



AMI

(Asterisk Manager Interface)

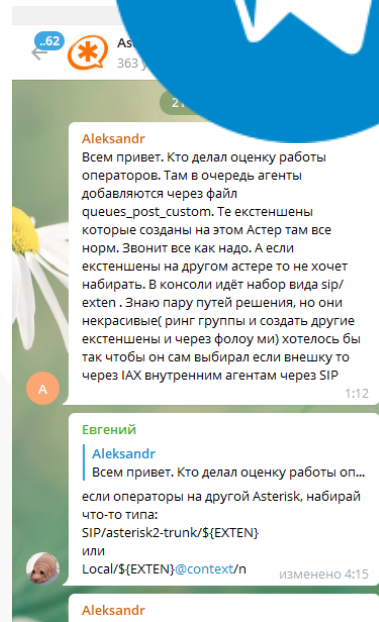
Наша группа в Telegram

Задавайте свои вопросы и общайтесь с коллегами в Telegram-канале

@asterisker

Авторы наиболее продуктивных и полезных вопросов и комментариев получают ценные призы от спонсоров

Вопрос задавайте с хештегом #вопрос



ПОДАРКИ ОТ СПОНСОРОВ



Хотите стать спонсором подарков?
Пишите на sergey@voxlink.ru



За лучший вопрос
в чате Telegram
@Asterisker.
Хэштег «#вопрос»

Телефон Fanvil X1SP

- Самый доступный IP телефон на рынке.
- При этом в нем: Opus и HD кодеки, экран с подсветкой.
- Именно в этой модификации даже есть PoE

Спонсор подарка:



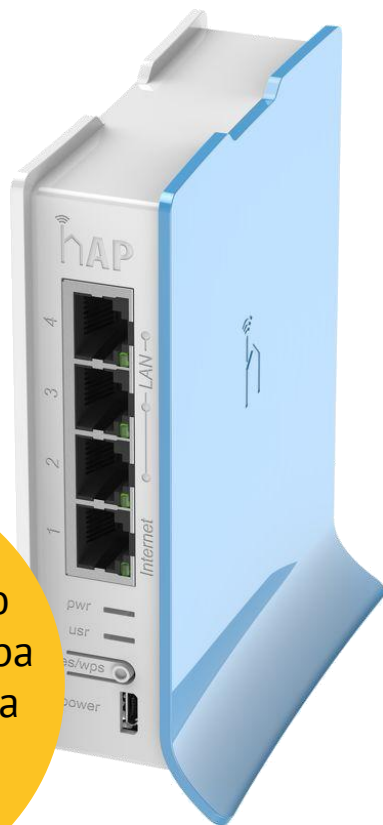
VOXLINK: ASTERISK
ПРОФЕССИОНАЛЬНО



Mikrotik HAP Lite

- Самый крутой роутер в мире за 25 долларов
- Все, что нужно для хорошей работы IP-телефонии
- Все виды VPN
- Независимые порты и WiFi
- Стоечный и настенный вариант крепления

За лучшую идею
для темы вебинара
или для статьи на
сайте **Хэштег**
«#идея»



Спонсор подарка:



VOXLINK: ASTERISK
ПРОФЕССИОНАЛЬНО



За самое клевое
селфи
SelfieChallenge

Мерч от VoxLink

- Сумка «TCP-пакет | IP-пакет»
- Кружка VoxLink
- Блокнот и ручка
- Наклейка с логотипом Asterisk

Спонсор подарка:

VoxLink – интегратор систем IP-телефонии на базе Asterisk



VOXLINK: ASTERISK
ПРОФЕССИОНАЛЬНО

Что можно делать с помощью AMI



Обзвонщик, используя
Originate

```
[Syntax]
Action: Originate
[ActionID:] <value>
Channel: <value>
[Exten:] <value>
[Context:] <value>
[Priority:] <value>
[Application:] <value>
[Data:] <value>
[Timeout:] <value>
[CallerID:] <value>
[Variable:] <value>
[Account:] <value>
[EarlyMedia:] <value>
[Async:] <value>
[Codecs:] <value>
[ChannelId:] <value>
[OtherChannelId:] <value>

[Synopsis]
Originate a call.

[Description]
Generates an outgoing call to a <Extension>/<Context>/<Priority> or
<Application>/<Data>
```

Что можно делать с помощью AMI



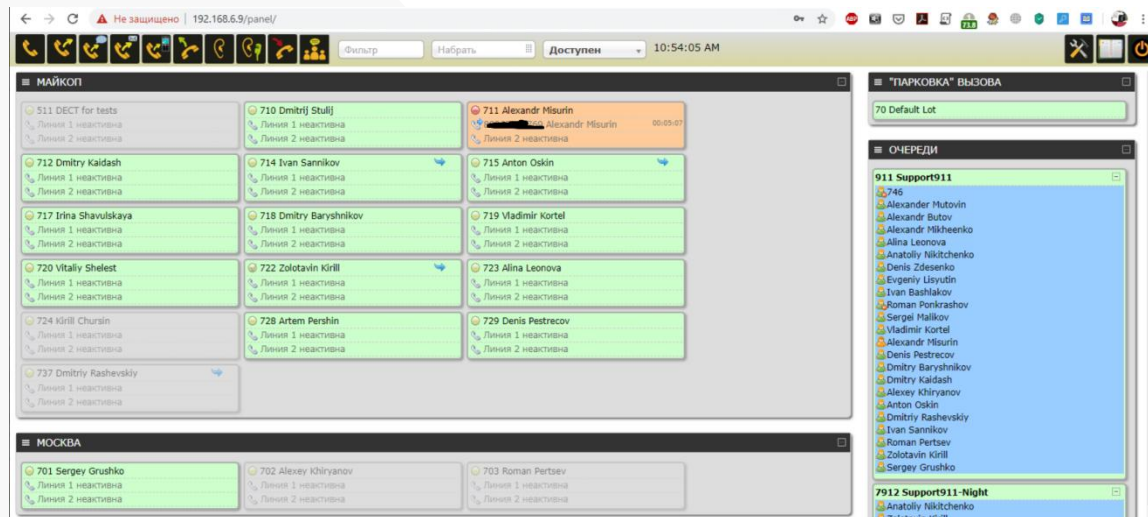
Перенаправлять (трансфер) нужные каналы на любой модуль в диалплане Asterisk (Redirect)

```
[Syntax]
Action: Redirect
[ActionID:] <value>
Channel: <value>
[ExtraChannel:] <value>
Exten: <value>
[ExtraExten:] <value>
Context: <value>
[ExtraContext:] <value>
Priority: <value>
[ExtraPriority:] <value>

[Synopsis]
Redirect (transfer) a call.
```

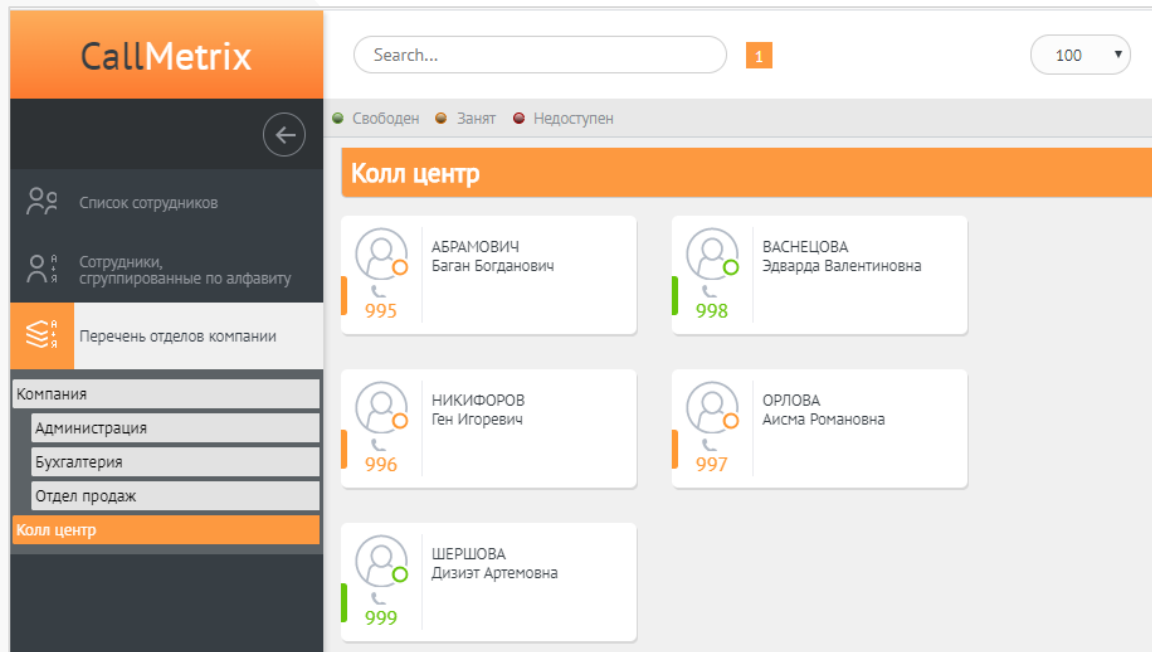

Что можно делать с помощью АМІ

Написать
свою панель
мониторинга
типа FOP2



Что можно делать с помощью АМІ

Написать
свою панель
мониторинга
типа FOP2



Что такое AMI?

Asterisk Manager Interface - это интерфейс мониторинга и управления системой, предоставляемый Asterisk. Проще говоря это **API Asterisk**.

- Позволяет в реальном времени отслеживать события, происходящие в системе. Такие события называются **Events**
- Позволяет выполнять определенные действия. Такие действия называются **Actions**

Комбинирование Events и Actions позволяют писать различные интересные и полезные приложения для взаимодействия с вашей IP-атс

Подключение к АМІ

Для подключения к АМІ потребуется минимально настроить manager.conf:

```
[general]
enabled = yes           ;Включить АМІ
bindaddr = 127.0.0.1    ;Разрешить принимать соединения только с localhost

[testuser]
secret=SuperP4ssw0rd    ;Сложный пароль
read=all                ;Получать все типы событий
write=all               ;Разрешить пользователю выполнять все действия
```

Для применения изменений необходимо выполнить в CLI Asterisk следующую команду: **manager reload**

Подключение к AMI

Теперь можно подключиться к интерфейсу AMI с помощью telnet

```
[root@testdistro13 ~]# telnet localhost 5038
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Asterisk Call Manager/2.10.5
Action: Login
Username: testuser
Secret: SuperP4ssw0rd

Response: Success
Message: Authentication accepted

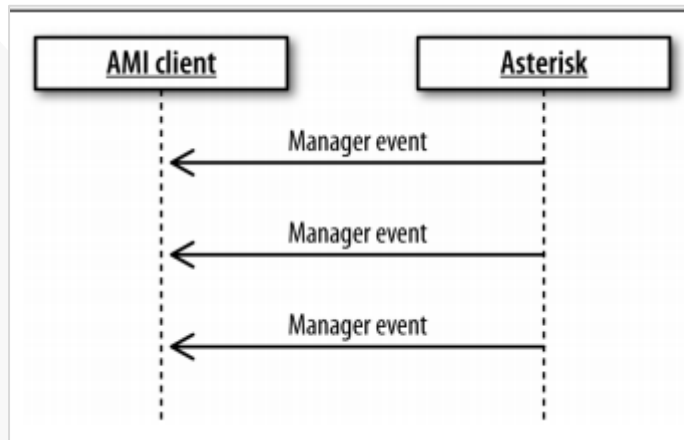
Event: FullyBooted
Privilege: system,all
Status: Fully Booted
```

```
Action: Ping

Response: Success
Ping: Pong
Timestamp: 1588755403.702799
```

Обзор протокола

События менеджера **Events** - это односторонние сообщения, посылаемые Asterisk клиентам AMI для сообщения того, что произошло в системе



Обзор протокола

```
Event: PeerStatus
Privilege: system,all
Timestamp: 1589288772.005061
ChannelType: SIP
Peer: SIP/100
PeerStatus: Unregistered
Cause: Expired
```

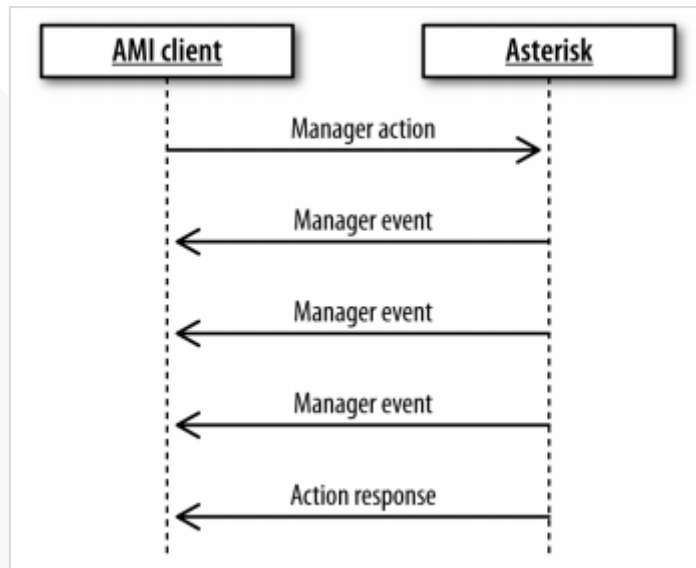
```
Event: SuccessfulAuth
Privilege: security,all
```

```
Event: PeerStatus
Privilege: system,all
Timestamp: 1589288863.711879
ChannelType: SIP
Peer: SIP/100
PeerStatus: Registered
Address: 10.10.33.1:5066
```

```
Event: ExtensionStatus
Privilege: call,all
```

Обзор протокола

Действия **Actions** - это запросы от клиента к Asterisk для выполнения некоторого действия и возврата результата



Обзор протокола

```
Action: QueueSummary
Queue: 600

Response: Success
EventList: start
Message: Queue summary will follow

Event: QueueSummary
Queue: 600
LoggedIn: 1
Available: 1
Callers: 0
HoldTime: 0
TalkTime: 0
LongestHoldTime: 0

Event: QueueSummaryComplete
EventList: Complete
ListItems: 1
```



Что можно делать с помощью AMI

В режиме реального времени отслеживать состояние транков (**Registry**)

```
Event: Registry
Privilege: system,all
Timestamp: 1589290816.566628
ChannelType: SIP
Username: 74993801
Domain: 16.voxlink.ru
Status: Registered
```

```
[Synopsis]
Raised when an outbound registration completes.

[Syntax]
Event: Registry
ChannelType: <value>
Username: <value>
Domain: <value>
Status: <value>
Cause: <value>

[Arguments]
ChannelType
    The type of channel that was registered (or not).
Username
    The username portion of the registration.
Domain
    The address portion of the registration.
Status
    The status of the registration request.
    Registered
    Unregistered
    Rejected
    Failed
Cause
    What caused the rejection of the request, if available.
```

Технические характеристики AMI

- По умолчанию AMI является асинхронным протоколом
- Работает по TCP
- Поддерживает TLS
- Стандартный порт 5038
- Настройки в файле `manager.conf`
- Настройки для AJAM в файле `http.conf`
- С 12 версии астериска AMI имеет спецификацию v2 и следует семантическому версионированию
- В **11** версии asterisk суммарно **184** событий и действий то в **17** уже **323**

Обзор протокола

- Команды **CLI Asterisk** относящиеся к AMI начинаются с **manager** – далее **TAB**'ом можно посмотреть доступные действия
- Вывод событий в **CLI Asterisk** можно включить командой **manager set debug on**, при этом вам нужно **обязательно** подключиться по **telnet** к **AMI** для создания активной сессии
- Просмотреть список всех доступных **событий (events)** можно через команду **manager show events**. А более подробно нужное событие через **manager show event <нужный event>**
- Просмотреть список всех доступных **действий (actions)** можно через команду **manager show actions**. А более подробно нужное действие **через manager show action <нужный action>**
- Так же можно обратиться в **wiki asterisk**:
<https://wiki.asterisk.org/wiki/display/AST/Asterisk+13+AMI+Actions>
<https://wiki.asterisk.org/wiki/display/AST/Asterisk+13+AMI+Events>



Синтаксис AMI

- Все сообщения AMI (события, действия, ответы) кодируются одинаково и являются текстом
- Формат события/действия «заголовок»: «значение»
- Конец каждого параметра в событии - "\r\n" (CR LF) в стиле SMTP, HTTP и т.д.
- Если в значении параметра больше одной строки, строки значения будут разделены с помощью "\n".
- Конец каждого сообщения - пустая строка (новое сообщение в потоке нужно разбивать по "\r\n\r\n").
- События всегда имеют заголовок Event и заголовок Privilege. В заголовке Event указывается имя события, а в заголовке Privilege - уровни разрешений, связанные с данным событием.

Синтаксис AMI

- В ответе на запрос может быть список событий. В этом случае в событии с параметром **Response** будет также параметр **EventList: start**. После вывода всего списка событий будет дополнительно отправлено событие завершения списка с параметром **EventList: Complete**, а так же длина этого списка **ListItems: X**
- Особый случай - конец вывода ответа на **Command** заканчивается на **--END COMMAND--** (если ответ разбит на несколько строк - строки будут также разделены с помощью `"\n"`, а не `"\r\n"`)
- В запросах **AMI** можно использовать параметр **ActionID: X**, тогда в каждом ответе на это сообщение будет добавлен параметр **ActionID** с тем же значением. При этом значение этого параметра должно быть уникальным в рамках одного подключения/сессии AMI. Можно начинать с 1 и увеличивать на 1 для каждого нового запроса.



Что можно делать с помощью AMI

Отслеживать направления дозвона и завершать нежелательные вызовы
(DialBegin, Hangup)

```
[Syntax]
Action: Hangup
[ActionID:] <value>
Channel: <value>
[Cause:] <value>

[Synopsis]
Hangup channel.

[Description]
Hangup a channel.

[Arguments]
ActionID
    ActionID for this transaction. Will be returned.
Channel
    The exact channel name to be hungup, or to use a regular expression, set
    this parameter to: /regex/
    Example exact channel: SIP/provider-0000012a
    Example regular expression: /^SIP/provider-.*$/
Cause
    Numeric hangup cause.
```

```
Event: DialBegin
[Synopsis]
Raised when a dial action has started.

[Syntax]
Event: DialBegin
Channel: <value>
ChannelState: <value>
ChannelStateDesc: <value>
CallerIDNum: <value>
CallerIDName: <value>
ConnectedLineNum: <value>
ConnectedLineName: <value>
Language: <value>
AccountCode: <value>
Context: <value>
Exten: <value>
Priority: <value>
Uniqueid: <value>
Linkedid: <value>
DestChannel: <value>
DestChannelState: <value>
DestChannelStateDesc: <value>
DestCallerIDNum: <value>
DestCallerIDName: <value>
DestConnectedLineNum: <value>
DestConnectedLineName: <value>
DestLanguage: <value>
DestAccountCode: <value>
DestContext: <value>
DestExten: <value>
DestPriority: <value>
DestUniqueid: <value>
DestLinkedid: <value>
DialString: <value>
```



Что можно делать с помощью AMI

Отслеживать направления
дозвона и завершать
нежелательные вызовы

```
Event: DialBegin
Privilege: call,all
Timestamp: 1589297722.838104
Channel: SIP/100-0000000d
ChannelState: 4
ChannelStateDesc: Ring
CallerIDNum: 74959898533
CallerIDName: <unknown>
ConnectedLineNum: <unknown>
ConnectedLineName: <unknown>
Language: ru
AccountCode:
Context: macro-dialout-trunk
Exten: s
Priority: 20
Uniqueid: 1589297722.43
Linkedid: 1589297722.43
ChanVariable: CALLFILENAME=out-84959898533-unknown-20200512-183522-1589297722.43
ChanVariable: FROM_DID=
ChanVariable: CHANNEL(hangupsource)=
ChanVariable: DIALSTATUS=
ChanVariable: PREDICTION=
ChanVariable: SIPURI=sip:100@10.10.33.1:5066
DestChannel: SIP/vox-74993809009-0000000e
DestChannelState: 0
DestChannelStateDesc: Down
DestCallerIDNum: 84959898533
DestCallerIDName: <unknown>
DestConnectedLineNum: 74959898533
DestConnectedLineName: <unknown>
DestLanguage: ru
DestAccountCode:
DestContext: from-trunk-fix
DestExten: 84959898533
DestPriority: 1
DestUniqueid: 1589297722.44
DestLinkedid: 1589297722.43
DestChanVariable: CALLFILENAME=out-84959898533-unknown-20200512-183522-1589297722.43
DestChanVariable: FROM_DID=
DestChanVariable: CHANNEL(hangupsource)=
DestChanVariable: DIALSTATUS=
DestChanVariable: PREDICTION=
DestChanVariable: SIPURI=
DialString: vox-74993809009/84959898533
```



Классы событий AMI

В разделе [username] параметры read и write определяют к каким действиям и событиям AMI имеет доступ конкретный пользователь.

На текущий момент существует 20 таких классов: **all, system, call, log, verbose, agent, user, config, command, dtmf, reporting, cdr, dialplan, originate, agi, cc, aoc, test, security и message.**

В файле manager.conf.sample вы найдете более подробное описание каждого из них, а так же новые классы если они появятся в будущих версиях asterisk.

Фильтры событий АМІ

- Если событий слишком много то вы можете использовать фильтры событий для того что бы уменьшить их и сфокусироваться только на нужных. Для этих целей используется опция eventfilter которая позволяет внести событие в белый список или черный.
- Если после знака = идет восклицательный знак, значит событие добавляется в черный список
- Eventfilter поддерживает регулярные выражения, например:

```
eventfilter=Event: Newchannel  
eventfilter=Channel: (PJ)?SIP/(james|jim|john)-  
eventfilter=!Channel: DAHDI/
```

Фильтры событий АМІ

Обработка фильтров происходит следующим образом:

- Если фильтры не настроены, то отображаются все сообщения которые разрешены в параметре read
- Если есть только белые фильтры, то сперва подразумевается черный список всех событий, затем отображаться будут только те события которые указаны в белом списке, остальные игнорируются
- Если есть только черные списки, то сперва подразумевается белый список всех событий, затем отображаться будут все сообщения кроме тех что указаны в черном

Фильтры событий АМІ

Обработка фильтров происходит следующим образом:

- Если есть и белые и черные то сперва подразумевается черный список всех событий, затем обрабатывается белый список и в конце обрабатывается черный список

Что можно делать с помощью AMI

Мониторить качество вызовов с помощью событий RTCP (RTCPSent)

```
SentPackets
  The number of packets the sender has sent. Only valid when PT is '200(SR)'.
SentOctets
  The number of bytes the sender has sent. Only valid when PT is '200(SR)'.
ReportXSourceSSRC
  The SSRC for the source of this report block.
ReportXFractionLost
  The fraction of RTP data packets from 'ReportXSourceSSRC' lost since the
  previous SR or RR report was sent.
ReportXCumulativeLost
  The total number of RTP data packets from 'ReportXSourceSSRC' lost since
  the beginning of reception.
ReportXHighestSequence
  The highest sequence number received in an RTP data packet from
  'ReportXSourceSSRC'.
ReportXSequenceNumberCycles
  The number of sequence number cycles seen for the RTP data received from
  'ReportXSourceSSRC'.
ReportXJitter
  An estimate of the statistical variance of the RTP data packet interarrival
  time, measured in timestamp units.
ReportXLSR
  The last SR timestamp received from 'ReportXSourceSSRC'. If no SR has been
  received from 'ReportXSourceSSRC', then 0.
ReportXDLR
  The delay, expressed in units of 1/65536 seconds, between receiving the
  last SR packet from 'ReportXSourceSSRC' and sending this report.
```

[Synopsis]
Raised when an RTCP packet is sent.

[Syntax]
Event: RTCPSent
Channel: <value>
ChannelState: <value>
ChannelStateDesc: <value>
CallerIDNum: <value>
CallerIDName: <value>
ConnectedLineNum: <value>
ConnectedLineName: <value>
Language: <value>
AccountCode: <value>
Context: <value>
Exten: <value>
Priority: <value>
Uniqueid: <value>
Linkedid: <value>
SSRC: <value>
PT: <value>
To: <value>
ReportCount: <value>
[SentNTP:] <value>
[SentRTP:] <value>
[SentPackets:] <value>
[SentOctets:] <value>
ReportXSourceSSRC: <value>
ReportXFractionLost: <value>
ReportXCumulativeLost: <value>
ReportXHighestSequence: <value>
ReportXSequenceNumberCycles: <value>
ReportXJitter: <value>
ReportXLSR: <value>
ReportXDLR: <value>

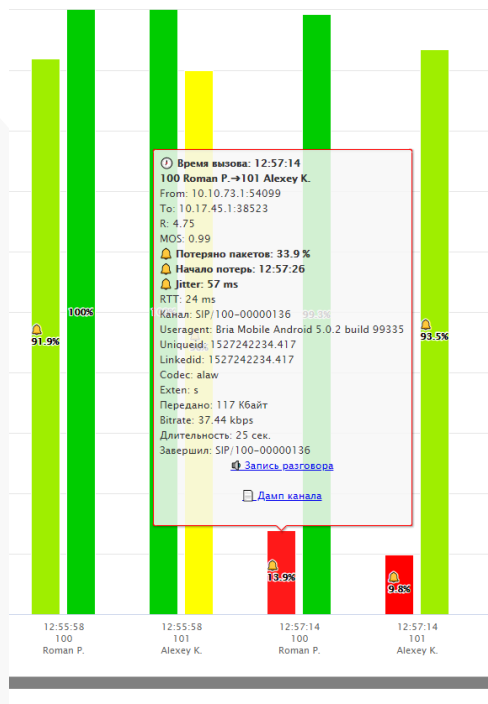
```
Event: RTCPSent
Privilege: reporting,all
Timestamp: 1589292716.703980
Channel: SIP/100-00000007
ChannelState: 6
ChannelStateDesc: Up
CallerIDNum: 100
CallerIDName: Roman P.
ConnectedLineNum: *43
ConnectedLineName: Echo Test
Language: ru
AccountCode:
Context: app-echo-test-echo
Exten: 1
Priority: 1
Uniqueid: 1589292701.33
Linkedid: 1589292701.33
ChanVariable: CALLFILENAME=
ChanVariable: FROM_DID=
ChanVariable: CHANNEL(hangupsource)=
ChanVariable: DIALSTATUS=
ChanVariable: PREDICTION=
ChanVariable: SIPURI=sip:100@10.10.33.1:5066
To: 10.10.33.1:8001
From: 10.17.45.1:38351
SSRC: 0x22ca10aa
PT: 200(SR)
ReportCount: 1
SentNTP: 1589292716.703800
SentRTP: 119520
SentPackets: 747
SentOctets: 119520
ReportXSourceSSRC: 0x28c13078
ReportXFractionLost: 0
ReportXCumulativeLost: 0
ReportXHighestSequence: 50503
ReportXSequenceNumberCycles: 0
ReportXJitter: 17
ReportXLSR: 757490909
ReportXDLR: 1.3630
```



Что можно делать с помощью AMI

Мониторить качество вызовов с помощью событий **RTCP (RTCPSent)**

Потеря до Anna SIP/6117 - 1.75%	14:01
Потеря до Anna SIP/6117 - 1.78%	14:04
Потеря до Anna SIP/6117 - 5.45%	14:10
Потеря до Anna SIP/6117 - 4.28%	14:11



AJAM

Asynchronous Javascript Asterisk Manager

- **AJAM** - это «новая» технология, которая позволяет веб-браузерам или другим приложениям с поддержкой **HTTP** получать прямой доступ к интерфейсу Asterisk Manager (**AMI**) через **HTTP**.
- **AMI** через **TCP** предлагает только **один** тип структуры сообщений, **AJAM** предлагает несколько вариантов кодирования. Вы можете получать ответы **в том же формате** что и в TCP, **в формате XML** или **в виде базовой HTML-страницы**. Тип кодировки выбирается на основе поля в URL

AJAM

Asynchronous Javascript Asterisk Manager

При подключении к интерфейсу **TCP** для **AMI** события доставляются асинхронно. При использовании **AMI (AJAM)** через **HTTP** необходимо получить события путем опроса для них. Вы получаете события по протоколу HTTP, выполняя действие **WaitEvent**.

Вам так же нужно будет разработать механизмы в приложении чтобы гарантировать что буферизованные события часто опрашиваются.

AJAM

Asynchronous Javascript Asterisk Manager

- Включение доступа к AMI по протоколу HTTP требует наличия `/etc/asterisk/manager.conf` и `/etc/asterisk/http.conf`.
- AMI должен быть включен в `manager.conf` с параметром `enabled`, установленным в `yes` и `webenabled` должен быть установлен в значение `yes` в разделе `[general]` чтобы разрешить доступ по протоколу HTTP.
- Опция `enabled` в `http.conf` должна быть установлена в `yes` чтобы включить сам HTTP-сервер.
- Для применения изменений необходимо выполнить **`module reload http`** и **`manager reload`**

AJAM

кодирование

Проверить доступные типы кодирования можно с помощью CLI команды Asterisk – **http show status**:

```
Connected to Asterisk 13.29.2 currently running on testdistrol3 (pid = 15185)
testdistrol3*CLI> http show status
HTTP Server Status:
Prefix:
Server: Asterisk/13.29.2
Server Enabled and Bound to 0.0.0.0:8088

Enabled URI's:
/httpstatus => Asterisk HTTP General Status
/amanager => HTML Manager Event Interface w/Digest authentication
/arawman => Raw HTTP Manager Event Interface w/Digest authentication
/manager => HTML Manager Event Interface
/rawman => Raw HTTP Manager Event Interface
/static/... => Asterisk HTTP Static Delivery
/amxml => XML Manager Event Interface w/Digest authentication
/mxml => XML Manager Event Interface
/ari/... => Asterisk RESTful API
/ws => Asterisk HTTP WebSocket
```



AJAM

аутентификация

Существует два метода выполнения аутентификации через HTTP.

Первый - это использование действия Login, аналогичного аутентификации с помощью собственного интерфейса TSP. После успешной аутентификации Asterisk предоставит файл cookie, который идентифицирует аутентифицированный сеанс.

```
$ curl -v "http://localhost:8088/rawman?action=login&username=name&secret=password"
```

АЈАМ

аутентификация

Существует два метода выполнения аутентификации через HTTP.

Второй вариант аутентификации - это HTTP-дайджест аутентификации. Чтобы указать, что следует использовать дайджест аутентификацию HTTP, префикс типа кодировки в URL-адресе запроса должен содержать а:

```
$ curl -v --digest -u login:password http://localhost:8088/arawman?action=ping
```

AJAM

кодирование

AJAM предоставляет несколько вариантов типов кодирования:

- **rawman/arawman** – такой же тип кодирования сообщений как и через TCP сокет AMI

```
curl -v --digest -u login:password http://localhost:8088/arawman?action=ping
```

AJAM

кодирование rawman/arawman

```
← → ↻ ⓘ Не защищено | 10.17.45.1:8088/rawman?action=sippeers

Response: Success
EventList: start
Message: Peer status list will follow

Event: PeerEntry
Channeltype: SIP
ObjectName: 100
ChanObjectType: peer
IPaddress: -none-
IPport: 0
Dynamic: yes
AutoForcerport: no
Forcerport: yes
AutoComedia: no
Comedia: yes
VideoSupport: no
TextSupport: no
ACL: yes
Status: UNKNOWN
RealtimeDevice: no
Description:
Accountcode:

Event: PeerEntry
Channeltype: SIP
ObjectName: 101
ChanObjectType: peer
IPaddress: -none-
IPport: 0
Dynamic: yes
AutoForcerport: no
Forcerport: yes
AutoComedia: no
Comedia: yes
VideoSupport: no
TextSupport: no
ACL: yes
Status: UNKNOWN
RealtimeDevice: no
Description:
Accountcode:
```

AJAM

кодирование

AJAM предоставляет несколько вариантов типов кодирования:

- **manager/amanager** - предоставляет ответ в простой HTML-форме. Интерфейс в первую очередь полезен для экспериментов с AMI

```
curl -v --digest -u login:password http://localhost:8088/amanager?action=ping
```

AJAM

кодирование manager/amanager

Manager Tester

Action: login ▾ or
CLI Command
user all pass

Response	Error
Message	Permission denied

Manager Tester

Action: -----> ▾ or
CLI Command
user pass

Response	Success
EventList	start
Message	Peer status list will follow

Event	PeerEntry
Channeltype	SIP
ObjectName	100
ChanObjectType	peer
IPaddress	-none-
IPport	0
Dynamic	yes
AutoForcerport	no
Forcerport	yes
AutoComedia	no
Comedia	yes
VideoSupport	no
TextSupport	no
ACL	yes
Status	UNKNOWN
RealtimeDevice	no
Description	
Accountcode	



AJAM

кодирование

AJAM предоставляет несколько вариантов типов кодирования:

- **mxml/amxml** - предоставляет ответы в формате XML

```
curl -v --digest -u login:password http://localhost:8088/amxml?action=ping
```

AJAM

кодирование mxml/atxml

← → ↺ ⓘ Не защищено | 10.17.45.1:8088/mxml?action=ping

This XML file does not appear to have any style information associated with it. The document tree is shown below.

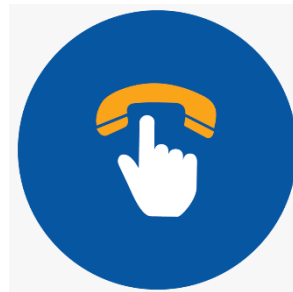
```
<ajax-response>
  <response type="object" id="unknown">
    <generic response="Success" ping="Pong" timestamp="1588749234.415606"/>
  </response>
</ajax-response>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<ajax-response>
  <response type="object" id="unknown">
    <generic response="Success" eventlist="start" message="Peer status list will follow"/>
  </response>
  <response type="object" id="unknown">
    <generic event="PeerEntry" channeltype="SIP" objectname="100" chanobjecttype="peer" ipaddress="-
none-" ipport="0" dynamic="yes" autoforcerport="no" forcerport="yes" autocomedia="no"
comedia="yes" videosupport="no" textsupport="no" acl="yes" status="UNKNOWN" realtimedevic="no"
description="" accountcode=""/>
  </response>
```



Что можно делать с помощью AMI/AJAM



Реализовать Click2Call в вашей CRM
или любой другой системе

```
curl --digest -u login:pass "http://localhost:8088/arawman?action=originate&channel=SIP/100&context=from-internal&exten=84959898533&priority=1"
```

Безопасность АМІ

Безопасность АМІ так же важна как и SIP безопасность. Ведь имея доступ к возможности управления IP-АТС можно с легкостью заставить звонить станцию на “нужные” направления, получить доступ к конфигурационным файлам и т.д.



VOXLINK: ASTERISK
ПРОФЕССИОНАЛЬНО

Безопасность AMI

- **Не открывайте** порт **5038/8088** во внешний мир или ограничивайте его списком конкретных ip-адресов как на роутере/firewall так и с помощью **deny/permit** в настройках пользователя AMI
- Используйте **TLS** для внешних подключений извне
- **Не наделяйте** пользователя AMI **полными правами**
- Используйте **сложные** пароли
- Используйте **MD5** аутентификацию с помощью action **Challenge**
- Особое внимание уделяйте командам **system**, **command** и **originate**

Что можно делать с помощью AMI

Отслеживать смену **User-Agent** при регистрации или перерегистрации пира (**SIPpeers**, **SIPshowpeer**)

```
745.Test
User-Agent on peer SIP/105 (37.110.158.23:5060) changed
from "None" to "Cisco-CP7960G/8.0" 11:31
```

```
745.Test
User-Agent on peer SIP/100 (10.10.33.1:5066) changed from
"None" to "Zoiper for Windows 2.43 r24984" 11:37
```

```
745.Test
User-Agent on peer PJSIP/777 (192.168.192.103:5060)
changed from "New peer" to "Zoiper for Windows 2.43
r24984" 14:26
```

Библиотеки для работы с АМІ

В вики астериска можно найти список готовых библиотек под разные языки программирования для работы с АМІ:

<https://wiki.asterisk.org/wiki/display/AST/AMI+Libraries+and+Frameworks>

Вы так же можете поискать что-то готовое на просторах GitHub



Библиотеки для работы с AMI

Name	Language	Website	Protocols
Asterisk-Java	Java	https://blogs.reucon.com/asterisk-java/	AMI/FastAGI
StarPy	Python+Twisted	https://github.com/asterisk/starpy	AMI/FastAGI
Panoramisk	Python+AsyncIO	https://github.com/gawel/panoramisk	AMI/FastAGI
PAMI	PHP	https://github.com/marcelog/PAMI	AMI
Pyst2	Python	https://github.com/rdegges/pyst2	AMI/AGI
Adhearsion	Ruby	http://www.adhearsion.com/	AMI/FastAGI
node-asterisk	Node.js	https://github.com/danjenkins/node-asterisk-ami	AMI
AMI-IO	Node.js	https://github.com/NumminorihSF/ami-io	AMI
NodeJS-AsteriskManager	Node.js	https://github.com/pipobscure/NodeJS-AsteriskManager	AMI
AsterNET	.NET	https://github.com/AsterNET/AsterNET	AMI/FastAGI
AmiClient	.NET	https://github.com/alexforster/AmiClient	AMI



Плюсы АМІ



Идеален для мониторинга чего-либо (транки, пиры, очереди) в Asterisk



Позволяет интегрироваться практически с любыми другими системами



Гибче чем использование Call Files для обзвонів



Куча готовых библиотек на любой вкус и цвет



Гораздо более продвинутые возможности по сравнению с простым диалпланом

Особенности АМІ

Трудно парсить события

Решение:

- Использование готовых библиотек

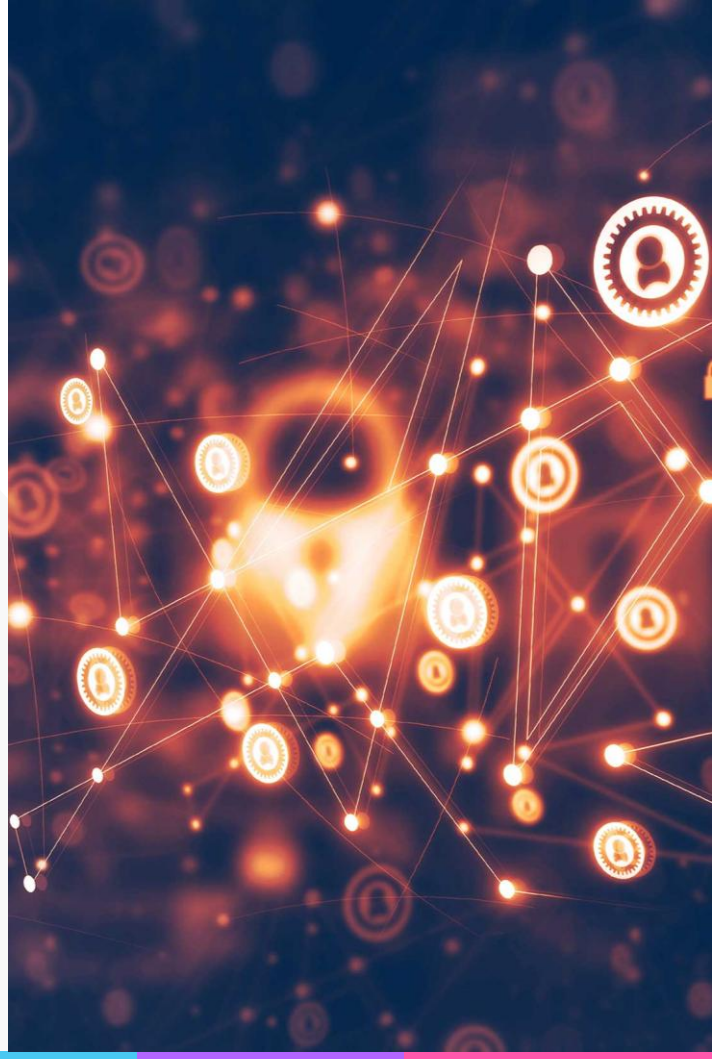


Особенности AMI

Проблемы при большом количестве подключений и событий

Решение:

- Использовать прокси
- Использовать фильтры событий

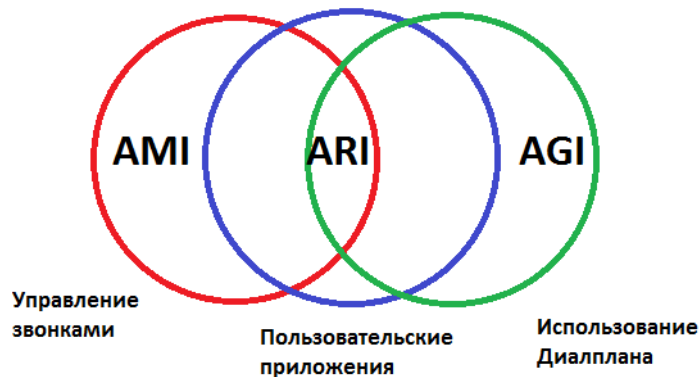


Особенности AMI

Не позволяет писать полноценные приложения как ARI

Решение:

- Использовать дополнительно AGI/ARI

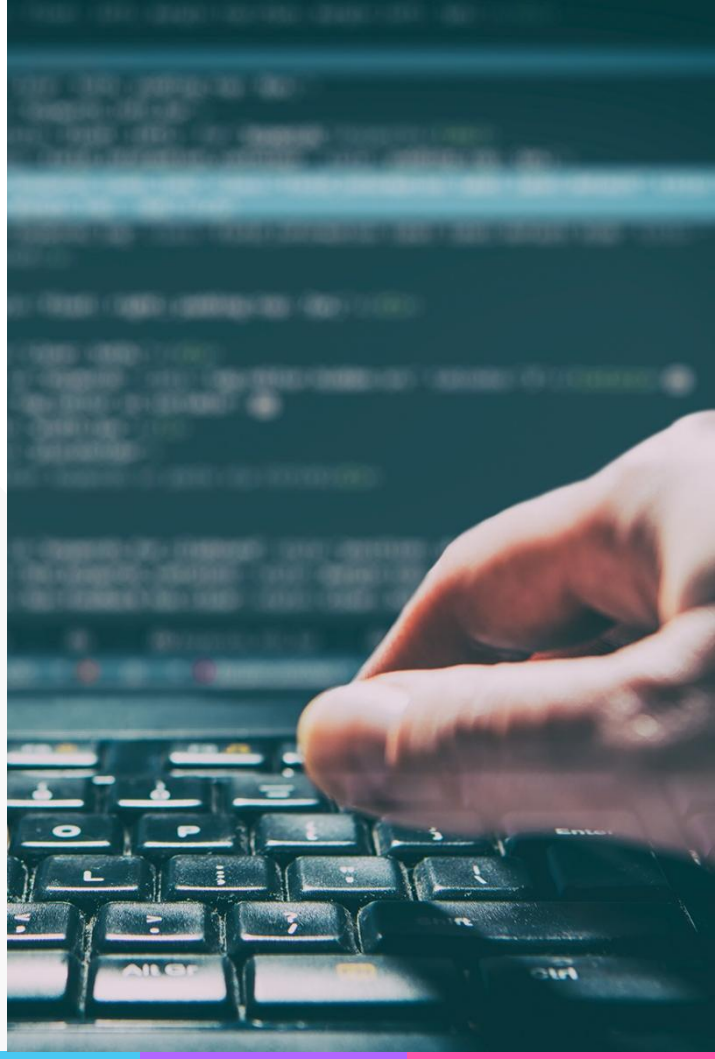


Особенности AMI

Трудности дебага

Решение:

- Писать свою систему логирования

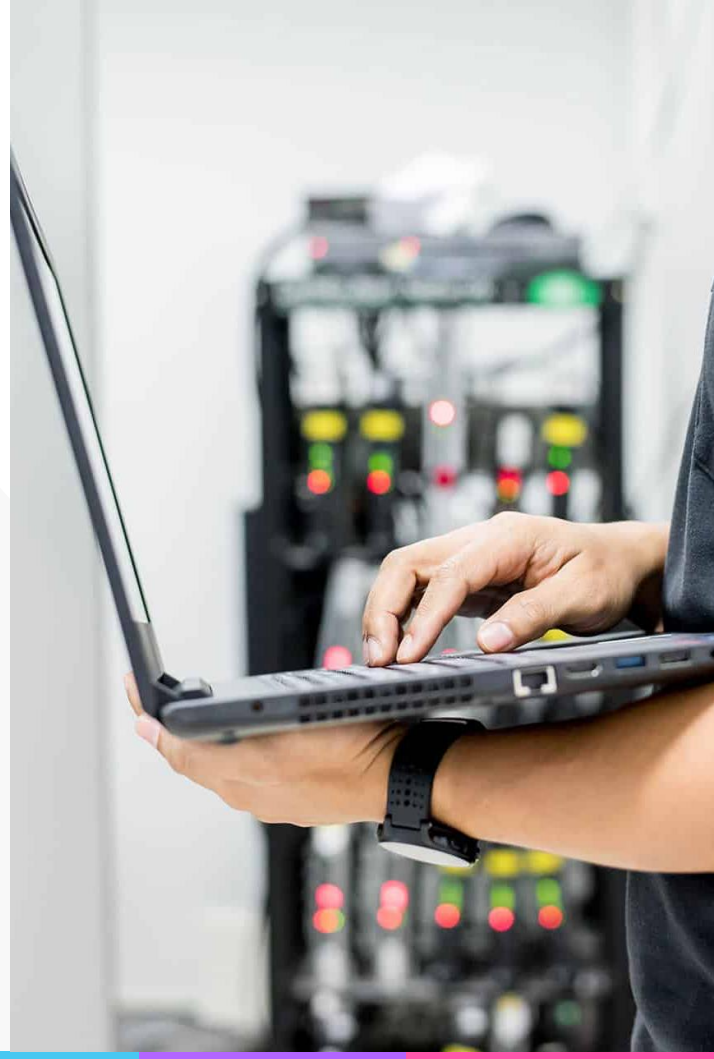


Особенности AMI

Средний/высокий уровень вхождения
для новичков

Решение:

- Использование готовых библиотек
- Пройти курсы от VoxLink



Что можно делать с помощью AMI

Управление **кастомными BLF** статусами (эквалайзер из ожидающих в очереди на панели телефона)

<https://www.youtube.com/watch?v=L0scisVxB3c>



Советы AMI



Используйте опцию **timestampevents=yes** в разделе **[general]** для отображения **unixtime** в событиях AMI



Используйте **классы событий** и **фильтры** для ограничения объема поступающей информации в AMI



Используйте опцию **channelvars=** для добавления к событиям свои переменные



При большом количестве подключений к AMI используйте **AMI-прокси**



Советы AMI

Если по какой-то причине в AMI нет необходимого Action, то можно использовать Getvar и функции диалплана в запросах:

Action: Getvar

Variable: DB_KEYS(AMPUSER)

```
Action: Getvar
Variable: DB_KEYS(AMPUSER)

Response: Success
Variable: DB_KEYS(AMPUSER)
Value: ,100,101,102,103,104,105,106,107,108,109,110,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,34567
```

Ваши вопросы?

Вопросы
из онлайн





СПАСИБО ЗА ВНИМАНИЕ!

Перцев Роман

roman@voxlink.ru