# **SIP Troubleshooting #ONE**

# WORKSHOP



Alexandr Dubovikov & Lorenzo Mangani

**HOMER Development Team** 

http://sipcapture.org





Workshop Sponsored by QXIP / NTOP - http://qxip.net



# Who are we?



#### meet the **SIPCAPTURE** Development Team!

proud makers of

#### Alexandr Dubovikov

Senior Voice Expert for QSC AG, one of the major German voice and data providers. Alexandr holds a diploma in physics of Odessa State University and brings 20 years of experience in telecommunication techniques, contributing to many Open Source projects like FreeSwitch, SER, Kamailio, SEMS, Asterisk, SIPp, Wireshark. Alexandr is the main developer of Homer SIP Capture project. Also founder of IRC RusNet Network, one of the biggest national IRC networks in the world.



#### Lorenzo Mangani

Sr. Voice Engineer and Designer for the largest international cable operator worldwide, founder of Amsterdam based QXIP BV, Co-Founder and Developer of Homer SIP Capture project and voice specialist of the NTOP Team. Formerly a Sound Engineer, Lorenzo has been deeply involved with telecommunications and VoIP for well over a decade and has contributed ideas, design concepts and code to many voice-related Open-Source and commercial projects specializing in active and passive monitoring solutions.



#### Joseph Jackson

Sr. Network Engineer for VoIP Long Distance wholesale provider. Specializing in high performance and redundant network design with a special interest in high speed packet capturing and analysis. Focusing on providing real time VoIP metrics.



# Who are you?

In order to adapt the speed and phasing of this workshop to a fair median we would like to quickly scope our audience

(please raise your hand when a matching group is mentioned)

Voice Pal	Works with SIP occasionally and/or deals with other aspects of the network/business
Voice OP	Works with SIP daily, dealing with real cases/solutions practicing deep commandline-fu
Voice Dev	Works with SIP all day, leads or contributes to several Voice related projects all night



# SIP Troubleshooting #1: Toolset in 30 minutes

We all know it - SIP is an ASCII/UTF-8 application-layer control protocol defined by RFC3261 that can initiate, modify and terminate sessions, presenting a wide variety of header fields, often carrying additional body data such as SDP used to allow RFC3550 endpoint RTP communication.

If you work with SIP & RTP you know they can bring both tears of joy and pain - on the other hand, we would be jobless if it all was perfect ;)

This brief workshop will *attempt* to cover:

- tools of the trade to get the job done from the "one-off" session to permanent capture setups
- technical approaches and quick recipes for capturing SIP/RTP network packets in all weather conditions
- relevant community references, useful resources, ideas and links
- tools we ourselves developed to make your voice life a little easier

This workshop will unfortunately <u>not</u> cover:

- how to master SIP Protocol and its every RFC in less than 30 minutes w/ free drinks
- how to read packet captures blindfolded and complete SIP investigations using sniffer dogs
- techniques for capturing and decoding audio streams using the power of your mind and arduino

NOTE: Several Tools and Tool Suites will be referenced during this workshop, while most of them are freely available and/or fully Open-Source we decided to also mention and compare the features of some relevant commercial solutions suited as companions or extension to Open-Source components for completeness of analysis of the options in the higher end of the scope and for those in need of them. The choice is yours!



# SIP Troubleshooting

# **INTRODUCTION** Battlefield Hardline: VoIP



# Introduction VoIP Breakdown of Typical Areas of Investigation

Although issues with SIP setups can manifest themselves in many forms and shapes, the vast majority of them can be covered by investigating the following critical areas:

#### - INTEROPERABILITY ISSUES

- Different vendors of semi compatible "standard" solutions
- o Different Interpretations and Implementations of RFCs and Standards in UAs
- Misconfiguration of remote party/interconnect (the hardest to prove and argue)

#### - NEGOTIATION ISSUES

- No common codecs or rates (ptime), DTMF transport/tone mismatch
- No network path, NAT Detection and resolve Issues. (Vendor: A)
- SDP from hell (Vendor: C) multiple Via: 127.0.0.1

#### - SYSTEM PERFORMANCE ISSUES

- Stressed or Misconfigured Hardware/Software on either side of the call
- Overloaded Transcoders, Gateways, etc.
- Attacks/Scanning/DDOS attacks overloading voice sub-systems

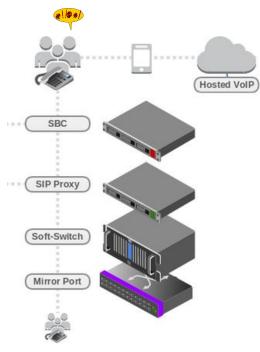
#### - NETWORK & NETWORK PERFORMANCE

- Routing Issues, NAT Issues, SIP ALG Issues
- Latency, Jitter and UDP Packet Loss in transit

#### - OSI-8 ERROR

Dial Errors, Broken Handsets, Broken B-Party Handset, Broken Ears

#### NEXT: How do we get to the juicy protocols out?



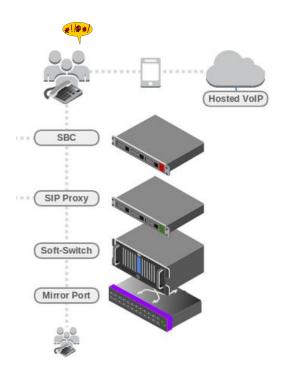


### Introduction VoIP Ecosystem and Elements

This workshop assumes basic familiarity with the standard elements and protocols typically involved with SIP Services and their roles. Beyond their implicit functional differences each system can produce plenty of valuable information and useful details:

Some Examples:

- SIP USER-AGENT, SBC / B2BUA, SSW
  - SIP, RTP, RTCP, CDRs, QoS Metrics, Application Logs
- SIP PROXY, REGISTRAR, ROUTER
  - SIP, Database & Application Logs
- MIRRORED ROUTER/SWITCH Ports
  - $\circ$   $\qquad$  SIP, RTP, RTCP protocol traffic to/from peering networks
- OSI-8 / END-USER
  - Usage Logs, Issue Timestamps, Ultra Mean Opinion Score





# CAPTURING THE PACKETS PHYSICAL METHODS

# **SPAN / MIRROR PORT**

Traditional method of capturing data

PRO:

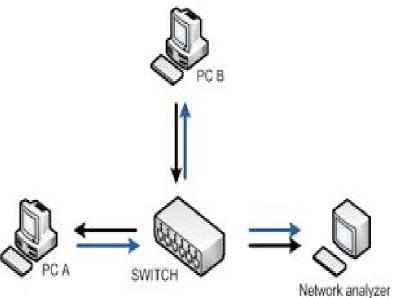
Widely support on almost any "managed switch" even small work group swi Easy to use.

Bidirectional traffic flows.

Many to one (depending on hardware).

#### CON:

Dumb capturing - no control of traffic selection. Packet loss on over subscribed destination port. Bad or corrupted packets are not transmitted.





## CAPTURING THE PACKETS PHYSICAL METHODS

# Granular Packet Capturing.

Vendor dependant and platform dependant.

Cisco:

Vlan Access Control List (VACL)

Mark interesting traffic and only have that sent to capture port, rest of traffic is forwarded as normal

VLAN based - if all your devices are in the same vlan you won't see intra vlan traffic.

Platform dependant - Catalyst 6500 and the Nexxus

Flow-based SPAN alt to VACL on Catalyst X Series.

Juniper:

Much more robust capturing.

Using Firewall filters can mark interesting traffic and forward to capture port.

Depending on model (SRX) can drop to linux OS and run tcpdump.



### CAPTURING THE PACKETS PHYSICAL METHODS

# **Network Taps**

Active Taps:

Intelligent - able to identify traffic based on layers 3-7 and send to capture device.

Pros: Intelligent and programmable.

Cons: Epensive

Passive Taps:

Dumb tap. All traffic is replicated to the capture ports.

Pros: Cheap!

Cons: Super dumb

\*Remember never plug a TX into a capture port on an optical tap\*



SIP Troubleshooting

# **REAL-TIME CAPTURE TOOLS** Terminal Heroes



# Standard Tools The ABC of packet capturing

"Everybody lies, but not SIP " Doctor House



Let's face it - If the packets we need are not available for us to investigate when we need them, we're in trouble.

Regardless of the title or experience, a good voice engineer should be prepared to do whatever the conditions dictate to capture voice packets needed to get the job done. Sometimes we own the systems and can pick our fancy weapons, other times we are bound to strict limitations - *you simply never know* - this is why the *ABC* really never gets *too* old.

Amongst the "evergreen" packet capture tools every voice op should know and use, we will briefly mention:

tcpdump, wireshark, tshark, ngrep, sipgrep, sngrep, pcapsipdump, captagent

Several of the above will offer overlapping features and/or equivalents to perform similar actions - this is great news for any voice generalist, as you never know which default tools will be found waiting for you on an impaired alien system.

NEXT: Let's see a few everyone should be familiar with...



### Standard Tools The old school ways: 8 bits games, tcpdump

Let's assume everyone knows **tcpdump**, the grandfather of packet capture tools and highlander of any unix system. **tcpdump** familiarity is definitely not an optional - when everything else fails, this good old friend won't let you down.

#### Capturing SIP Packets with tcpdump:

Display SIP packets with verbose details:

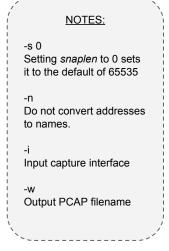
# tcpdump -nqt -s 0 -A -vvv -i eth0 port 5060

Capture SIP packets to disk in PCAP format:

# tcpdump -nq -s 0 -i eth0 -w /tmp/dump.pcap port 5060

Capture SIP packets to disk in PCAP format, rotate file every 15mb w/ file timestamp:

# tcpdump -s 0 -w /tmp/capture-dep`date +%Y%m%d-%H%M%Z`.pcap -C15 udp and port 5060





## Standard Tools The old school ways: 16 bits games, tshark

**TShark** is a network protocol analyzer part of the wireshark family. It lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file delivering the power of wireshark filtering alongside many advanced functions including RTP heuristics.

#### Capturing Packets with Tshark:

Capture all SIP on specified port and switch files every hour:

# tshark -nq -i eth0 -b duration:3600 -w /tmp/trace/sip.pcap port 5080

Extract SIP Server/Client details from INVITEs:

```
# tshark -r myFile -R "sip.CSeq.method eq INVITE"
```

Capture SIP, RTP, ICMP, DNS, RTCP, and T38 traffic in a ring buffer capturing 100 50MB files continuously:

# tshark -i eth0 -o "rtp.heuristic\_rtp: TRUE" -w /tmp/capture.pcap -b filesize:51200 -b files:100 -R 'sip or rtp or icmp or dns or rtcp or t38'

Filter on RTCP packets reporting any packet loss or jitter over 30ms:

```
# tshark -i eth0 -o "rtp.heuristic_rtp: TRUE" -R 'rtcp.ssrc.fraction >= 1 or rtcp.ssrc.jitter >= 240' -V
```

Analyze RTP Network Stream Quality by portrange:

# tshark -q -f 'udp portrange 20000-30000' -o rtp.heuristic\_rtp:TRUE -z rtp,streams
Src IP addr Port Dest IP addr Port SSRC Payload Pkts Lost Max Delta(ms) Max Jitter(ms) Mean Jitter(ms)
10.1.3.143 5000 10.1.6.18 2006 0xDEE0EE8F G.711 PCMA 236 0 (0.0%) 34.83 0.83 0.37



# Standard Tools The old school ways: Remote Captures

There are occasions where you might need to capture key packets on a remote system and analyze them locally. To avoid the trouble of saving and transferring pcap files, native linux options might come handy and apply fine to severals of our available tools:

#### **Capturing Packets Remotely:**

Capture remote traffic to local pcap with tcpdump:

# ssh root@host 'tcpdump -w - -p -n -s 0 port 5060 and host 1.2.3.4' > remote\_capture.cap

Analyze a remote real-time capture stream using a local wireshark over ssh:

# wireshark -k -i <(ssh -l root 192.168.10.20 tshark -w - not tcp port 22)</pre>

Capture from remote system via named pipe, display using sipgrep and forward to HEP Collector:

# mkfifo /tmp/pcap
# ssh root@192.168.10.20 "tcpdump -s 0 -U -n -w - -i any portrange 5060-5090" > /tmp/pcap
# sipgrep -I /tmp/pcap -H udp:192.168.50.60<u>:9060</u>



# Standard Tools Decoding and Analyzing SIP TLS packet captures with Wireshark

The world is finally catching up with Encryption - this is great news for end users but can result in complications for voice ops. Unless you are capturing traffic from within your VoIP platform *(using an internal capture agent)* you might have to deal with troubleshooting TLS sessions.

Wireshark can decode SSL/TLS sessions when the following conditions are fulfilled:

- the private key of the TLS server is known (both keys might be needed if mutual TLS (=client certificate) is used)
- the TLS connections does not use a Diffie-Hellman cipher
- Wireshark captures the TLS session from the beginning (including handshake)

#### Configure Wireshark to decode SSL/TLS:

- Copy the server's private key to the PC running Wireshark. Configure Wireshark to use the key:
- Edit → Preferences → Protocols → SSL → RSA Keys List: *i.e.: ip.address.of.server,5061,sip,/opt/ssl/agent.pem*
- If the server uses Diffie-Hellman (DH) Ciphers by default you should configure the server to use other ciphers.

#### WIRESHARK EXAMPLE:

```
wireshark -o "ssl.desegment_ssl_records: TRUE" \
  -o "ssl.desegment_ssl_application_data: TRUE" \
  -o "ssl.keys_list: 4.2.2.2,5061,sip,/opt/ssl/agent.pem" \
  -o "ssl.debug_file: /tmp/wireshark.log" \
  -i eth0 -f "tcp port 5061"
```

#### TSHARK EXAMPLE:

```
tshark -o "ssl.desegment_ssl_records: TRUE" \
  -o "ssl.desegment_ssl_application_data: TRUE" \
  -o "ssl.keys_list: 4.2.2.2,5061,sip,/opt/ssl/agent.pem" \
  -o "ssl.debug_file:/tmp/tshark.log" \
  -i eth0 \
  -f "tcp port 5061"
```



SIP Troubleshooting

# **REAL-TIME CAPTURE TOOLS** Terminal Heroes pt II



### **PCAPSIPDUMP** The old school ways: Dumping SIP Sessions to PCAP files

**pcapsipdump** is a console tool for dumping SIP sessions and RTP packets *(only when available)* to disk in a fashion similar to "tcpdump -w" by creating a single PCAP per each detected SIP session with optional number filters, for later analysis.

This old-school tool can still be useful for "one-off" activities and to temporarily monitor/intercept traffic, but clearly introduces a growing level of complexity when analyzing numerous results over long time ranges or when dealing with busy networks alone.

Capture from eth0 and store all SIP sessions in /tmp/

# pcapsipdump -i eth0 -d /tmp/

pcapsipdump version 0.1.4-trunk
Usage: pcapsipdump [-fpU] [-i <interface>] [-r <file>] [-d <working directory>] [-v level]
-f Do not fork or detach from controlling terminal.
-p Do not put the interface into promiscuous mode.

- -U Make .pcap files writing 'packet-buffered' slower method, but you can use partitially written file anytime, it will be consistent.
- -v Set verbosity level (higher is more verbose).
- -n Number-filter. Only calls to/from specified number will be recorded
- -t T.38-filter. Only calls, containing T.38 payload indicated in SDP will be recorded

PCAPSIPDUMP: http://sourceforge.net/projects/pcapsipdump/



## SIPGREP<sub>2</sub> CLI Usage and Features (add images)

**Sipgrep2** is a modern pcap-aware tool command line tool to capture, filter, display and help troubleshoot SIP signaling over IP networks, allowing the user to specify extended regular expressions matching against SIP headers and with nifty extra features.

#### Some Handy Examples:

	dialog t f 232323	here From 2	user co	ontains	'2323	3232'		
eport	dialog t f 1111 -	here To u G	ser cont	ains '1	.111'	and	print	dialog
	/ only 60 ^SIP/2.0	3 replies 603' -m	without	: dialog	g mato	ch		
Dicplay		TTONS and	NOTTEV	noquost				

# Display only OPTIONS and NOTIFY requests
sipgrep '^(OPTIONS|NOTIFY)'

# Display only SUBSCRIBE dialog
sipgrep 'CSeq:\s?\d\* (SUBSCRIBE|PUBLISH|NOTIFY)' -M

# Collect all messages while pcap\_dump smaller than 20kb sipgrep -q 'filesize:20' -0 sipgrep.pcap

U 2014/03/27 10:40:25.29899 : :2051 -> :5060
BYE sip:5000@ 5060;transport=udp SIP/2.0.
Via: SIP/2.0/UDP .:2051;branch=z9hG4bK-1ilh24yfah89;rport.
From: "From Work with Love" <sip:107@sip com="">;tag=t61qxsf4jf.</sip:107@sip>
To: <sip:5000@sip com;user="phone">;tag=apyFUyrtQcZ9j.</sip:5000@sip>
Call-ID: 5333f1ffd238-71xl4jk51vfn. Tor Go
CSeq: 3 BYE. USSLIND (https://github.com/stubbornella/csslint) for CSS
Max-Forwards: 70
Contact: <sip:107@: :2051;line="h5oh6sor">;reg-id=1.</sip:107@:>
User-Agent: snom360/8.7.3.25.
RTP-RxStat: Total Rx Pkts=316,Rx Pkts=0,Rx Pkts Lost=0,Remote Rx Pkts Lost=0.
RTP-TxStat: Total Tx Pkts=415,Tx Pkts=415,Remote Tx Pkts=0.
Content-Length: 0. Lighting - you don't need to move your eyes from the code to see the violations.
12 13. ## Installation
-3+ ## Installation
U 2014/03/27 10:40:25.302154 :5060 -> :2051
SIP/2.0 200 OK. Install storilint
<pre>via: SIP/2.0/UDP :2051;branch=z9hG4bK-1ilh24yfah89;rport=2051.</pre>
From: "From Work with Love" <sip:107@sipcom>;tag=i61qxsf4jf.</sip:107@sipcom>
To: <sip:5000@sip. com;user="phone">;tag=apyFUyrtQcZ9j.</sip:5000@sip.>
Call-ID: 5333f1ffd238-71xi4jk51vfn.
CSeq: 3 BYE. Propose file change
User-Agent: service.
Allow: INVITE, ACK, BYE, CANCEL, OPTIONS, MESSAGE, INFO, REGISTER, REFER, NOTIFY.
Supported: timer, precondition, path, replaces.
Content-Length: 0

ΗĘρ

### SIPGREP<sub>2</sub> CLI Usage and Features

More Handy Examples:

# Kill friendly-scanner automatically
sipgrep -J

# Kill friendLy-scanner with custom UAC name
sipgrep -j sipvicious

# Collect all Calls/Registrations dialogs during 120 seconds, print reports and exit: sipgrep -g -G -q 'duration:120'

# Split pcap\_dump to 20 KB files in sipgrep\_INDEX\_YYYYMMDDHHMM. pcap sipgrep -Q 'filesize:20' -0 sipgrep.pcap

# Split pcap\_dump in sipgrep\_INDEX\_YYYYMMDDHHMM.pcap each 120
seconds
sipgrep -Q 'duration:120' -0 sipgrep.pcap

Sipgrep packages are available natively on Debian SID:

https://packages.debian.org/sid/sipgrep

11 2014/03/27 10:40:25 2080	9 :2051 -> :5060
	5060;transport=udp SIP/2.0.
	<pre>:2051;branch=z9hG4bK-1ilh24yfah89;rport.</pre>
	<pre>com&gt;;tag=i61qxsf4jf.</pre>
	com;user=phone>;tag=apyFUyrt0cZ9j.
Call-ID: 5333f1ffd238-71xi	
	github.com/stubbornella/csslint) for CSS
Max-Forwards: 70.	
	:2051;line=h5oh6sor>;reg-id=1.
User-Agent: snom360/8.7.3.	
	316,Rx_Pkts=0,Rx_Pkts_Lost=0,Remote_Rx_Pkts_Lost=0.
	415,Tx_Pkts=415,Remote_Tx_Pkts=0.
	- you don't need to move your eyes from the code to see the violations.
Clean UI - Lt hond	
12	
## Installation	
U 2014/03/27 10:40:25.3021	54 :5060 -> :2051
SIP/2.0 200 OK.	
Via: SIP/2.0/UDP	:2051;branch=z9hG4bK-1ilh24yfah89;rport=2051.
	<pre>" <sip:107@sipcom>;tag=i61qxsf4jf.</sip:107@sipcom></pre>
	com;user=phone>;tag=apyFUyrt0cZ9j.
Call-ID: 5333f1ffd238-71xi	
CSeq: 3 BYE. Propose file ch	
User-Agent: servic	e.
Allow: INVITE, ACK, BYE, C	ANCEL, OPTIONS, MESSAGE, INFO, REGISTER, REFER, NOTIFY.
Supported: timer, precondi	tion, path, replaces.
Content-Length: 0.	



ΗĘρ

# SIPCAPTUR

SNGREP by Kaian/irontec Troubleshooting SIP sessions in the terminal... HEP included!

**sngrep** is a great tool for displaying SIP calls message flows from a terminal, exporting HEP3 packets to a HOMER instance and great for watching traffic over multiple local views:

- Call List Window: Allows to select the calls to be displayed •
- Call Flow Window: Shows a diagram of source and destiny of messages
- Call Raw Window: Display SIP messages texts
- Message Diff Window: Displays differences between two SIP messages

Display SIP packets from a PCAP file using filters

# sngrep -I file.pcap host 192.168.1.1 and port 5060

Display Live packets, save to a new PCAP file

# sngrep -d eth0 -O save.pcap port 5060 and udp

Export HEP3 Encapsulated packets to HOMER (eep.send)

# sngrep -H -d eth0 port 5060 and udp

SNGREP: https://github.com/irontec/sngrep

		call flow for -	1NVLTE sip: @10.10.1.245:5063 SIP/2.0
	142:5061 10.10.9	40:5060 10.10.1.122:5065 10.10	0.1.245:5063 Via: 51P/2.6/U0P 10.10.9.40:5660;branch=20h64bK16bb22a8 Max-Forwards: 70
5:01:30.395894			From: "Kailan * «sip:3607810.10.0.40»; tagmas/d7eb267 To:scipii #0.1.245.56842
5:01:30.396222			To: ≪sip::010, 10, 1, 245;5063* Contact:csip::0607010.10, 9, 40:50600 Coll. L0: 7.2050.165207698.0129726755010, 10, 9, 40:5060
			CSeq: 102 INVITE User-Agent: "Ironte: IV02"
			Date: Wed, 07 Jan 2015 Ja:01:30 GMT Allow: Invite, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIDE, NOTEFY, INFO, PUBLISH
			Supported: replaces, there X-CD: #340551-94106bfbg10.10.1.142
		INVITE (SOP)	N=100 = Party IO: "Main or sign: 2007(10.10.9.40+; party=calling; privacy=off; screen=no Content-Type: application/sdp
			Content-Length: 200
			v=0 0=r001 853892039 853892039 IN IP4 10.10.9.40
			s=hsterisk PER 1.8.15.1 (RSP CS branch 1.8.15.1) c=Nv [P4 10.10.9.46
			t=0 0 muudio 12700 RTP/AVP 0
			a+rtpmap:8 PCM/2000 arptmsc20
		ADX	
		200 OK	
Calls List Ent	er Rin Messade Space Company	The Help is SDP mode is Toggle Naw in Extended	S Compressed to New 17 Colour by 171 New width







SIP Troubleshooting

# **CENTRALIZED SOLUTIONS** Capture Servers & Long-Term Storage



# **Centralized Capture Systems**

Voice Packets echoing from the Past!

*Centralized Capture Systems* are generally designed for voice network operators, providers and ITSPs in need of permanent monitoring and troubleshooting resources for their Voice and Customer support and engineering teams on a daily basis and provides key features to protect and maximize voice products and accurately measure infrastructure or peering investments.

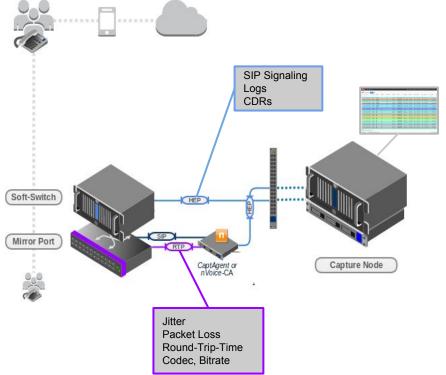
Several commercial and a few free options are available on the market covering this key role, each focusing on different areas but sharing some common advantages:

Key Benefits:

- system/platform agnostic capture viewpoint
- permanent monitoring of service resources
- instant troubleshooting present and past events
- long-term storage of signaling and usage metrics

User Benefits:

- accelerate access to aggregated information
- reduce initial investigation complexity
- reduce unsecured user access to key resources
- empower teamwork in case handling





SIP Troubleshooting

# **CENTRALIZED SOLUTIONS** HOMER + SIPCAPTURE





### Proudly Introducing

# HOMER 5







# SIPCAPTURE New to our Projects?

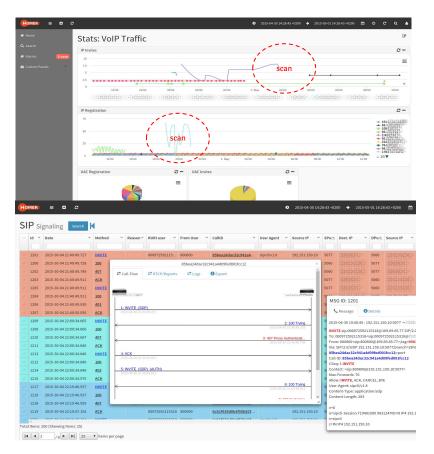
**SIPCAPTURE** is a powerful suite of tools enabling Voice Engineers to focus on their actual job without having to spend hours figuring how to get the data they need to work with on each instance. Our flagship product **HOMER** is a self-contained SIP Analysis and Troubleshooting environment fully customizable based on the preferences of its users:

H

**HOMER** is a turnkey solution providing many advantages:

- Instant centralized access to present and past signaling & stats
- Full SIP/SDP payload with precise timestamping
- Automatic correlation of sessions and reports
- Visual representation of multi session call-flows
- Fast detection of usage and system anomalies
- System agnostic view of VoIP traffic flows
- Unlimited plug & play capture agents and HEP data feeds
- Easy data integration with other systems via API
- No Desktop/Mobile client software required
- Ease of installation (no more 1st setup headaches!)

HOMER: http://github.com/sipcapture/homer





### HOMER 5 What's New in Homer 5 UI?

**HOMER 5** brings many core improvements and module extensions to handle so much more than just signaling, and delivers a complete overhaul of the web User-Interface component migrating to modern JS framework while retaining the simplicity and style many users worldwide rely upon daily.

#### HOMER's Main Features:

- 100% HTML5 & API Based User Interface
- No Defaults! All Pages and Dashboards fully customizable
- Multiple DB options (MySQL/MariaDB, PSQL, ElasticS, InfluxDB ... )
- Modern & Extensible Angular Drag & Drop UI
- User Customizable Widgets for Charts & Analytics
- Powerful SIP Search and Filtering functionality
- Native Canvas Call-Flow display with multi-session correlation
- Native support for PCAP and Text file export of all results
- Supports token Authentication for API and User Interface
- Multi-User support with Local, LDAP, Radius options
- Production Ready, supports high volumes and PPS rates
- Supported by a strong and growing community

P Si	gnaling 🐱	arch 🔀							2013-04-13-10:33		<b>y</b> 2013/04	-23 20.35	1:52 +0200 🛗			1
Id 👻	Date ~	Nicro TS ~	Method	Y Reason	RURIuser Y	From Use	r ~ CalliD ~	User Agent 🗸	Source IP ~	SPo.Y.	Dest. IP 🗸 🗸	DPo.Y.	Source IP 🗸 🗸	Pr.X	Node ~	
81864	2015-04-25 18:3	1429987	OPTIONS		147	mod_sofia	a <u>498cce78-eb7a-</u>	Botauro service	109.69.65.77	5050	212.202.252.157	38768	109.69.65.77	1	homer01:2001	4
81863	2015-04-25 18:3	1429987	OPTIONS		109	mod_sofia	a <u>498ccfd6-eb7a-</u>	Botauro service	109.69.65.77	5050	94.221.137.2	5060	109.69.65.77	1	homer01:2001	1
81865	2015-04-25 18:3	1429987	OPTIONS		121	mod_soft	a <u>498cd15c-eb7a</u>	Botauro service	109.69.65.77	5050	2.93.183.63	5060	109.69.65.77	1	homer01:2003	1
81866	2015-04-25 18:3	1429987	OPTIONS		104	mod_sofia	a <u>498cd2c4-eb7a</u>	Botauro service	109.69.65.77	5050	212.202.252.157	1024	109.69.65.77	1	homer01:2003	1
81867	2015-04-25 18:3	1429987	OPTIONS		204	mod_sofia	a <u>498cd44a-eb7a</u>	Botauro service	109.69.65.77	5050	78.94.165.3	19829	109.69.65.77	1	homer01:2003	i
81868	2015-04-25 18:3	1429987	OPTIONS		145	mod_sofia	a <u>498cd59e-eb7a</u>	Botauro service	109.69.65.77	5050	92.205.75.178	36166	109.69.65.77	1	homer01:2003	a
81869	2015-04-25 18:3	1429987	OPTIONS		101	mod_sofia	a <u>498cd6ac-eb7a</u>	Botauro service	109.69.65.77	5060	92.205.75.178	5799	109.69.65.77	1	homer01:2003	a
81870	2015-04-25 18:3	1429987	200	OK		mod_sofia	a <u>498cd44a-eb7a</u>	ACN IRIS X 1.0	78.94.165.3	19829	109.69.65.77	5060	78.94.165.3	1	homer01:2003	i
81872	2015-04-25 18:3	1429987	200	ок		mod_sofia	a 498cc8a6-eb7a	FRITZIOS	78.94.161.7	5050	109.69.65.77	5060	78.94.161.7	1	homer01:2001	4
81871	2015-04-25 18:3	1429987	200	ок		mod_sofia	a <u>498cca22-eb7a-</u>	FRITZIOS	78.94.165.3	5050	109.69.65.77	5060	78.94.165.3	1	homer01:2003	a
81873	2015-04-25 18:3	1429987	200	ок		mod_sofia	a <u>498cd59e-eb7a</u>	ACN IRIS X 1.0	92.205.75.178	36166	109.69.65.77	5060	92.205.75.178	1	homer01:2003	1
81874	2015-04-25 18:3	1429987	200	ок		mod_sofia	a 498ccfd6-eb7a	FRITZIOS	94.221.137.2	5060	109.69.65.77	5060	94.221.137.2	1	homer01:2003	1
81875	2015-04-25 18:3	1429987	200	ок		mod_sofia	a <u>498cc568-eb7a-</u>	AVM FRITZIB	212.90.61.174	1024	109.69.65.77	5060	212.90.61.174	1	homer01:2003	1
31876	2015-04-25 18:3	1429987	200	ок		mod_sofia	a 498ccba8-eb7a	FRITZIOS	93.130.200.200	5050	109.69.65.77	5060	93.130.200.200	1	homer01:2003	1
81877	2015-04-25 18:3	1429987	200	ок		mod_sofia	a <u>498cc70c-eb7a-</u>	AVM FRITZIB	151.249.219.93	5050	109.69.65.77	5060	151.249.219.93	1	homer01:2003	1
81878	2015-04-25 18:3	1429987	200	ак		mod_sofia	a <u>498cd2c4-eb7a</u>	Polycom/VX	212.202.252.157	1024	109.69.65.77	5060	212.202.252.157	1	homer01:2003	1
81879	2015-04-25 18:3	1429987	200	ок		mod_sofia	a <u>498cce78-eb7a-</u>	ACN IRIS X 1.0	212.202.252.157	38768	109.69.65.77	5060	212.202.252.157	1	homer01:2001	1
81880	2015-04-25 18:3	1429987	200	OK		mod_sofia	a <u>498cc3ec-eb7a-</u>	AVM FRITZIB	5.56.106.199	5050	109.69.65.77	5060	5.56.105.199	1	homer01:2001	4
01051	2015-04-25 18:3	1429987	200	ак		mod_sofia	a <u>498cd15c-eb7a</u>		2.93.183.63	5050	109.69.65.77	5060	2.93.183.63	1	homer01:2001	1
		1422987	200	OK		mod_sofia	a 498cc28e-eb7a	AVM FRITZIB	93.74.61.198	5050	109.69.65.77	5060	93.74.61.198	1	homer01:2001	a,
11852	2015-04-25 18:3															
terns: 2	2015-04-25 18:3 200 (Showing Items:	1429987 1429987 25)	259 ems per page	OK.		mod_sofia		snom360/8.7	212.202.252.157	-	109.69.65.77 100 cm cf 37 2015-04	5060 5060 25 19:12	212.202.252.157		homer01:2001	)1 
81883	2015-04-25 18:3 200 (Showing Items:	1429967 25) 25 • it	ems per page	OK AV		mod_sofi	a <u>428ccd2e-eb7a</u>			-	100 20 20 37				76 - 200 of 200 it C Q	il i
B1BB3 (Items: 2 B OMER forme	2015-04-25 18:3 200 (Showing Items:	1429967 25) 25 • HOI	ems per page	OK OY		mod_sofia	a <u>428ccd2e-eb7a</u>			-	100 20 20 37				76 - 200 of 200 it C Q	1 
tems: 2	2015-04-25 18:3 200 (Showing Items:	1429967 25) 25 • it HOI Quic	rems per page me k Search	OK OK		mod_sofia	a 428ccd2e-sb7a		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	1 
tems: 2	2013-04-25 18:3 200 (Showing Items: ) = 0 (Snew	25) 25 • HOI Quic	ems per page me k Search From	04		mod_sofii	a 428ccd2e-sb7a		2015-04-25 17:12	-	2015-04				76 - 200 of 200 it C Q	ter
tems: 2 ems:	2013-04-25 18:3 2016 14 14 14:1 200 (Showing Items: 7.5 P. H ) = • (Snew Panels <	1429967 25) 25 • it HOI Quic	ems per page me k Search From	04		mod_sofia	a 428ccd2e-sb7a		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	ter
tems: 2 tems: 2 B B Come earch Jarma Home Search	2015-04-25 18:3 200 (Showing Items: ) /s         ) = Snew Pasets <	25) 25 • II 25 • II Quic Quic	ems per page me k Search From				a <u>attaccize-atra</u>		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	ter
tems: 2 tems: 2 tem	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	25) 25 • II 25 • II Quic Quic	ems per page me k Search From				a 428ccd2e-sb7a		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	1 
tems: 2 tems: 2 tem	2015-04-25 18:3 200 (Showing Items: ) /s         ) = Snew Pasets <	25) 25 • II 25 • II Quic Quic	ems per page me k Search From	ок ли			a <u>attaccize-atra</u>		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	ter
tems: 2 tems: 2 tem	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	25) 25 • II 25 • II Quic Quic	ems per page me k Search from Call-ID	0			a <u>statecile sitra</u> Area Chart 7 6 5		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	ter
tems: 2 tems: 2 tem	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	1429987 25) 25 ↓ 1 25 ↓ 1 20 HOI Quic Quic Quic	ems per page me k Search from Call+D	0K 74		mod_safi	A REACTANT		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	1 
tems: 2 a 8 berns: 2 berns: 2 a 8 berns: 2 berns:	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	1429987 25) 25 • iti Quice	ems per page me k Search from Call-ID	08		-	Area Chart		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	1 
tems: 2 a 8 berns: 2 berns: 2 a 8 berns: 2 berns:	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	1429987 25) 25 • iti Quice	ems per page me k Search from Coll-ID coll-ID coll-ID coll-Coll-ID coll-Coll-Coll-Coll-Coll-Coll-Coll-Coll-	04			A REACTANT		2015-04-25 17:12	14+020	2015-04				76 - 200 of 200 it C Q	1 
tems: 2 a 8 berns: 2 berns: 2 a 8 berns: 2 berns:	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	1429987 25) 25 • iti Quice	ems per page me k Search from Coll+D Coll+D Coll+D Coll+D Coll+C	DK CV			Area Chart		2015-04-25 17:12	14+0200	hart		34 +0200		76 - 200 of 200 % C Q C -	ter
tems: 2 tems: 2 tem	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	225) 25 • 1 Quic Quic Quic Links	ems per page me k Search from Coll+D Coll+D Coll+D Coll+D Coll+C	04			Area Chart	•	2015-04-29 1712 	14+0200	hart 22 May 20	25 19:12	34 +0200		76 - 200 of 200 it C Q	ter
tems: 2 tems: 2 tem	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	225) 25 • 1 Quic Quic Quic Links	ems per page me k Search from Coll+D Coll+D Coll+D Coll+D Coll+C	OK SE			Area Chart	•	2015-04-29 1712 	14+0200 krea C	hart 22 May 20	25 19:12	34 +0200		76 - 200 of 200 % C Q C -	ter
tems: 2 a 8 berns: 2 berns: 2 a 8 berns: 2 berns:	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	225) 25 • 1 Quic Quic Quic Links	ems per page me k Search from Coll+D Coll+D Coll+D Coll+D Coll+C	er NewYork (US) Tempsture:		- -	Area Chart	•	2015-04-29 1712 	14+0200 krea C	hart 22 May 20	25 19:12	34 +0200		76 - 200 of 200 % C Q C -	ter
tems: 2 tems: 2 tem	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	225) 25 • 1 Quic Quic Quic Links	ems per page me k Search from Coll+D Coll+D Coll+D Coll+D Coll+C	er New York (US)		- -	Area Chart	•	2015-04-29 1712 	14+0200 krea C	hart 22 May 20	25 19:12	34 +0200		76 - 200 of 200 % C Q C -	ter
tems: 2 tems: 2 tem	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	225) 25 • 1 Quic Quic Quic Links	ems per page me k Search from Coll+D Coll+D Coll+D Coll+D Coll+C	er NewYork (US) Tempsture:		- -	Area Chartt		2015-04-29 1712 	14+0200 krea C	hart 22 May 20	25 19:12	34 +0200		76 - 200 of 200 % C Q C -	ter
ILBEST ILEENS: 2 ILEENS: 2 ILEENS: 2 ILEENS: 1 ILEENS: 1 Stats: 1	2015-04-25 18:3 200 (Showing Items: ) / 6 P P ) = • (S new Panels <	25) 25 • n P P P P P P P P P P P P P	ems per page me k Search from Coll+D Coll+D Coll+D Coll+D Coll+C	er NewYork (US) Tempsture:	)	- -	Area Chart	o 2. kw	2015-04-2 1712 A	14+0200 krea C	hart 22 May 20	25 19:12	34 +0200		76 - 200 of 200 % C Q C -	ter

#### HOMER: <a href="http://github.com/sipcapture/homer">http://github.com/sipcapture/homer</a>



# HOMER 5: How does it work?

Build your own HOMER Capture Server using SIPCAPTURE modules

HOMER setups requires two basic building blocks:



#### **CAPTURE SERVER**

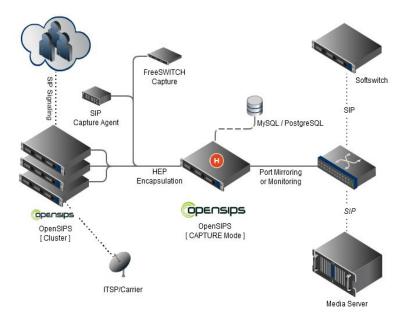
A Capture Server Collects, Indexes and Stores SIP packets received from Capture Agents using HEP v1/2/3, SBCs using IPIP or Raw SIP from Ethernet interfaces and mirrored switch ports, using flexible rules defined in the powerful, extensible and fully customizable capture plan

• Requires:OpenSIPS + <u>sipcapture</u> module

### 

A Capture Agent sends encapsulated packets to a Capture Server using the *HEP* Encapsulation protocol designed for **HOMER** 

The Capture Agent role can be covered by different elements running on different platforms or architectures and distributed in a completely modular fashion, allowing it to support any network topology and complexity and to easily scale with the monitored architectures, as displayed in the illustration on the right.





#### HOMER Capture Server: Capture Plan, Alarms and Statistics configuration

SIPCAPTURE module logic for packet capture, alarms and statistics is completely customizable and extensible with no limits:

```
####### Packet Capture Logic ########
        if(is method("INVITE|BYE|CANCEL|UPDATE|ACK|PRACK|REFER"))
               $var(table) = "sip capture call";
        else if(is method("REGISTER"))
               $var(table) = "sip capture registration";
        else if(is method("INFO"))
               $var(table) = "sip capture call";
        else if(is method("OPTIONS"))
               $var(table) = "sip capture rest";
        else {
               $var(table) = "sip capture rest";
        $var(a) = $var(table) + " %Y%m%d";
        sip capture("$var(a)");
```

More Examples: https://github.com/sipcapture

```
if (is method("INVITE|REGISTER")) {
   if($ua =~ "(friendly-scanner|sipvicious)") {
        sql query("cb", "INSERT INTO alarm data mem (create date, type,
total, source ip, description) VALUES(NOW(), 'scanner', 1, '$si', 'Friendly
scanner alarm!') ON DUPLICATE KEY UPDATE total=total+1");
        route(KILL VICIOUS);
    #TP Method
        sql query("cb", "INSERT INTO stats ip mem ( method, source ip, total)
VALUES('$rm', '$si', 1) ON DUPLICATE KEY UPDATE total=total+1");
    if($au != $null) $var(anumber) = $au;
    else $var(anumber) = $fU;
    #hostname in contact
    if($sel(contact.uri.host) =~ "^(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})$") {
            if($sht(a=>alarm::dns) == $null) $sht(a=>alarm::dns) = 0;
            $sht(a=>alarm::dns) = $sht(a=>alarm::dns) + 1;
```



# HOMER 5: What the HEP is HEP? HEP = Homer Encapsulation Protocol

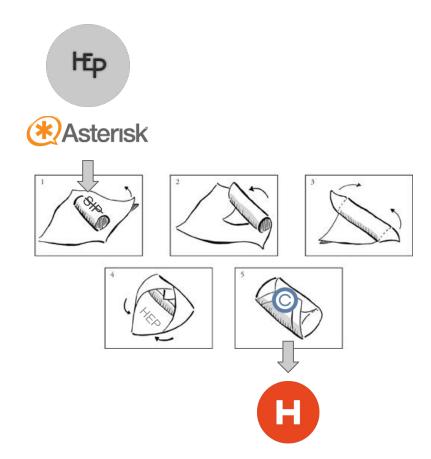
**HOMER**'s own Encapsulation protocol (*HEP/EEP*) is used to wrap and transfer captured packets between a capture Agent and Server.

The HEP 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 and payload MUST be provided to the collector without alterations or data loss.

The HEP3/EEP definition includes both generic *(internal)* and vendorspecific custom defined chunk types providing ground for implementors to extend the spectrum of the deliverable data within the HEP protocol alongside the encapsulated IP datagram.

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

Find the full specs at: http://github.com/sipcapture/hep





# HOMER 5

# **SIP Search Application**





# HOMER 5

### Your new SIP Search Dashboard is ready to use!

	c	2015-05-21 08	8:58:01 +0200 🗲 2015-05-21 09:58:01 +0200 🛗 O C Q i
🖨 Home	SIP Search Custom Form F	ields	
<b>Q</b> SIP Search	Session Parameters	- Session Heade	ers Search Time Range –
🛢 Dangerous Demo	€ RURI	Q User-Age	nt
Alarms	€ From	Q Method	
🚳 Custom Panels 🛛 🤇	СТО	Q CSeq	
» Alarms » Custom	Call-ID	Q Reason	
<ul><li>» Home</li><li>» SIP Search</li></ul>		Clear Search Q Message	
<ul> <li>Stats: IP Network</li> <li>SIP Search</li> </ul>	Network Parameters	Q Diversion	
» Dangerous Demo	□ Source IP	Search Control Parran	notore _
» System Admin	Source Port	Parran	neters —
<ul> <li>» ES Aggregations Test</li> <li>» Stats: VoIP Traffic</li> </ul>	Q Dest. IP	Q Transacti	CALLS *
	Dest. Port	<b>Q</b> Limit Que	ry
		Q Result Ty	pe TABLE •
		Q DB Node	homer01 *



## HOMER 5 Let's find some SIP traffic next!

	°	1 2015-05-21 08:58:01 +0200 → 2015-05-21 09:58:01 +0200 🛗 Ø C Q 🌢
者 Home	SIP Search	
<b>Q</b> SIP Search	Session Parameters -	
🛢 Dangerous Demo	€ RURI	Quick Search:
Alarms	¢From (2)	1) Select Time Range
🚯 Custom Panels 🛛 <	¢T0	2) Filter any SIP Header
» Custom	Call-ID	3) Choose Transaction Type
	A Search	1) Coorobl
» SIP Search	4 Search	4) Search!
<ul> <li>Stats: IP Network</li> <li>SIP Search</li> </ul>	Network Parameters	-
» Dangerous Demo	므 Source IP	
- » System Admin		Search Parrameters -
» ES Aggregations Test	□ Source Port	CALLS
» Stats: VoIP Traffic	Q Dest. IP	Q Transaction REGISTRATIONS OTHER
	🖵 Dest. Port	Q Limit Query
		Q Result Type TABLE
		Q DB Node external



# HOMER 5 Example: Search Results

**1** Find the session of interest

H	OMER	≡ ⊕ ≎	Search	n Result I	Filtering			Ð	2015-05-12 12:09:	50 +0200	→ 2015-05-	12 13:09:	50 +0200 🛛 🛗	Ø	cα	4
S	IP s	ignaling Search	•	Ţ			Session Call-ID									
~	Id	Date Y	Method ~	Reason ~	RURI user 🛛 🗡	From User 💙	CallID ~	User Agent 🛛 🗡	Source IP 🛛 👻	SPo.X.	Dest. IP 💙	DPo.X.	Source IP	Pr.X.	Node 🗡	=
	-														-	
~	1465	2015-12-05 12:24:45.682	INVITE		00972597562	14	<u>b5675fb30a513329f16</u>	sipcli/v1.8	85.52121111	5108	109 55 55 77	5060	85.52127111	1	homer01:200	1
	1470	2015-12-05 12:24:45.925	INVITE		00972597562	14	b5675fb30a513329f16	sipcli/v1.8	85.521217111	5108	109 55 577	5060	85.5217711	1	homer01:200	1
~	1474	2015-12-05 12:24:46.429	INVITE		00097259756	14	d65e33aeb15d8a5ff9e	sipcli/v1.8	85.22222221111	5093	109 53 55 77	5060	85.22121211	1	homer01:200	1
~	1478	2015-12-05 12:24:46.620	INVITE		00097259756	14	d65e33aeb15d8a5ff9e	sipcli/v1.8	85.2222222222	5093	109 99 99 97 77	5060	85.33177711	1	homer01:200	1
~	1482	2015-12-05 12:24:47.380	INVITE		90097259756	14	1bb1b5f0caf8cac6a3e	sipcli/v1.8	85.52121211	5089	109555577	5060	85.32123211	1	homer01:200	1
~	1486	2015-12-05 12:24:47.709	INVITE		90097259756	14	1bb1b5f0caf8cac6a3e	sipcli/v1.8	85.52121111	5089	109525577	5060	85.3212121	1	homer01:200	1
~	1490	2015-12-05 12:29:33.830	INVITE		90097259262	2001	d5962b9c0461478857	sipcli/v1.8	19954346668	5070	109929277	5060	1955234256	1	homer01:200	1
~	1503	2015-12-05 12:38:16.216	INVITE		107	101	777650246@10 0 0 200	S450 IP/0222	92.2555.24278	5799	109985577	5060	92.3255.3447	1	homer01:200	1
~	1507	2015-12-05 12:38:16.409	INVITE		107	101	777650246@10 0 0 200	S450 IP/0222	92.2255.27478	5799	109525577	5060	92.1205.2143	1	homer01:200	1
	1508	2015-12-05 12:38:16.433	INVITE		107	101	d6ebb56e-7335-1233	Botauro service	1099355777	5060	212 22 22 22 22 22 22 22 22 22 22 22 22	2048	10998655777	1	homer01:200	1
	1525	2015-12-05 12:50:35.422	INVITE		00972597562	111	9ed46e190a1bc3641d	sipcli/v1.8	85.5.21717111	5093	1099365777	5060	85.32121211	1	homer01:200	1
	1529	2015-12-05 12:50:35.540	INVITE		00972597562	111	9ed46e190a1bc3641d	sipcli/v1.8	85.3217111	5093	109955577	5060	85.32117.111	1	homer01:200	1
	1533	2015-12-05 12:50:36.955	INVITE		00097259756	111		sipcli/v1.8	25.217.111 85.5.217.111	5083	109 9 6 5 5 7 7	5060	85.5217.111	1	homer01:200	1
	1537	2015-12-05 12:50:37.044	INVITE		00097259756			sipcli/v1.8	85.521717111	5083	109525577	5060	85.32121711	1	homer01:200	
	1541	2015-12-05 12:50:38.300	INVITE		90097259756			sipcli/v1.8	85.32121211	5078	109 9 6 9 6 9 7 7	5060	85.32121211	1	homer01:200	
	1541	2015-12-05 12:50:38.407	INVITE		90097259756			sipcli/v1.8	85.5217.111	5078	109 55 55 77	5060	85.323217.111	1	homer01:200	
														1		
	1549	2015-12-05 12:51:33.507	INVITE		00972592621	6000	fb3a61ee1c643850bd	sipcli/v1.8	19554540606	5071	1099,65,69,7	5060	1955434056)	1 1	homer01:200	T.



# HOMER 5 Example: Session Details

2 Click a Call-ID to correlate a Session

						• 2015-05-12 12:09:50 +0200 → 2015-05-12 13:09:50 +0200
SIP si	gnaling Search	C				Session Details
✓ Id ×	Date ~	Method Y Reason	RURI user 💙	From User Y	CallID ~	User Agent <sup>×</sup> Source IP <sup>×</sup> SPo. <sup>×</sup> . Dest <sup>×</sup> DPo. <sup>×</sup> . Source IP <sup>×</sup> Pr. <sup>×</sup> . Node <sup>×</sup> ≡
1465 1470	2015-12-05 12:24:45.682 2015-12-05 12:24:45.925		00972597562 00972597562	14	b5675fb30a513329f16	b5675fb30a513329f1600477b4c71b5e         2001           #* Call-Flow         #* RTCP/Reports         Export
1474 1478 1482	2015-12-05 12:24:46.429 2015-12-05 12:24:46.620 2015-12-05 12:24:47.380	INVITE INVITE	Call-Flo	w & Correla	d65e33aeb15d8a5ff9e ation	2001 2001 152571111:5108 110159.775.5060 2001
1486 1490	2015-12-05 12:24:47.709 2015-12-05 12:29:33.830 2015-12-05 12:38:16.216	INVITE INVITE	90097259756 90097259262 107		1bb1b5f0caf8cac6a3e           d5962b9c0461478857           777650246@10 0 0 200	1: INVITE (SDP) 2015-05-12 12:24:45.682 :2001
<ul><li>1507</li><li>1508</li></ul>	2015-12-05 12:38:16.409 2015-12-05 12:38:16.433	INVITE INVITE	107 107	101 101	777650246@10 0 0 200 d6ebb56e-7335-1233	C 2: 100 Trying
1525 1529 1533	2015-12-05 12:50:35.422 2015-12-05 12:50:35.540 2015-12-05 12:50:36.955	INVITE INVITE INVITE	00972597562 00972597562 00097259756	111	9ed46e190a1bc3641d 9ed46e190a1bc3641d 420013b69f4c6e6aae4	2013-03-12 12:24:45:925 2001 2015-05-12 12:24:45:925 2001
<ul><li>✓ 1537</li><li>✓ 1541</li></ul>	2015-12-05 12:50:37.044 2015-12-05 12:50:38.300	INVITE	00097259756 90097259756		420013b69f4c6e6aae4 355bcbd16282ff6c7ce	5: INVITE (SDP) (AUTH) 2015-05-12 12:24:45.925 :2001
1545           1549	2015-12-05 12:50:38.407           2015-12-05 12:51:33.507	INVITE INVITE	90097259756 00972592621		355bcbd16282ff6c7ce fb3a61ee1c643850bd	€ 100 Trying 2015-05-12 12:24:45.925 2001 € 2001 € 2001 2015-05-12 12:24:45.945



### HOMER 5 Example: Session and Packet Details

(3) Click & Inspect any SIP Message

C	ом	ERI	≡	9 C						ø	2015-05-12 12:09:50	+0200 🔶	2015-05-	12 13:09:50	+0200	Ê	0	c	۹	4
S	IF	Sig	gnaling	Search																
	I	d ~	Date	~	Method ~	✓ Reason ✓	RURI user 🛛 👻	From User	✓ CallID ✓	User Agent 🛛 👻	Source IP Y S	Po.:. Dest	. IP Ý	DPo.X. S	ource IP	~	Pr.X. N	lode	~	=
						SIP M	essage D	etails												
	1	465		5 12:24:45.682	INVITE				b5675fb30a513329f16		b5675	5fb30a51332	9f1600477b	b4c71b5e				×	:2001	
	1	470		5 12:24:45.925	INVITE		009 97562		b5675fb30a513329f16	🛱 Call-Flow	RTCP/Report	s 📫 Lo	gs 🚯 B	Export					:2001	
	1	.474		5 12:24:46.429			00097259756	14	d65e33aeb15d8a5ff9e	d.									:2001	
	1	.478	2015-12-05	MSG ID: 146	7				^ <u></u>	-									:2001	
	1	.482	2015-12-05	🍾 Message	Oetails				•••	85.25.2							5.77:50	060	:2001	
	1	486	2015-12-05	2015-05-12 10:2	4:45:0109.65.75	1:5060 -> 5,25	25,217,15,5108			1: INVITE (	(SDP)							_	:2001	
	1	.490	2015-12-05	SIP/2.0 407 Prox					**	2015-05-12 1	2:24:45.682							1	:2001	
2	1	.503	2015-12-05	14. 51 /2.0/001					.00	·							0 Trying		:2001	
	1	.507	2015-12-05	b5675fb30a513 From: 14 <sip:14< th=""><th>32911600477b40 4@01.09.65.65.7</th><th></th><th>108</th><th></th><th>.00</th><th></th><th></th><th></th><th></th><th></th><th>2015-05</th><th>5-12 12:24</th><th></th><th></th><th>:2001</th><th></th></sip:14<>	32911600477b40 4@01.09.65.65.7		108		.00						2015-05	5-12 12:24			:2001	
	1	.508	2015-12-05	10.0001200100			的形式的标志;tag	=0ZvUp654vNUQ	Q	←				3	: 407 Prox	y Auther		4	:2001	
	1	.525	2015-12-05	Call-ID: b5675fb CSeq: 1 INVITE	030a513329f160	0477b4c71b5e				11.754					and door for to				:2001	
	1	.529	2015-12-05	User-Agent: Bot						4: ACK 2015-05-12 1	2:24:45.925							>	:2001	
	1	.533	2015-12-05	Accept: applicat Allow: INVITE, A		., OPTIONS, ME	SSAGE, INFO, R	EGISTER, REFER,	NOTIFY										:2001	
	1	.537	2015-12-05	Supported: time	er, precondition,	path, replaces				5: INVITE ( 2015-05-12 1	(SDP) (AUTH) 12:24:45.925						3	>	:2001	
	1	.541	2015-12-05	Allow-Events: ta Proxy-Authentic			2", nonce="1c5.	1f6a8-f891-11e4-	9f7e-4958111f9453",							0.10	O Tadar		:2001	
	1	.545	2015-12-05	algorithm=MD5	, qop="auth"	100 60 66 7				K					2015-05	6: 100 5-12 12:24	0 Trying 4:45.925		:2001	
	1	.549	2015-12-05	Content-Length	:: 0										7.	: 403 Fo	rhidden		:2001	
										K						5-12 12:24		1		



### HOMER 5 Example: Session and Packet Details

**4** Click & Inspect RTCP-XR Reports

Номе	ERI	≡ (							٥	2015-05-12 12:09:50 +020	00 🔶 2015-0	5-12 13:09	:50 +0200 🛛 🎬	0	c	۹ 🛔
SIP	Sig	naling	Search	•						RTCP-XR Qo	s					
V Id		Date	~	Method ~	Reason ~	RURI user 🛛 🗡	From User 💙	CallID	V User Ag	Reports	·	✓ DPo.ĭ.	Source IP	Y Pr.∷	Node	~ =
14			12:24:45.682	INVITE		00972597562		<u>b5675fb30a513329f16</u>		b5675fb3	0a513329f160047	7b4c71b5e	3		×	:2001
14	470	2015-12-05	12:24:45.925	INVITE		00972597562	14	b5675fb30a513329f16	Call-Flow	RTCP/Reports	at Logs	3 Export				2001
- 14			12:24:46.429	INVITE		00097259756	14	d65e33aeb15d8a5ff9e.	•							:2001
- 14	478	2015-12-05	MSG ID: 146	7				×				=	Total	packets		2001
- 14	482	2015-12-05	📞 Message	O Details					150				10	56		2001
- 14	486	2015-12-05		د سه نده نده بده ازد. مو	5050 · 50				100		$\wedge$		10	50		2001
- 14	490	2015-12-05		4:45:0109.65.65.		943,44449,0108		-	ate				packets lost	max packe		2001
× 15	503	2015-12-05		xy Authentication 9 85.25.217.111:51	100 C	z9hG4bK-		.0	0 50		$\langle \rangle$		2.5%	3.0	%	2001
× 15	507	2015-12-05		329f1600477b4c7				.0	0							2001
- 15	508	2015-12-05		4@01.09.65.07.7>;t 2926 <sip:0097259< td=""><th></th><td>0 付加付加付加付/&gt;;tag=</td><td>07vUp654vNUOO</td><td>e</td><td>16:15:15</td><td>16:15:20 16:15:25 1</td><td>6:15:30 16:15:35</td><td>-</td><td>М</td><td>OS</td><td></td><td>2001</td></sip:0097259<>		0 付加付加付加付/>;tag=	07vUp654vNUOO	e	16:15:15	16:15:20 16:15:25 1	6:15:30 16:15:35	-	М	OS		2001
- 15	525	2015-12-05		30a513329f16004			21000001110000			Date				Go	bod	2001
- 15	529	2015-12-05	CSeq: 1 INVITE User-Agent: Bot	auro convico							Highchau	rts. com			,ou	2001
~ 15		2015-12-05	Accept: applicat							From:			A	()		2001
		2015-12-05				IESSAGE, INFO, RE	GISTER, REFER, N	IOTIFY	Beginning 5/9/2015	End 0522223 5/9/2015	78@82.000 0008 SI	P/2.0	Avg. jitter			2001
		2015-12-05		er, precondition, p alk, hold, conferen		25			12:35:00 /	12:42:57 To: 0522233	28@82.880\$ 0008 SI	P/2.0	25			2001
			Proxy-Authentic	ate: Digest realm		∰", nonce="1c51	f6a8-f891-11e4-9f	7e-4958111f9453",	5/9/2015 <b>12:42</b>	0.57			Max jitter	(ms)		
		2015-12-05	algorithm=MD5, Content-Length						> 5/9/2015 12:42				٨٢			2001
- 15	549	2015-12-05	content-tength						5/9/2015 <b>12:45</b>	::54			40			2001
									5/9/2015 <b>12:48</b>	3:35						



# HOMER 5

## Got Charts?





### HOMER 5 Create a Stats Dashboard in seconds

	1	<ul> <li>2015</li> </ul>	5-05-21 08:58:01 +0200 → 2015-05-21 09:58:01 +0200 🛗 O C Q 🔺
A Home Chart Ty	vpe Preferences	×	SIPCapture Charts ×
Q SIP Search	रा		
📰 Dangerous Demo	Chart Type 3 Fields	Queries 3 Debug	Chart Type S Fields Queries Debug
Alarms		Ext Legend: 🕑	API Query Path
🚳 Custom Panels 🔍	Registration	Align: right Chart Query Fields	statistic/ip API Query Value
	h: ter width (0) - default	SIPCapture Charts	t "timestamp": {
» Custom Heigh			"from": "@from_ts", "to": "@to_ts"
» SIP Search	D	Chart Type S Fields Queries Debug	}, "param": { "filter": [
<ul> <li>Stats: IP Network</li> <li>SIP Search</li> </ul>	Bar 🔻	API Query Fields Timefield:	{"method": "REGISTER"}
» Dangerous Demo 3D:		to_ts	"limit": 200, "total": false
» System Admin Stack	ked: 🖉 % 🗌	Fieldname (single or semicolumn separated):	}
<ul> <li>» ES Aggregations Test</li> <li>» Stats: VoIP Traffic</li> </ul>		source_ip	ok
		FieldValue (single or semicolumn separated):	
		total	
		FieldValue sum: 🔲	
			Query Details
			Close



... Stats: IP Network Elasticsearch Histograms C -Elasticsearch Top IP SRC C -SIPCapture Packet Rate C -200 200 500 Ξ Ξ  $\equiv$ 150 150 400 Bytes 100 100 age 300 ALL Mes 50 200 50 0 1921681.25 192,189,1.24 192.168.1.23 1270.01 80:16552.·· 1921681.17 1921681.16 80:13:Hoch. 23.97.148.3 -80.88<sup>3,28C.\*\*</sup> 100 0 22:00 Sunday, May 10, 00:10 nprobe: 168 0 nprobe 22:30 23:00 23:30 22:00 10. May InfluxDB Server Load C -Elasticsearch L7 Proto Aggs SIPCapture Method Distrib.. C -C -0.8 100 100 Ξ Ξ 75 0.6 75 50 0.4 50 Saturday, May 9, 23:24:40 25 0.2 homer.5MinAvg: 0.03 25 0 0 22:00 22:30 23:00 23:30 10. May 0 10. May 23:00 23:30 LINNIR Reals 550° UNICOM HITP Droppot 22:00 22:30 Skipe ONS 554 MONS 200 REGISTER 407 OPTIONS INVITE

BYE

homer.1MinAvg

homer.5MinAvg

homer.15MinAvg

SIPCAPTURE



### HOMER Capture Server using OpenSIPS: QoS Reports and Logging

```
# PUBLISH REPORT
```

```
if(is_method("PUBLISH") && has_body("application/vq-rtcpxr"))
```

```
$var(table) = "report_capture";
$var(callid) = $(rb{re.subst,/(.*)CallID:([0-9A-Za-z@-]{5,120})(.*)$/\2/s});
```

```
$var(temp) = $(rb{re.subst,/^(.*)JitterBuffer:(.*)JBN=([0-9]{1,5})(.*)$/\3/s});
if(float2int("$var(temp)", 1)) $var(jbn) = $rc;
```

```
#Mos
```

```
$var(temp) = $(rb{re.subst,/^(.*)QualityEst:(.*)MOSCQ=([0-9.]{1,4})(.*)$/\3/s});
if(float2int("$var(temp)", 10)) $var(mos) = $rc;
```

```
statsd_set($var(customer)+"Mos", $var(mos));
statsd_set($var(customer)+"JBN", $var(jbn));
```

```
#save to db
report_capture("$var(table)", "$var(callid)");
```

drop;

More Examples: https://github.com/sipcapture

**RTCP-XR** provides a range of VoIP call and network quality metrics generated by user agents and devices supporting the protocol. The reports can be very useful to debug the user quality of a given session and are supported by HOMER. RTCP-XR packets can be handled in two different ways by a capture agent:

 STORE Mode Using HEP proto\_id 99 QoS reports are sent to DB

#### - FORWARD Mode

Using HEP SIP proto\_id, QoS reports are forwarded to kamailio.cfg where users can parse and extract relevant information for statistical purposes and store to internal hashmap, Homer DB, or statsd module

#### HINT: Don't miss our QoS Dangerous Demo!

References:

RFC 3611 (RTP Control Protocol Extended Reports) RFC 6035 (SIP Package for Voice Quality Reporting)



### **OpenSIPS + flatstore recipe** On-demand, long-term archiving of SIP signaling

This configuration option instructs the sipcapture module to use the flatstore db module which is configured to create all of its files in the "/db/homer\_dat" directory - note such directory must exist and have write permissions for the process user!

modparam("sipcapture", "db\_url", "flatstore:/db/homer\_data")

Define sip\_capture table as:

```
$var(table) = "sip_capture_%Y%m%d%H%M.flat"
```

and each hour we start bzip2 inside this table and move to special directory:

find /db/homer\_data -type f -name "\*.flat" -exec "movefile.sh" {} \;

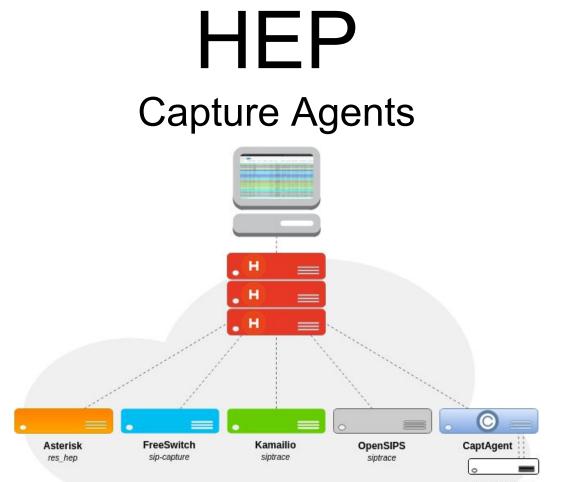
#!/bin/sh
FILE=\$1
bzip2 -kv9 \$FILE
mv \$(FILE).bz2
/db/homer\_bzip/

movefile.sh

Flatstore files can be restored to a local mysql DB if and when necessary.

A dedicated node connector can also be defined from Homer's UI and used for searches on demand.









### HEP - Homer Encapsulation Protocol Integrated Capture Agents in OSS Platforms

**HOMER**'s own encapsulation protocol (*HEP/EEP*) is used to transfer your packets unmodified and carries several key information in its headers designed for perfect capturing.

**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
- sipXecs

Examples are also provided for the following languages:

- C/C++
- Java
- Javascript / Node.JS
- Erlang
- Go

The *HEP/EEP* Protocol is defined in a mature Draft pending submission and is freely available for developers to integrate.

Find more about HEP: http://hep.sipcapture.org/

S
Other HEP Agents
OpenSIPS Example:
https://github.com/sipcapture/homer/wiki/Examples%3A-OpenSIPS
Kamailio Example:
<pre>https://github.com/sipcapture/homer/wiki/Examples%3A-Kamailio</pre>
FreeSWITCH Example:
https://github.com/sipcapture/homer/wiki/Examples%3A-FreeSwitch
CaptAgent Example:
<pre>https://github.com/sipcapture/homer/wiki/Examples%3A-Captagent4</pre>
nProbe VoIP Example:
https://github.com/sipcapture/homer/wiki/Examples%3A-nProbe
ACME SBC Example:
<pre>https://github.com/sipcapture/homer/wiki/Examples%3A-ACME-Packet</pre>



### CAPTAGENT 6 Modular Capture Agent w/ HEP3 Support

**Captagent** started as a SIP-only capture agent for HOMER. The codebase over time has been completely redesigned from the ground up to follow the evolution of the **HEP** protocol and **Captagent** grew to become 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:

- UNI Proto Module
  - SIP, XMPP and other text signaling Protocols
- RTCP Module
  - RTCP and RTCP-XR Parser and Collector
- CLI Module
  - CLI Shell Access and control of Captagent
- HEP Module
  - HEP Encapsulation output (v1/2/3)
- SSL/TLS Module
  - Encryption and Compression Module for HEP3

#### Upcoming modules:

- Remote API Module
  - Configure and Control a feet of Captagents from a Central server

```
CAPTAGENT: https://github.com/sipcapture/captagent
```

```
<!-- CORE MODULES -->
```

```
<configuration name="core_hep.conf" description="HEP Socket">
   <settings>
   <param name="version" value="3"/>
   <param name="capture-host" value="capture.server.org"/>
   <param name="capture-port" value="9060"/>
   <param name="capture-port" value="udp"/>
   <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>
</configuration>
```

<!-- PROTOCOLS -->

<configuration name="proto\_uni.conf" description="UNI Proto Basic capture">

```
<settings>
<settings>
(param name="port" value="5060"/>
<!-- (param name="portrange" value="5060-5090"/> -->
<!--
        use -D flag for pcap import
        use "any" for all interfaces in your system
--->
caram name="dev" value="eth0"/>
caram name="promise" value="true"/>
<!-- comment it if you want to see all IPProto (tcp/udp) -->
cparam name="proto-type" value="sip"/>
<!-- <pre>cparam name="filter" value="not src port 5099"/> -->
</settings>
</configuration>
```

# SIPCAPTURE

### Example: Captagent 6 programming

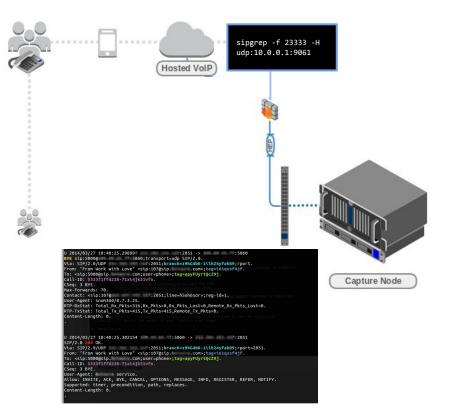
```
<module name="socket pcap" description="HEP Socket" serial="2014010402">
                                                                                    #sip capture plan.cfg
<profile name="socketspcap sip" description="HEP Socket" enable="true"</pre>
                                                                                    capture[pcap] {
serial="2014010402">
        <settings>
                                                                                       # here we can check source/destination IP/port, message size
           <param name="dev" value="any"/>
                                                                                       if(msg check("size", "100")) {
           <param name="promisc" value="true"/>
           <param name="reasm" value="false"/>
                                                                                           #Do parsing
           <param name="capture-plan" value="sip capture plan.cfg"/>
                                                                                           while(parse sip()) {
           <param name="filter">
                      <value>portrange 5060-5091</value>
                                                                                               /* many packets */
                                                                                               clog("NOTICE", "parsing SIP message ");
           </param>
       </settings>
</profile>
                                                                                               if(source ip("10.0.0.1")) {
<profile name="socketspcap rtcp" description="RTCP Socket" enable="true"</pre>
                                                                                                       #Can be defined many profiles in transport hep.xml
serial="2014010402">
                                                                                                        if(!send hep("hepsocket homer01")) {
                                                                                                             clog("ERROR", "Error sending HEP!!!!");
        <settings>
           <param name="dev" value="any"/>
           <param name="promisc" value="true"/>
                                                                                               }
           <param name="reasm" value="false"/>
                                                                                              else {
           <param name="capture-plan" value="rtcp capture plan.cfg"/>
                                                                                                        #Can be defined many profiles in transport hep.xml
           <param name="filter">
                                                                                                        if(!send hep("hepsocket homer02")) {
                      <value>portrange 30000-50000</value>
                                                                                                             clog("ERROR", "Error sending HEP!!!!");
           </param>
       </settings>
</profile>
                                                                                               #Duplicate all INVITEs to JSON transport
</module>
                                                                                               if(sip is method() && sip check("method","INVITE")) {
                                                                                                    #Can be defined many profiles in transport json.xml
                                                                                                    if(!send json("jsonsocket")) {
                                                                                                          clog("ERROR", "Error sending JSON");
         More Examples: https://github.com/sipcapture
                                                                                      drop:
```



### SIPGREP<sub>2</sub> Sipgrep as disposable HEP3 Agent

**Sipgrep** is able to act as a quick on-demand HEP3 capture agent and forward packets to a collector very easily when a simple terminal check does not suffice.

In the following example, Sipgrep is used to display the traffic of interest as well as log it to a remote location, useful for instance when troubleshooting issues on hosted platforms or disposable instances on the cloud.



#### HEP3 Example:

Display dialogs and duplicate all traffic to HOMER sipcapture in HEPv3:

sipgrep -f 23333 -H udp:10.0.0.1:9061



### NPROBE SIP Mirroring Capture & Mirror SIP Signaling using nProbe/nVoice SIP Plugin

NTOP **nProbe** (*w*/*VoiP PRO Plugin*) can act as *HEP3* capture agent for SIP Protocol mirroring to a centralized collector such as Homer and can perform this task at high packet rates. The HEP3 features are simply controlled by the following switches:

--hep <host>:<port> --hep-auth <capture id>:<password>

Send JSON flows via HEPv3 protocol
Specify the HEP authentication parameters.

Example HEP3 SIP Syntax:

# nprobe -T "%SIP\_CALL\_ID" --drop-flow-no-plugin -i eth0 hep <u>10.0.10.20:9063</u> --hep-auth 10:myhep123 -b 0 -G

NTOP nProbe SIP Plugin can also send out its SIP detections via JSON, NetFlow, or dump logs locally for server-less, ad-hoc implementations or simple batch processing:

--sip-dump-dir <dump dir> --sip-exec-cmd <cmd> | Directory where SIP logs will be dumped | Command executed whenever a directory has been dumped %SIP CALL ID %SIP CALLING PARTY %SIP CALLED PARTY %SIP RTP CODECS %SIP INVITE TIME %SIP\_TRYING\_TIME %SIP RINGING TIME %SIP INVITE OK TIME %SIP INVITE FAILURE TIME %SIP BYE TIME %SIP\_BYE\_OK\_TIME %SIP CANCEL TIME %SIP\_CANCEL\_OK\_TIME %SIP RTP IPV4 SRC ADDR %SIP\_RTP\_L4\_SRC\_PORT %SIP RTP IPV4 DST ADDR %SIP\_RTP\_L4\_DST\_PORT %SIP RESPONSE CODE %SIP REASON CAUSE %SIP C IP %SIP CALL STATE

SIP call-id SIP Call initiator SIP Called party SIP RTP codecs SIP time (epoch) of INVITE SIP time (epoch) of Trying SIP time (epoch) of RINGING SIP time (epoch) of INVITE OK SIP time (epoch) of INVITE FAILURE SIP time (epoch) of BYE SIP time (epoch) of BYE OK SIP time (epoch) of CANCEL SIP time (epoch) of CANCEL OK SIP RTP stream source IP SIP RTP stream source port SIP RTP stream dest IP SIP RTP stream dest port SIP failure response code SIP Cancel/Bye/Failure reason cause SIP C IP adresses SIP Call State

NPROBE VoIP: http://ntop.org



SIP Troubleshooting

# **MEDIA QUALITY STATISTICS** RTP & RTCP Analysis



### **RTP Statistics**

### Network and Media quality probing using RTP, RTCP, RTCP-XR, RTP Reports...

In order to capture, investigate and analyze media stream quality and network issue, we need to interact with the protocols involved with transmission, controlling and reporting of media streams - We should all all be familiar with the following:

#### **RTP (Real-time Transport Protocol)**

The aim of RTP is to provide a uniform means of transmitting data subject to real time constraints over IP (audio, video, etc.). The principal role of RTP is to implement the sequence numbers of IP packets to reform voice or video information even if the underlying network changes the order of the packets. More generally, RTP makes it possible to: identify the type of information carried, add temporary markers and sequence numbers to the information carried, monitor the packets' arrival at the destination. RTP works over UDP and its header carries synchronization and numbering information such as sequence number, timestamp and unique identifier for the source.

#### **RTCP (Real-time Transport Control Protocol)**

RTCP is a protocol associated with RTP based on periodic transmissions of control packets by all participants in the session and used provide different types of information and a return regarding the quality of reception.

#### RTCP-XR (Real-time Transport Control Protocol Extended Reports)

Extended Report (XR) packet type for the RTP Control Protocol (RTCP) are used to convey information beyond what is already contained in the reception report blocks of RTCP sender report (SR) or Receiver Report (RR) packets, such as internal statistics about the stream quality and network conditions encapsulated in various types of SIP PUBLISH/OPTIONS reports sent by enabled endpoints during and after a call session.

#### X-RTP-Stats, P-RTP-Stat (User Agent generated End of Call Statistics)

The Reporting of End-of-Call QoS Statistics in Session Initiation Protocol (SIP) BYE Message feature enabled user-agents to send quality statistics to a remote end when a call terminates. The call statistics are sent as a new header included in the BYE message or in the 200 OK response, and include Real-time Transport Protocol (RTP) packets sent or received, total bytes sent or received, total number of packets that are lost, delay jitter, round-trip delay, call duration and more, providing the endpoint view over the call performance.



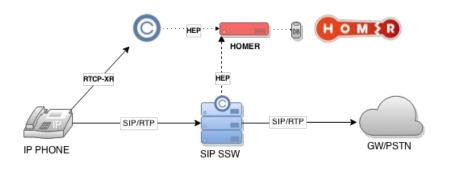
### RTCP-XR Statistics CaptAgent as RTCP-XR Collector or Reporter

#### How can we use RTCP-XR to troubleshoot call quality?

CaptAgent 6 features a powerful RTCP-XR collector module.

*RTCP-XR* enabled User-Agents (*Snom, Cisco, Polycom, etc*) can directly use **captagent** as a quality report collector. The dedicated module will forward an HEP encapsulated RTCP-XR report to your capture server (such as Homer or PCapture) for later analysis and correlation with the call sessions they belong with and indexed for general statistical purposes.

**Captagent** can also collect raw RTCP packets and send them as HEP3 or JSON/RAW format to a capture server and can also optionally generate and transmit final RTCP-XR reports *(CallTerm)* including RTP statistics generated for the call duration including Jitter, Delay, Packet Loss and so on, performing an RTCP-> RTCP-XR format adaption/conversion



#### PUBLISH SIP/2.0

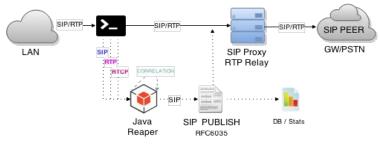
From: <sip:446@intern.snom.de>;tag=45hkui59ns
To: <intern.snom.de>;tag=nohhk4xu21
Call-ID: 3c26a8de500f-12ct7zov3kjs
CSeq: 3 PUBLISH
Max-Forwards: 70
Contact: <sip:446@192.168.5.251:2060;transport=tls;line=w2wuvhk9>;reg-id=1
Event: vq-rtcpxr
Accept: application/sdp, message/sipfrag
Content-Type: application/vq-rtcpxr
Content-Length: 832

VOSessionReport LocalMetrics: Timestamps:START=2010-02-17T13:59:42Z STOP=2010-02-17T13:59:46Z SessionDesc:PT=0 PD=G.711U PPS=50 SSUP=off CallID: 3c26a8de500f-12ct7zov3kis x-UserAgent:snom360/8.2.sf FromID:<sip:446@intern.snom.de> ToID:<sip:447@intern.snom.de;user=phone> LocalAddr: IP=192.168.5.251 PORT=62754 SSRC=0xCBE3450E RemoteAddr: IP=192.168.0.233 PORT=54018 SSRC=0xB80B52F3 DialogID:3c26a8de500f-12ct7zov3kjs;to-tag=866ed0cf03;from-tag=45hkui59ns x-SIPmetrics:SVA=RG SRD=310 SFC=0 x-STPterm:SDC=OK litterBuffer: 184=0 188=0 18N=0 18M=0 18X=65535 PacketLoss:NLR=0.0 JDR=0.0 BurstGapLoss:BLD=0.0 BD=0 GLD=0.0 GD=6569 GMIN=16 Delay:RTD=0 ESD=0 IAJ=4 RemoteMetrics: JitterBuffer: JBA=0 JBR=0 JBN=0 JBM=0 JBX=0 PacketLoss:NLR=0.0 JDR=0.0 BurstGapLoss: BLD=0.0 BD=0 GLD=0.0 GD=4677 GMIN=16 Delav:RTD=0 ESD=0 IAJ=2



### RTP Statistics SIP Voice Quality Report **Reaper** (java)

The **Reaper** is a java tool is designed to sniff **SIP/RTP/RTCP** packets (using a modified tcpdump agent pipe) and generate correlated voice quality reports in accordance with *RFC6035* forwarding the media stream statistics into the SIP signaling flow for post-processing.



RTCP Reports are processed as forwarded as received:

RTCP

 $\rightarrow$  <u>VQIntervalReport</u>  $\rightarrow$  <u>SIP PUBLISH</u>

RTP Final Statistics are released once the call is Terminated:  $\star$  RTP  $\rightarrow$  VOSessionReport  $\rightarrow$  SIP PUBLISH

In order to work the Reaper depends on a modified tcpdump binary forwarding packets to special queues feeding the Java process. This makes this solution only suitable for small, custom setups.

REAPER Github: <u>https://github.com/TerryHowe/SIP-Voice-Quality-Report-Reaper</u> RFC6035: <u>https://tools.ietf.org/html/rfc6035</u> PUBLISH sip:collector@127.0.0.1:5999;transport=udp SIP/2.0. Call-ID: f1f90855d85e9c874a0dd8e3b14bc607@127.0.0.2. CSeq: 1 PUBLISH. From: "reaper" <sip:reaper@127.0.0.2:5070>;tag=ReaperV1.0. To: "collector" <sip:collector@127.0.0.1:5999>. Via: SIP/2.0/UDP 127.0.0.2:5070;branch=reaperv1.0f1f90855d85e9c874a0dd8e3b14bc607-127.0.0.2-1-publish-127.0.0.2-5070333031. Max-Forwards: 70. Contact: "reaper" <sip:reaper@127.0.0.2:5070>. Content-Type: application/vg-rtcpxr. Content-Length: 451. VOSessionReport : CallTerm. LocalMetrics:. SessionDesc:PT=8 PD=PCMA SR=8000. Timestamps:START=2015-02-28T21:04:31.000582Z STOP=2015-02-28T21:04:36.000638Z. CallID:1233727184. FromID:<sip:caller@domain.net>. ToID:<sip:callee@domain.net>. OrigID:<sip:caller@domain.net>. LocalAddr: IP: 192.168.1.23 PORT: 7079. LocalMAC:99:72:b9:28:c2:82. RemoteAddr: IP:192.168.1.55 PORT: 30539. RemoteMAC:99:e6:ba:df:7b:dd. PacketLoss:NLR=4.6. Delay:IAJ=166.



### **RTCP Statistics** Asterisk RTCP Statistics

The latest **Asterisk** patch developed by Alexandr Dubovikov and Matt Jordan implements module <u>res\_hep\_rtcp</u>

The module performs RTCP packet capturing for the internal RTP engine in Asterisk and transmits HEP3 encapsulated call quality metrics & statistics in HEP encapsulated JSON format.

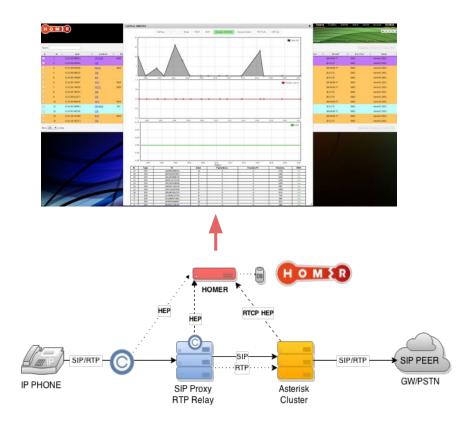
The module can be coupled with <u>res\_hep</u> to build a full HEP capture node and send SIP signaling as well as call QoS.

With the above setup, statistics can be observed historically and in real time as they reach the server when observing a call including pseudo-MOS score calculated on the client-side.

Example HOMER integration is presented on the side slide:

For more information and patch details:

https://github.com/sipcapture/homer/tree/master/asterisk\_rtcp\_patch





### **RTP Statistics** SIP User Agent: End-of-Call Reports

The Reporting End-of-Call Statistics in Session Initiation Protocol (SIP) BYE Message feature enables user-agents to send call statistics to a remote end when a call itself terminates. The call statistics are sent as a new header in the BYE message or in the 200 OK message (*response to BYE message*).

The statistics include Real-time Transport Protocol (RTP) packets sent or received, total bytes sent or received, total number of packets that are lost, delay jitter, round-trip delay, and call duration.

Commonly implemented SIP headers are X-RTP-Stat and P-RTP-Stats and the less complex RTP-RxStat / RTP-TxStat

X-RTP-Stat: PS=207;OS=49680;;PR=314;OR=50240;PL=0;JI=600;LA=40;	<pre>P-RTP-Stat: PS=326,0S=52160,PR=318,0R=50880,PL=0,JI=0,LA=0,DU=7, EN=G711a,DE=G711a</pre>
The X-RTP-Stat header contains the following fields:	The P-RTP-Stat header contains the following fields:
PS= <voice packets="" sent=""></voice>	PS= <packets sent=""></packets>
OS= <voice octets="" sent=""></voice>	OS= <octets sent=""></octets>
PR= <voice packets="" received=""></voice>	PR= <packets recd=""></packets>
OR= <voice octets="" received=""></voice>	OR= <octets recd=""></octets>
PL= <receive loss="" packet=""></receive>	PL= <packets lost=""></packets>
JI= <jitter in="" ms=""></jitter>	JI= <jitter></jitter>
LA= <latency in="" ms=""></latency>	LA= <round delay="" in="" ms="" trip=""></round>
	DU= <call duration="" in="" seconds=""></call>
	EN= <audio encoder=""></audio>
<pre>Specs: <u>https://www.avm.de/de/Extern/files/x-rtp/xrtpv32.pdf</u></pre>	DE= <audio decoder=""></audio>



### **RTP Statistics** RTPProxy Statistics injection into P-RTP-Stat Header

Although RTP Statistics are to be generated by the UA/client in order to be fully meaningful, **RTPProxy** can still provide back its own *internal rtp statistics (as seen by the relay)* to be included in *BYE / 200 OK* messages using the data sent back to the SIP Proxy core by RTPProxy module, and formatted in a **P-RTP-Stat** compatible header.

Additional information can be injected into the header from database queries or other local or external sources.

A pseudo basic example script extension could look as follows:

```
## Pseudo P-RTP-Stats snippet for RTPProxy

if (is_method("BYE")) {
    setflag(FLT_ACC); # do accounting ...
    setflag(FLT_ACCFAILED); # ... even if the transaction fails

    $var(xrtpstat) = $(rtpstat{s.striptail,1});

    # Work the new stats
    $var(rtp0) = $(var(xrtpstat){s.select,1, });
    $var(rtp1) = $(var(xrtpstat){s.select,2, });
    $var(rtp2) = $(var(xrtpstat){s.select,3, });
    $var(rtp3) = $(var(xrtpstat){s.select,4, });
    $var(rtp4) = $(var(xrtpstat){s.select,5, });
    if ($var(rtp0) != "" || $var(rtp1) != "")
    {
        append_hf("P-RTP-Stat: EX=RTPProxy,PS=$var(rtp0),PR=$var(rtp1),PL=$var(rtp3)\r\n");
        }
}
```



### RTP Statistics at Wire-Speed nProbe RTP Plugin w/ Pseudo-MOS Estimation

NTOP **nProbe** (*w*/ *VoIP RTP Plugin*) can produce granular RTP Statistics for network streams detected via nDPI and is able perform full SIP session report bi-directional correlation and codec aware Pseudo-MOS/R-Factor estimations, all exportable at user defined sample rates via JSON over TCP or HTTP/S to a centralized collector.

#### Example RTP Plugin Syntax:

# nprobe -T "%IPV4\_SRC\_ADDR %L4\_SRC\_PORT %IPV4\_DST\_ADDR %L4\_DST\_PORT %PROTOCOL %
RTP\_IN\_JITTER %RTP\_OUT\_JITTER %RTP\_IN\_PKT\_LOST %RTP\_OUT\_PKT\_LOST %
RTP\_IN\_PAYLOAD\_TYPE %RTP\_OUT\_PAYLOAD\_TYPE %SIP\_CALL\_STATE %RTP\_SIP\_CALL\_ID %
SIP\_CALL\_ID %RTP\_RTT %RTP\_MOS %RTP\_R\_FACTOR %IN\_PKTS %OUT\_PKTS %RTP\_IN\_TRANSIT %
RTP\_OUT\_TRANSIT %RTP\_RTT" --redis 127.0.0.1 --drop-flow-no-plugin -i eth1 -b 3 -json-Labels -t 30 --hep <u>10.0.10.20:9063</u>--hep-auth 10:myhep123 -b 0 -G

#### Example RTP Statistics:

{"FIRST\_SWITCHED":1411119211,"IPV4\_SRC\_ADDR":"1.2.2.222","L4\_SRC\_PORT":11034," IPV4\_DST\_ADDR":"1.1.1.233","L4\_DST\_PORT":37308,"PROTOCOL":17,"RTP\_IN\_JITTER":2391," RTP\_OUT\_JITTER":475,"RTP\_IN\_PKT\_LOST":1,"RTP\_OUT\_PKT\_LOST":0,"RTP\_IN\_PAYLOAD\_TYPE": 18,"RTP\_OUT\_PAYLOAD\_TYPE":18,"RTP\_SIP\_CALL\_ID":"h8A02kd73jdc","IN\_PKTS":729," OUT\_PKTS":240,"IN\_BYTES":43740,"OUT\_BYTES":24000,"RTP\_IN\_TRANSIT":1135," RTP\_OUT\_TRANSIT":11,"RTP\_RTT":0,"L7\_PROTO\_NAME":"RTP","RTP\_MOS":435,"RTP\_R\_FACTOR": 9033,"TOTAL\_FLOWS\_EXP":19731}

NPROBE VoIP: http://ntop.org

%RTP FIRST SSRC %RTP FIRST TS %RTP LAST SSRC %RTP LAST TS %RTP IN JITTER %RTP OUT JITTER %RTP IN PKT LOST %RTP OUT PKT LOST %RTP IN PAYLOAD TYPE %RTP OUT PAYLOAD TYPE %RTP IN MAX DELTA %RTP OUT MAX DELTA %RTP SIP CALL ID %RTP MOS %RTP R FACTOR %RTP\_IN\_TRANSIT %RTP OUT TRANSIT %RTP RTT %RTP DTMF TONES %SIP RTP CODECS %SIP RTP IPV4 SRC ADDR %SIP RTP L4 SRC PORT %SIP\_RTP\_IPV4\_DST\_ADDR %SIP RTP L4 DST PORT

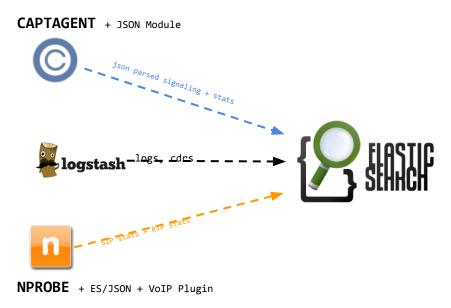
First flow RTP Sync Source ID First flow RTP timestamp Last flow RTP Sync Source ID Last flow RTP timestamp RTP jitter (ms \* 1000) RTP jitter (ms \* 1000) Packet lost in stream (src->dst) Packet lost in stream (dst->src) RTP payload type RTP payload type Max delta (ms\*100) between pkts (src->dst) Max delta (ms\*100) between pkts (dst->src) SIP call-id corresponding to this RTP stream RTP pseudo-MOS (value \* 100) RTP pseudo-R FACTOR (value \* 100) RTP Transit (value \* 100) (src->dst) RTP Transit (value \* 100) (dst->src) RTP Round Trip Time (ms) DTMF tones sent (if any) during the call SIP RTP codecs SIP RTP stream source IP SIP RTP stream source port SIP RTP stream dest IP SIP RTP stream dest port



### Voice CDRs & LOGS Elasticsearch + CaptAgent / nVoice

Already collecting metrics in Elasticsearch or any other JSON-centric backend? *Good News!* You can integrate your voice statistics to your existing data infrastructure with very little work with minimal technical efforts and investment.

CDRs and System logs can now be aggregated with their network counterpart adding a further dimension to your data.



RProbe - VolF	FIONS	Page: nProbe - V	oir riows	• 61	iours ago to a few	seconus ago reir	concurevely bill			8	8	۰
EVENTS OVER TIME	* (8156) count per 5m   (8156 hits)									0 3	0 1	ŀ )
	and a state of the	toout								m		
0 11:30 10-22	12:00 12:30 10:22 10:22	13:00 10-22		14:00	14:30 10-22	15:00	15:30 10-22	16:00 10-22	16:30		17:00	
10-22	10-22 10-22	10-22	10-22	10-22	10-22	10-22	10-22	10-22	10-22		10-22	
FLOWS			0 2 0 4	× :	SIP FLOWS					0 2	• +	. ,
								<sip:gst.+o.r< td=""><td>@am: /qxi</td><td>o.net;use</td><td>er=phone</td><td>e&gt; 🛛</td></sip:gst.+o.r<>	@am: /qxi	o.net;use	er=phone	e> 🛛
								<sip:la< td=""><td>b@c54qxi</td><td>o.net;use</td><td>er=phone</td><td>e&gt; [</td></sip:la<>	b@c54qxi	o.net;use	er=phone	e> [
								"(sticam"	<sip:o-brear< td=""><td>1@61112</td><td>.qvip.ne</td><td>et&gt;</td></sip:o-brear<>	1@61112	.qvip.ne	et>
			87.2					<siptina c<="" na="" td=""><td>@an</td><td>o.net;use</td><td>r=phone</td><td>e&gt;</td></siptina>	@an	o.net;use	r=phone	e>
188.226.27			01.4	£.,				<sip:< td=""><td>CE: 3 9@ et</td><td><ol> <li>2.qoip.</li> </ol></td><td>net:5060</td><td>0&gt;</td></sip:<>	CE: 3 9@ et	<ol> <li>2.qoip.</li> </ol>	net:5060	0>
								5/391145 <sip< td=""><td></td><td></td><td>-</td><td></td></sip<>			-	
								0689302 <sip< td=""><td>:5631.107@</td><td>sip.mess</td><td>agenet.i</td><td>it&gt; 🗌</td></sip<>	:5631.107@	sip.mess	agenet.i	it> 🗌
					<singstar 10<="" td=""><td>siam.gdp.net;use</td><td>r=nhone&gt;</td><td><sin 999<="" td=""><td>9@qxi</td><td>netuse</td><td>ranhone</td><td>e&gt;</td></sin></td></singstar>	siam.gdp.net;use	r=nhone>	<sin 999<="" td=""><td>9@qxi</td><td>netuse</td><td>ranhone</td><td>e&gt;</td></sin>	9@qxi	netuse	ranhone	e>
			87.210 2 2.1:		- opiger - ng	in an opposite of	n priorite		and the standard			
			62.1#0 \$30 \$	-10	<sip:883510%< td=""><td>0.11001@.jw.q</td><td>xip.net&gt;</td><td><sip:*98< td=""><td>799999@sip</td><td>broker.c</td><td>om:5060</td><td>0&gt;</td></sip:*98<></td></sip:883510%<>	0.11001@.jw.q	xip.net>	<sip:*98< td=""><td>799999@sip</td><td>broker.c</td><td>om:5060</td><td>0&gt;</td></sip:*98<>	799999@sip	broker.c	om:5060	0>
			81 200 42 1 62.540 532 7									
			62. 46 1 10 1 88.1 10 1 10 10	10-11	"1014" <sip:10< td=""><td>00@pbx.sipforeve</td><td>er.com&gt; 441524</td><td>230030<sip:441< td=""><td>5242300300</td><td>y 199.22</td><td>0.207.00</td><td>&gt;</td></sip:441<></td></sip:10<>	00@pbx.sipforeve	er.com> 441524	230030 <sip:441< td=""><td>5242300300</td><td>y 199.22</td><td>0.207.00</td><td>&gt;</td></sip:441<>	5242300300	y 199.22	0.207.00	>
			88 12 68 11	8-1			100441524230	030 <sip:100441< td=""><td>524230030</td><td>9 200.00</td><td>6.207.00</td><td>5&gt;</td></sip:100441<>	524230030	9 200.00	6.207.00	5>
			62. 700 MB 10 46. MB 700 M	ii - ii	1001 <sin:1001< td=""><td>@188.226.157.5</td><td>0004415242200</td><td>20crin-0000441</td><td>524220020</td><td>R tank and</td><td></td><td>-</td></sin:1001<>	@188.226.157.5	0004415242200	20crin-0000441	524220020	R tank and		-
			212 204 204		1001 000.2003	.e 100.110.101.01	60004415242500	30<3ip.0000441	324230030	4100.00	0.207.00	. L
			173. 21				+00441524230	030 <sip:+00441< td=""><td>524230030</td><td>2188.23</td><td>6, 157, 55</td><td>&gt;</td></sip:+00441<>	524230030	2188.23	6, 157, 55	>
LL EVENTS										0 2	0 4	ŀ,
				(	to 50 of 250 availa	ble for paging			÷			
ields O	@timestamp •	SIP_CALL_ID .		< SIP	CALLING_PARTY		SIP_CALLED_R	ARTY		SIP_C	ALL_STA	ATE
I (133) / Current (41)	2014-10-	2089370599-5064-49@BJ	C.BGI.B.BBA	<sip.c< td=""><td>aste vrr Ør' mussipur</td><td>etuser=phone&amp;g</td><td><sip:9999(81-11).c< td=""><td>pip.net;user=phoni</td><td>&gt;</td><td>CALL C</td><td>OMPLET</td><td>ED</td></sip:9999(81-11).c<></td></sip.c<>	aste vrr Ør' mussipur	etuser=phone&g	<sip:9999(81-11).c< td=""><td>pip.net;user=phoni</td><td>&gt;</td><td>CALL C</td><td>OMPLET</td><td>ED</td></sip:9999(81-11).c<>	pip.net;user=phoni	>	CALL C	OMPLET	ED
Type to filter	22716:40:24.292+02:00											
	2014-10-	2089370599-5064-49@BJ	C.BGI.B.BBA	<sip.8< td=""><td>835101012440018</td><td><pre>clow.quip.net&gt;</pre></td><td><sip.99999(8:1; *<="" td=""><td>pip.net;user=phone</td><td>2&gt;</td><td>CALL_C</td><td>OMPLET</td><td>ED</td></sip.99999(8:1;></td></sip.8<>	835101012440018	<pre>clow.quip.net&gt;</pre>	<sip.99999(8:1; *<="" td=""><td>pip.net;user=phone</td><td>2&gt;</td><td>CALL_C</td><td>OMPLET</td><td>ED</td></sip.99999(8:1;>	pip.net;user=phone	2>	CALL_C	OMPLET	ED
8 Otmestamp	22T16:40:24.293+02:00											



### Voice CDRs & LOGS Experiment with **HEPipe**

Troubleshooting is not all about network packets - many times system logs will hold valuable pointers at internal issues not expressed at the protocol level. There are many tools able to forward syslog/rsyslog to notorious collectors but for those looking to build their own voice data collection, we have developed a HEP3 playground utility called **HEPipe** 

**HEPipe** (pronounced HEP-pipe) is an application for logging arbitrary data (*ie: logs, cdrs, debug lines*)to a HEP/EEP capture server such as HOMER or PCAPTURE via command pipe.

The utility can be used to prototype HEP3 implementations as well as to feed real data into a HEP Collector for real life usage, for instance by using the session Call-ID as correlation parameter.

**INPUT FORMAT:** 

timestamp\_sec; timestamp\_usec; correlation\_id; source\_ip; source\_port; destination\_ip; destinaton\_port; payload in json

USAGE EXAMPLE:

echo '1396362930;1003;18731b65be;127.0.0.1;5060;10.0.0.1;5060;{"pl": 10, "jt": 10}'/./hepipe -s hepserver -p 9061 -t 100



SIP Troubleshooting

# AUTOMATED TESTS Friendly Probes



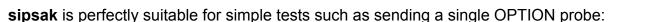
### SIP Testing with Scripted Agents PJSUA and SIPSAK

**pjsua** can be used as a simple call generator to test SIP Trunk or equipment availability:

# pjsua < (echo "sleep 2000;M;20;sip:192.168.1.10;sleep 10000;ha;sleep 5000;quit;")

pjsua can be launched in daemon mode and configured to act as a playback auto-responder:

# pjsua –null-audio –play-file=data3.wav –auto-play –auto-answer=200 –config-file=pj-config



# sipsak -vv -s sip:192.168.1.10:5060

sipsak can also send customer methods (NOTIFY Event: check-sync;reboot=true causing yealink phone to reboot):

# sipsak -f reboot\_yealink.sipfile -s sip:1234@192.168.1.10

**sipsak** is ideal for Nagios usage: <u>http://exchange.nagios.org/directory/Plugins/Network-Protocols/\*-VoIP/SIP/check\_sip-sipsak/details</u> (we use this ourselves since 2002 and still up)





### SIP Testing with quality-aware Agents BARESIP User-Agent w/ X-RTP-Stats

Baresip is a modular open-source (BSD) user agent built on top of LibRE/LibREM by Alfred E. Heggstad

One of our contributions to the project was the ability to export the valuable internal stream/codec details and statistics *(Jitter, Packet Loss, Payload details, etc)* by implementing <u>*X-RTP-Stat*</u> header export in <u>BYE/200 OK</u> SIP Messages.

This enables Baresip being used as a *"quality probing" SIP* user-agent (or echo-test agent) with call-quality results efficiently distributed alongside the session closure methods, featured in many existing brand Hardware SIP Phones.

Test Calls can be automatically scheduled (or triggered via HTTP Command API) and results collected by existing systems.

Header Example:

X-RTP-Stat: EX=BareSip;CS=0;CD=152;PR=7383;PS=7635;PL=0,0;PD=0,0;JI=0.8,0.1;EN=PCMU/8000;DE=PCMU/8000;IP=A.B.C.D:4202,E.F.G.H:29778;\*

BARESIP Git: <u>https://github.com/alfredh/baresip</u>

BARESIP Wiki: <u>https://github.com/alfredh/baresip/wiki</u>

XRTP Specs: <u>https://www.avm.de/de/Extern/files/x-rtp/xrtpv32.pdf</u>



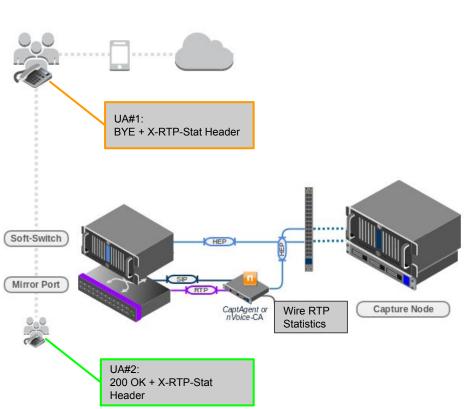
### SIP Testing with quality-aware Agents BARESIP User-Agent w/ X-RTP-Stats (continued)

**Baresip** agents can be deployed in tandem to validate call quality across specific SIP Paths.

In the following illustration:

- UA#1 Originates a session and steams prerecorded audio to UA#2
- UA#2 acting as an Echo-Test streaming all packets back to the UA#1 (auto-answer)
- Both Agents will publish Stream quality statistics on session termination as X-RTP-Stats

A Capture Server monitoring SIP Signaling (such as HOMER) will be able to extract and process the quality reports from SIP Headers and provide this additional insight for troubleshooting issues in investigations or for alarming on automated tests.





### SIP Testing with Scripted Agents SIPP Scenarios for Service Validation

**SIPp** is a free Open-Source test tool and traffic generator for the SIP protocol, able to read custom XML scenario files describing from very simple to complex call flows simulating both User-Agent Servers and Clients supporting optional media traffic through *RTP echo* and *RTP / PCAP replay*. While optimized for stress and performance testing, **SIPp** can be used to run one single call and exit, providing a passed/failed verdict *(Exit code 0: Test Successful, Exit code 1: Test with Failures)* and export its details and results to CSV files making it perfectly suitable for ad-hoc testing and able to be paired with other platforms/scripts.

А

**SIPp** scenarios are easy and fun to write and customize with many community collections ready to be used and extended for just about any purpose - Our favourite is kindly provided by Saghul on Github:

https://github.com/saghul/sipp-scenarios

Several old-school tools are available to convert PCAP traces to SIPp Scenarios:

- <u>http://sourceforge.net/projects/pcap2sipp/</u>
- http://frox25.no-ip.org/~mtve/wiki/Pcap2Sipp.html
- http://svn.digium.com/svn/sniff2sipp/trunk/sniff2sipp

**SIPp** also runs great on the *Raspberry-PI* and makes a fantastic pocket tool. good custom Pi-Tailored installer is maintained by Paul Miller on bitbucket:

# wget "http://bitbucket.org/idkpmiller/installation-scripts/raw/master/install\_sipp.sh"
# chmod +x install\_sipp.sh
# ./install\_sipp.sh

			[1-4]	: Change Screen
Call-rate(length) Port	Total-time 1	Total-calls	s Remote-	host
190 cps(0 ms) 5061	50.01 s	8586	5 127.0.0	.1:5060(UDP)
190 new calls during 1.000	s period 3	ms schedu	ler resol	ution
205 concurrent calls (limit	570) I	eak was 23	32 calls,	after 6 s
D out-of-call msg (discarde				
l open sockets				
		Retrans	Timeout	Unexpected-Msg
INVITE>	8586			
100 <				
180 <	8586			
200 < B-RT	D 8586			
ACK>	8586			
[ 1000 ms]				
BYE>	8381			



### Running too BIG for Homer and MySQL?

Meet PCAPTURE: The Extensible Capture Server and API

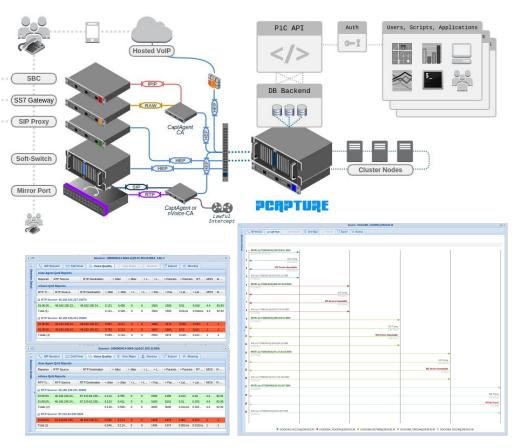
**PCAPTURE** is the commercial big-brother of *HOMER* and *SIPCAPTURE*, designed and crafted to provide a virtually infinite voice monitoring solution, leveraging the vast experience gathered assisting and developing solutions for some of the largest and busiest ITSPs,

Telecommunication Networks and Vendors of Voice Services and Equipment in the industry.

PCAPTURE provides many additional features:

- Real-Time Tracking and Monitoring of Sessions
- RTP/RTCP/PUBLISH QoS Reports, MOS/RFactor
- CDRs & Log Collectors with integrated parsing
- Automatic Correlation of Sessions legs, QoS, Logs
- Scalable, Multiple Distributed-Database layers
- Rich Multi-User User-Interface (HTML5/ExtJS)
- 1-Click Complete Session Details, Real-Time Usage
- Fully customizable Dashboard and Widgets
- Cross-Platform Capture Agents & Analyzers
- 100% REST API based & Integration Ready
- One Click-Troubleshooting for Tech and Non-Tech

#### Find out more: http://www.pcapture.com





### Install & Run a HOMER Capture Server & Capture Agent in a snap!

All SIPCAPTURE Projects are now available as packages supporting CentOS 6/7, Debian 7/8 and Ubuntu Server 14+

Pick an install script for your platform - it will detect your platform and architecture, install the gpg key that we use for repo signing and setup the repo accordingly. Once set, proceed to install the Homer bundle meta-packet for your OS distribution.

#### DEB:

BONUS

Debian 7/8, Ubuntu Server 14.04

curl -s https://packagecloud.io/install/repositories/qxip/homer/script.deb.sh | sudo bash

#### RPM:

CentOS 6/7, RHEL 6/7

curl -s https://packagecloud.io/install/repositories/qxip/homer/script.rpm.sh | sudo bash



### Any Questions?





### SIP Troubleshooting



### Time's UP! Want to go further?

Contact us to learn more about our advanced Capture and Troubleshooting Workshops <<u>training@qxip.net</u>>