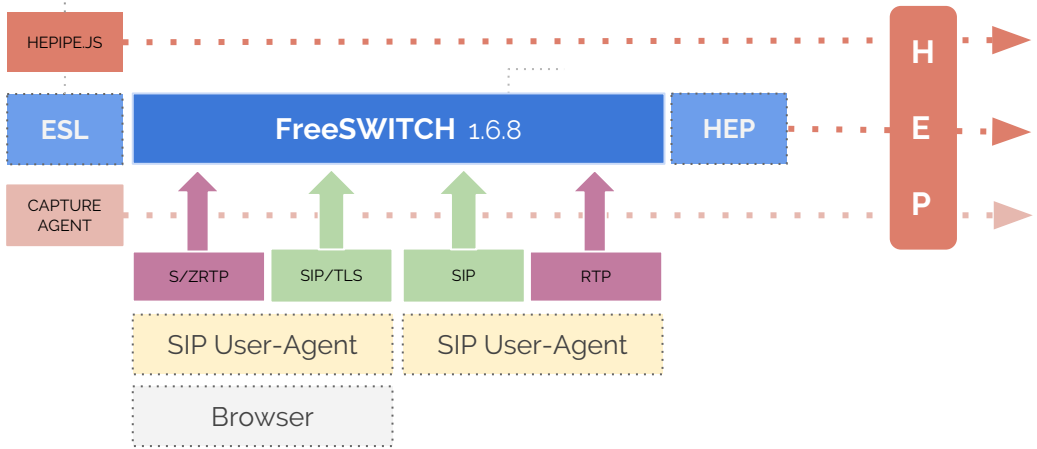
FreeSWITCH Monitoring

Example Illustration of Multi-Level Monitoring in FS HEP + ESL

CALL-ID
BLEG_CALL-ID

CHANNEL_CREATE
CHANNEL_ANSWER
CHANNEL_DESTROY

variable_rtp_audio_in_jitter_min_variance: **63.29**
 variable_rtp_audio_in_jitter_loss_rate: **0.00**
 variable_rtp_audio_in_jitter_burst_rate: **0.00**
 variable_rtp_audio_in_mean_interval: **20.11**
 variable_rtp_audio_in_flaw_total: 0
 variable_rtp_audio_in_mos: 4.50



FreeSWITCH HEP/EEP Configuration

Example Usage of the Integrated Capture Agent for Monitoring

FreeSWITCH ships with a built-in HEP agent used to mirror/transfer packets unmodified and carries timestamp and several session key values in its headers, designed for capturing simple and complex scenarios with minimal configuration efforts.

To enable **HEP** capturing, open *sofia.conf.xml* and set capture-server param:

```
<param name="capture-server" value="udp:10.0.0.1:9060" />
```

NEW! Freeswitch v1.6.8 (*master git*) now supports **HEPv2** + **HEPv3/EEP** encapsulation & parameters:

```
<param name="capture-server" value="udp:10.0.0.1:9060;hep=3;capture_id=100" />
```

To enable the **HEP** capture agent globally, open *internal.xml* and change sip-capture param to "yes"

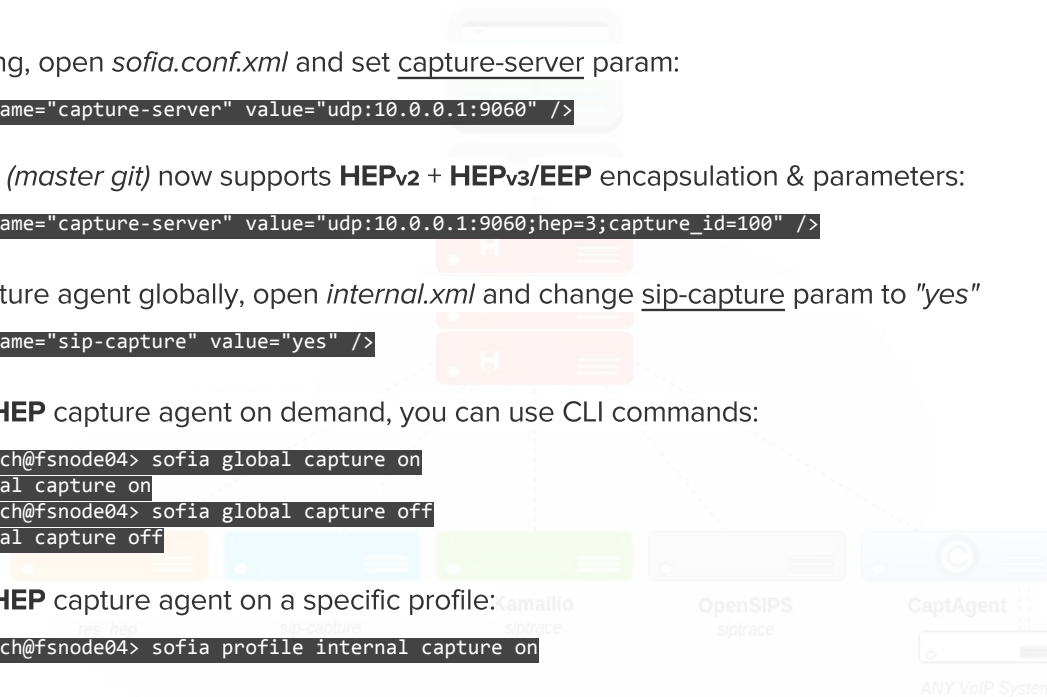
```
<param name="sip-capture" value="yes" />
```

To enable/disable the **HEP** capture agent on demand, you can use CLI commands:

```
freeswitch@fsnode04> sofia global capture on  
+OK Global capture on  
freeswitch@fsnode04> sofia global capture off  
+OK Global capture off
```

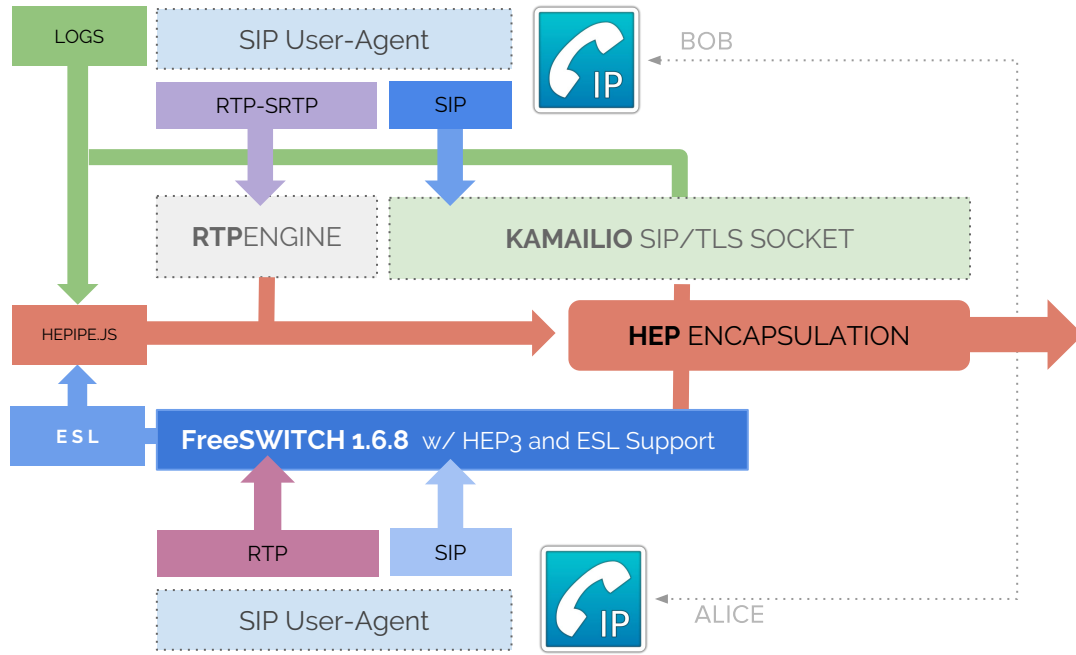
To enable/disable the **HEP** capture agent on a specific profile:

```
freeswitch@fsnode04> sofia profile internal capture on
```



FreeSWITCH + Kamailio Monitoring

Example Illustration SIP + RTP via Load Balancer w/ Correlation



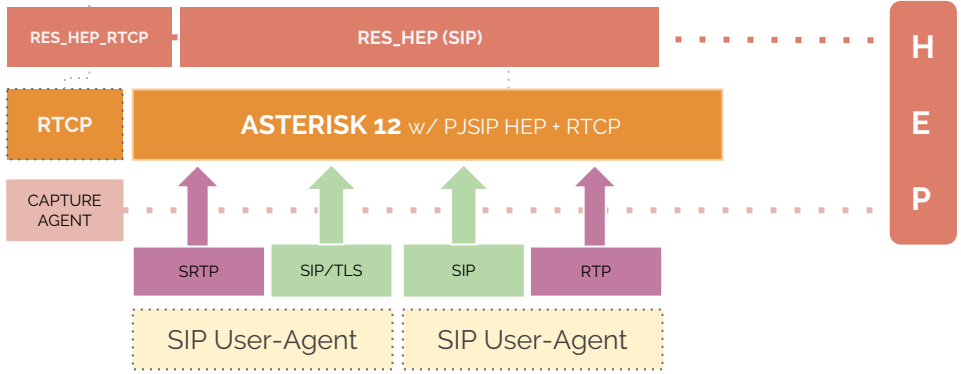
Asterisk Monitoring

Example Illustration of Multi-Level Monitoring in Asterisk 12+



PJSIP CALL-ID

```
{"ssrc":644444017,"type":200,"sender_information":{"rtp_timestamp":1340626419,"ntp_timestamp_sec":2086871206,"octets":484780,"ntp_timestamp_usec":4283484972,"packets":24239,"report_count":1,"report_blocks":[{"source_ssrc":829846894,"highest_seq_no":438,"fraction_lost":0,"packets_lost":0,"dlsr":0,"ia_jitter":3,"lsr":"0"}]}
```



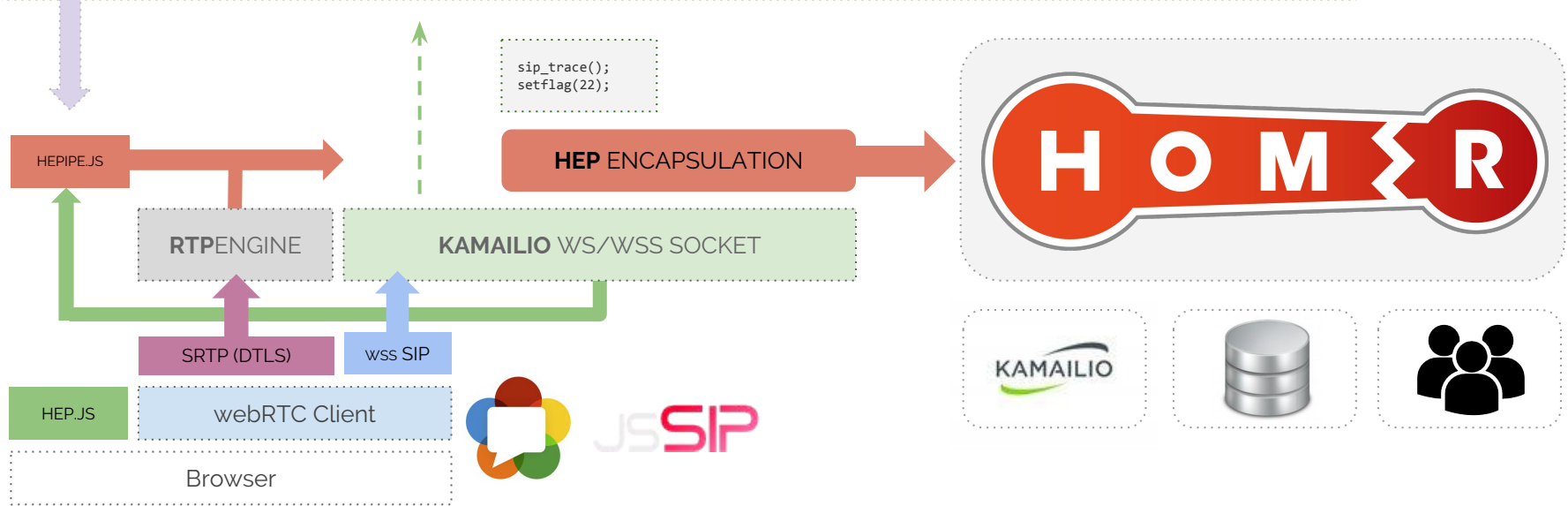


Kamailio WSS Monitoring

<http://github.com/sipcapture/wiki>

```
if (proto == WS || proto == WSS) { setflag(SRC_WS);

  xlog("L_INFO", "homerwss CID: [%ci], SIP: Method: $rm, CSEQ: $cs, RU: $rU, WSS Request: RM: $var(wss_rm), RU: $var(wss_ru),
    UAC: $var(wss_uac), Connection: $var(wss_connection), Upgrade: $var(wss_upgrade), Origin: $var(wss_origin),
    Host: $var(wss_host), Sec_Proto: $var(wss_sec_proto), Sec-Key: $var(wss_sec_key), WS_VERSION: $var(wss_sec_version)");
}
```



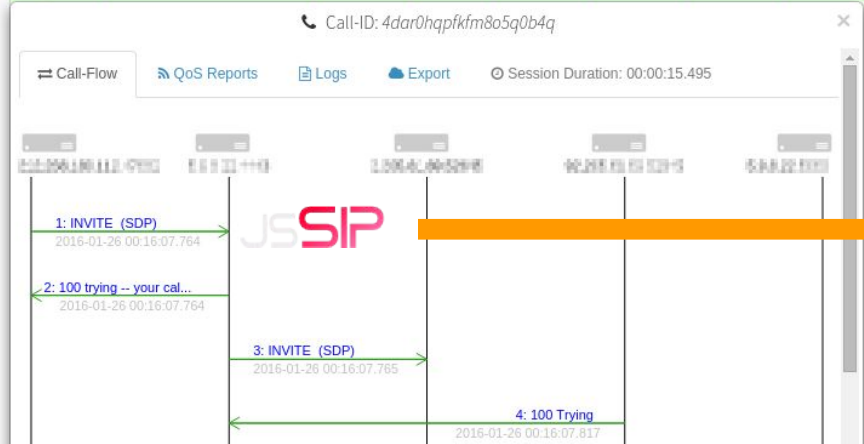


HOMER 5: WSS Call Flow

WSS to SIP Call Troubleshooting

SIP Signaling

✓	Id	Date	Method	Reason	RURI user	From User	To User	CallID	CallID_AL	User Age	Source H	SPort	Destinati	DPort	Pr	Node
✓	304	2016-01-26 00:16:15.671	200	OK		201	101	4dar0hqpfkfm8o5q0b4q			192.168.1.100	52645	192.168.1.100	4443	3	homer01:...
✓	306	2016-01-26 00:16:15.672	200	OK		201	101	4dar0hqpfkfm8o5q0b4q			192.168.1.100	4443	192.168.1.100:4443	47682	3	homer01:...
✓	307	2016-01-26 00:16:15.718	ACK		lq1pna1u	201	101	4dar0hqpfkfm8o5q0b4q		JsSIP 0.7...	192.168.1.100	47682	192.168.1.100	4443	3	homer01:...
✓	308	2016-01-26 00:16:22.192	BYE		lq1pna1u	201	101	4dar0hqpfkfm8o5q0b4q		JsSIP 0.7...	192.168.1.100	47682	192.168.1.100	4443	3	homer01:...
✓	309	2016-01-26 00:16:22.192	BYE		lq1pna1u	201	101	4dar0hqpfkfm8o5q0b4q		JsSIP 0.7...	192.168.1.100	5060	192.168.1.100	52645	3	homer01:...
✓	310	2016-01-26 00:16:22.258	200	OK		201	101	4dar0hqpfkfm8o5q0b4q			192.168.1.100	52645	192.168.1.100	4443	3	homer01:...
✓	311	2016-01-26 00:16:22.259	200	OK		201	101	4dar0hqpfkfm8o5q0b4q			192.168.1.100	4443	192.168.1.100:4443	47682	3	homer01:...



Call-ID: 4dar0hqpfkfm8o5q0b4q

Call-Flow | QoS Reports | Logs | Export | Session Duration: 00:00:15.495

Filter Logs

Jan 26 00:16:07 de2 /usr/local/kamailio-dev/sbin/kamailio[30724]: INFO: <script>: homerwss CID: [4dar0hqpfkfm8o5q0b4q], SIP: Method: INVITE, CSEQ: 2592, RU: 101, WSS Request: RM: GET, RU: GET, UAC Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.82 Safari/ Connection: Upgrade, Upgrade: websocket, Origin: https://qxip.net, Host: 192.168.1.100:4443, Sec_Proto: sip, Sec-Key /DVdxELik/RSckW2qnVntQ==, WS_VERSION: 13

Jan 26 00:16:15 de2 /usr/local/kamailio-dev/sbin/kamailio[30723]: INFO: <script>: homerwss CID: [4dar0hqpfkfm8o5q0b4q], SIP: Method: ACK, CSEQ: 2592, RU: lq1pna1u, WSS Request: RM: GET, RU: GET, U Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36, Connection: Upgrade, Upgrade: websocket, Origin: https://www.qxip.net, Host: 192.168.1.100:4443, Sec_Proto: sip, Sec-Key gmB8o/jekG74S2v3wEi8Q==, WS_VERSION: 13

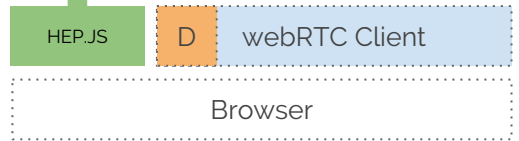
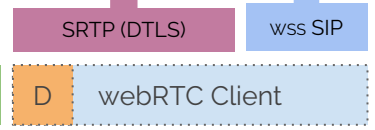
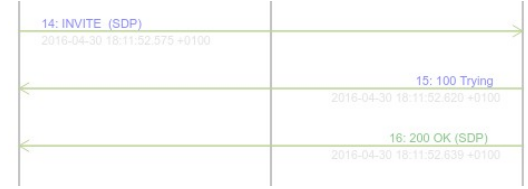
UA Remote Log Monitoring

<http://github.com/sipcapture/hepipe-js>



```

JsSIP:Transport WebSocket disconnected (code: 1006) +2m
jssip.js:22725 JsSIP:ERROR:Transport WebSocket abrupt disconnection +0ms
jssip.js:22550 JsSIP:Transport trying to reconnect to WebSocket wss://1.2.3.4:4443 jssip.js:
22550 JsSIP:Transport connecting to WebSocket wss://1.2.3.4:4443 +4s
jssip.js:22550 JsSIP:Transport WebSocket wss://1.2.3.4:4443 connected +132 ms
  
```



JS SIP



CAPTAGENT 6.1

Universal Modular Capture Agent w/ HEP3 Support

Captagent is a powerful, flexible, completely modular capture agent *framework* ready for virtually any kind of protocol and encapsulation method - past, present *and future* ;)

Currently available modules:

- ★ **SOCKET** Modules
 - Socket PCAP, Socket RAW, Socket RTCP-XR, Socket PF_RING
- ★ **PROTOCOL** Modules
 - SIP, RTCP and other signaling and controlling Protocols
- ★ **API** Module
 - HTTP JSON REST API for Control and Statistics
- ★ **TRANSPORT** Modules
 - HEP Encapsulation output (v1/2/3)
 - JSON Serialization output
- ★ **ENCRYPTION** Modules
 - Encryption and Compression Module for HEP3
 - TLS Decryption pipeline for supported key exchange methods
- ★ **DATABASE** Modules
 - HASH Table
 - Redis interface
 - MySQL interface

CAPTAGENT: <https://github.com/sipcapture/captagent>

```

<!-- CORE MODULES -->

  <module name="transport_hep" description="HEP Protocol" serial="2014010402">
    <profile name="hepsocket" description="Transport HEP" enable="true"
    serial="2014010402">
      <settings>
        <param name="version" value="3"/>
        <param name="capture-host" value="127.0.0.1"/>
        <param name="capture-port" value="9061"/>
        <param name="capture-proto" value="udp"/>
        <param name="capture-id" value="2001"/>
        <param name="capture-password" value="myhep"/>
        <param name="payload-compression" value="false"/>
      </settings>
    </profile>
  </module>

<!-- PROTOCOLS -->

  <module name="socket_pcap" description="HEP Socket" serial="2014010402">
    <profile name="socketspcap_sip" description="HEP Socket" enable="true"
    serial="2014010402">
      <settings>
        <param name="dev" value="eth0"/>
        <param name="promisc" value="true"/>
        <param name="reasm" value="false"/>
        <param name="capture-plan" value="sip_capture_plan.cfg"/>
        <param name="filter">
          <value>portrange 5060-5091</value>
        </param>
      </settings>
    </profile>
  </module>

```

CAPTAGENT 6.1 *(continued)*

```

<module name="socket_pcap" description="HEP Socket" serial="2014010402">
<profile name="socketspcap_sip" description="HEP Socket" enable="true"
serial="2014010402">
  <settings>
    <param name="dev" value="any"/>
    <param name="promisc" value="true"/>
    <param name="reasm" value="false"/>
    <param name="tcpdefrag" value="false"/>
    <param name="capture-plan" value="sip_capture_plan.cfg"/>
    <param name="filter">
      <value>portrange 5060-5091</value>
    </param>
  </settings>
</profile>

<profile name="socketspcap_rtcp" description="RTCP Socket" enable="true"
serial="2014010402">
  <settings>
    <param name="dev" value="any"/>
    <param name="promisc" value="true"/>
    <param name="reasm" value="false"/>
    <param name="capture-plan" value="rtcp_capture_plan.cfg"/>
    <param name="filter">
      <value>portrange 30000-50000</value>
    </param>
  </settings>
</profile>
</module>

```

socket_pcap.xml

Cascading Capture Plan example for SIP

```

#sip_capture_plan.cfg
capture[pcap] {

  # here we can check source/destination IP/port, message size
  if(msg_check("size", "100")) {

    #Do parsing
    while(parse_sip()) {

      /* packet processing pipeline */
      clog("NOTICE", "parsing SIP message ");

      if(source_ip("10.0.0.1")) {
        # Multiple profiles can be defined in transport_hep.xml
        if(!send_hep("hepsocket_homer01")) {
          clog("ERROR", "Error sending HEP!!!!");
        }
      }
      else {
        # Multiple profiles can be defined in transport_hep.xml
        if(!send_hep("hepsocket_homer02")) {
          clog("ERROR", "Error sending HEP!!!!");
        }
      }
    }
  }
}
drop;
}

```

sip_capture_plan.xml

Full examples: <https://github.com/sipcapture/captagent>

CAPTAGENT 6.1 HEP/EEP RTCP + SIP Mirroring

Example Usage of the Universal Capture Agent for Monitoring

If you configured everything correctly, your HOMER 5 **QoS statistics** will start being populated:



RTCP Stats [62560 Packets]

Avg. Packet Loss	Avg. Jitter (ms)	Avg. MOS
0.0%	37.53	4.32

Tot Packets Lost	Max Jitter (ms)	Min. MOS
18	67.00	4.19

X-RTCP Stats [8131 Packets]

Avg. Packet Loss	Avg. Jitter (ms)	Avg. MOS
0.02%	59.0	4.19

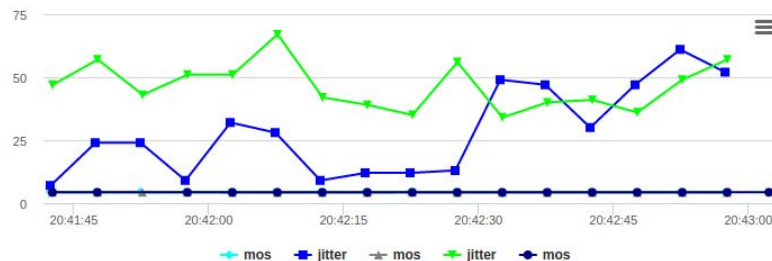
Tot Packets Lost	Max Jitter (ms)	Min. MOS
2	57.0	4.19

RTP LEG 1

RTP-1 AUDIO SRC:	RTP-1 AUDIO DST:
 rfc2833 PCMU/8000	 rfc2833 PCMU/8000

RTP-2 AUDIO SRC:	RTP-2 AUDIO DST:
 rfc2833 PCMU/8000	 rfc2833 PCMU/8000

QoS Metrics Chart



RTCP[4037467638]

- mos
- packets
- jitter
- packets_lost

RTCP[143887173]

- mos
- packets
- jitter
- packets_lost

RTCPXR

- mos
- jitter
- packets_lost

RTPAGENT PRO Modules

Commercial Capture Extensions with Advanced Functionality



RTPAgent is a “privacy-friendly” Analytics and Reporting probe for **HOMER 5** performing wire-speed RTP session and network packet analysis in-transit and in real-time without storing any data to disk (unless desired) and delivers granular periodic and final reports with a full stack of dedicated metrics at each interval:

- Source/Destination IP/PORT/MAC
- Bytes/Packets Total, Expected
- Packet Loss
- Jitter (min/man/mean)
- RTT Delta/Skew (min/man/mean)
- Codec ID, Clock Rate
- MOS Estimation
- R-Factor Estimation

RTP Reporting frequency can be defined by the integrator or self-adjusted by the probe to send higher number of periodic QoS reports for sessions where suspect quality issues are identified and to automatically reduce the number of reports for those delivering high scores in order to minimize the bandwidth overhead.

RTPAgent is designed to deal with multi-party and multi-codec calls including video sessions and can automatically detect/report a vast number of conditions.

Additional Modules:

- ★ On-Demand, Filtered Stream Recording to Disk (SIP/RTP/RTCP)
- ★ Lawful Interception (X1/2/3 ETSI 232)

```
{
  "CORRELATION_ID": "56a211936328-fgbtmubkimot",
  "RTP_SIP_CALL_ID": "56a211936328-fgbtmubkimot",
  "DELTA": 19.980,
  "JITTER": 0.023,
  "REPORT_TS": 1453461919,
  "TL_BYTE": 0,
  "SKEW": -0.180,
  "TOTAL_PK": 510,
  "EXPECTED_PK": 510,
  "PACKET_LOSS": 0,
  "SEQ": 0,
  "MAX_JITTER": 1.892, "MEAN_JITTER": 0.126,
  "MAX_DELTA": 35.547, "MAX_SKEW": -15.615,
  "MIN_MOS": 4.385, "MEAN_MOS": 4.394, "MOS": 4.394,
  "RFACOR": 92.449, "MIN_RFACOR": 92.013, "MEAN_RFACOR": 92.444,
  "SRC_IP": "192.168.178.34", "SRC_PORT": 58320, "DST_IP": "192.168.60.70", "DST_PORT": 32728,
  "SRC_MAC": "00-04-13-29-64-22", "DST_MAC": "34-31-C4-38-24-0D",
  "CODEC_PT": 9, "CLOCK": 8000, "CODEC_NAME": "g722", "DIR": 1,
  "REPORT_NAME": "192.168.178.34:58320", "PARTY": 0, "TYPE": "PERIODIC"
}
```



Q & A

HOMER 5

Dashboard and Widget management

Homer 5 brings a lot of changes and much more flexibility in line with other popular data-mining platforms around. Our evergreen **Wiki** on **Github** provides a number of useful resources to get started (or refreshed) including:

- ★ How to Install and Update Homer
- ★ How to get started with the User-Interface
- ★ How to customize Panels and Widgets
- ★ How to manage Users and Aliases
- ★ How to configure HEP Capture Agents
- ★ How to configure HEP Custom Agents
- ★ How to correlate Sessions and Reports
- ★ How to make your own Statistics and Widgets

..... and much more !

"Just HEP Yourself, to my SIPs..."

<https://github.com/sipcapture/homer/wiki/>

Welcome to the SIPCAPTURE WIKI!

Use the right menu to browse our help topics, examples and guides to learn how to **setup HOMER**, configure capture agents with secure encryption, ship custom logs, custom statistics and more!

Get started with **HOMER** by SIPCAPTURE.ORG

100% Open Source VoIP Capture and Monitoring

Learn all about our Project with our docs, webinars... | Design your capture with our tools, see, meet... | Easily install SIPCAPTURE on your system...

- Pages
- Homer 5 Wiki
 - How to Install
 - How to Install (Docker)
 - How to Install (Packages)
 - How to Update (5.x only)
- Using Homer 5
 - Dashboards
 - Administration
 - Searching
 - Customization
 - Results
 - Result Type
 - Using Timezones
 - Grid Options
 - Visualizers
 - Alarms
 - Aliases
 - Correlation
 - QoS Reports
 - Rison Parameters
- Using Homer API
 - Using DB Nodes
 - FAQ & Troubleshooting

“That’s all Folks!”



Time’s UP! Want to go further? "HEP" Yourself!

SIPCAPTURE @GITHUB	http://sipcapture.org + http://sipcapture.io
HOMER @GITHUB	http://github.com/sipcapture/homer
CAPTAGENT @GITHUB	http://github.com/sipcapture/captagent
HEPIPE.JS @GITHUB	http://github.com/sipcapture/hepipe.js
MAILING-LIST @USERS	https://groups.google.com/forum/#!forum/homer-discuss