

**KAMAILIO
CONF/25**

Рецепты конфигурации OpenSIPS через призму опыта





Артём Друзь

Старший инженер

Контур

Сертифицированный специалист по
OpenSIPS (OpenSIPS Certified
Professional ID 7122-03). VoIP архитектор
гибридных систем телефонии

Проблематика доклада:



- Изобилие костылей и неэффективных алгоритмических решений в среднестатистических скриптах Kamailio и OpenSIPS.
- Отсутствие у существенного числа VoIP инженеров кругозора в синтаксисе доступном в Kamailio и OpenSIPS
- Существование практики использования частей скриптов, выложенных в публичный доступ или сгенерированных GPT моделями, без критического осмысления применяемых решений

Что обсудим?



- Настройку получения переменных без внешних сервисов
- Правила с использованием даты и времени
- Использование parallel и serial forking
- Шаблонизацию конфигурации — простые подмены и работа с условиями



Получение переменных из dialplan



Column Name	#	Data Type	Not Null	Auto Inc	Key	Default	Comment
123 id	1	int(10) unsigned	[v]	[v]	PRI		Unique ID
123 dpid	2	int(11)	[v]	[]			Dialplan ID
123 pr	3	int(11)	[v]	[]		0	Priority of rule
123 match_op	4	int(11)	[v]	[]			Matching operator for rule (0-equal, 1-regexp)
A-Z match_exp	5	char(64)	[v]	[]			Matching expression (regexp or string)
123 match_flags	6	int(11)	[v]	[]		0	Matching flags (0-case sensitive, 1-case insensitive)
A-Z subst_exp	7	char(64)	[]	[]		NULL	Substitution expression
A-Z repl_exp	8	char(32)	[]	[]		NULL	Replacement expression (sed like)
A-Z timerec	9	char(255)	[]	[]		NULL	Time recurrence used to match this rule
123 disabled	10	int(11)	[v]	[]		0	Specifies if the command can be used, or is disabled
A-Z attrs	11	char(255)	[]	[]		NULL	General attributes string to be returned in case of rule matching

Получение переменных из dialplan



```
1 |id(int,auto) dpid(int) pr(int) match_op(int) match_exp(string) match_flags(int) s
  ubst_exp(string,null) repl_exp(string,null) timerec(string,null) disabled(int) at
  trs(string,null)
2 :1:0:1:^7[89]40[0-9]{7}$:0::Abkhaziya::0:
3 :1:0:1:^7343[23][0-9]{6}$:0::RU-ABC::0:Yekaterinburg
4 :1:0:1:^7343[0-9]{7}$:0::RU-ABC::0:Sverdlovsk region
5 :1:0:1:^749[59][0-9]{7}$:0::RU-ABC::0:Moscow
6 :1:0:1:^7812[0-9]{7}$:0::RU-ABC::0:Saint-Petersburg
7 :1:0:1:^7[348][0-9]{9}$:0::RU-ABC::0:Somewhere
8 :1:0:1:^79[0-9]{9}$:0::RU-DEF::0:
```

Получение переменных из dialplan



```
1 loadmodule "db_text.so"
2
3 loadmodule "dialplan.so"
4 modparam("dialplan", "db_url", "text:///opt/dbtext")
5
6 route {
7     ...
8     #Нюансы с приоритетами при db_text в отличие от обычных БД
9     if ( !dp_translate(1, $fU, $var(priznak), $var(dp_rule_attrs)) ) {
10         xlog("L_WARN", "No translation rule for $fU\n");
11     }
12     switch ($var(priznak)) {
13     case "RU-ABC":
14         xlog("L_INFO", "Abonent $fU is RF ABC from $var(dp_rule_attrs) region\n");
15         break;
16     case "RU-DEF":
17         xlog("L_INFO", "Abonent $fU is RF mobile phone\n");
18         break;
19     default:
20         xlog("L_INFO", "Be careful - $fU is NOT from RF pool\n");
21     }
22     ...
23 }
24
```

Обработка на основе даты и времени



```
1 loadmodule "cfgutils.so"
2
3 route {
4     ...
5     if (check_time_rec("Europe/Moscow|20250925T090000|20250925T180000 || DAILY|20250928T000000 || TH,FR,SA")) {
6         xlog("L_INFO", "It is Asterconf'25 time! :D\n");
7     } else {
8         xlog("L_INFO", "Waiting for AsterConf ... \n");
9     }
10    ...
11 }
12
```

Обработка на основе даты и времени



- Как форматируется строка:
 - **timezone | dtstart | dtend | duration | freq | until | interval | byday | bymday | byyday | byweekno | bymonth**
- timezone выглядит громоздко **Asia/Yekaterinburg**. Необязательное поле
- dtstart, dtend и until подчиняются формату **20250925T101500**
- duration имеет формат **P3D8H30M10S**
- frequency может быть **YEARLY, MONTHLY, WEEKLY, DAILY, HOURLY, MINUTELY** и **SECONDLY**

Обработка на основе даты и времени



- Доступно к использованию:
- Из скрипта с помощью функции из модуля **cfgutils**
- При использовании функций на основе записей в БД модулей:
 - **dialplan**
 - **drouting** (для правил из **dr_rules**)
- Есть лайфхаки при использовании **dialplan**



Обработка на основе даты и времени



```
1
2 route{
3     ...
4     if(dp_translate(1, "w", $var(dp))) {
5         xlog("L_INFO", "It's work time. Let's found sale manager\n");
6     } else {
7         xlog("L_INFO", "Nobody here. Need to play some IVR\n");
8     }
9     ...
10 }
11
```

Serial и parallel forking



```
1 loadmodule "registrar.so"
2 modparam("registrar", "attr_avp", "$avp(reg_avp)")
3
4 loadmodule "tm.so"
5
6 route {
7     ...
8     $T_fr_inv_timeout = 30;
9     $avp(ruri_set) := $ru;
10    $avp(ruri_set) = "sip:extrauser@sipserver.example.com";
11    lookup("location");
12    #До OpenSIPS 3.6 attr_avp, заданный в модуле, вытащился бы в $avp(reg_avp)
13    xlog("L_INFO", "First attr avp is $(msg.branch.attr(reg_avp)[0])\n");
14    #До OpenSIPS 3.6 append_branch()
15    append_msg_branch("sip:seconduser@sipserver.example.com");
16    t_on_failure("fork_failure");
17    ...
18    route(relay);
19    exit;
20 }
```

Serial и parallel forking



```
1 route[relay] {
2     ...
3     if ( !t_relay() ) {
4         send_reply(500, "Internal Error");
5     }
6 }
7
8 failure_route[fork_failure] {
9     xlog("L_INFO", "Fork failed with code $T_reply_code\n");
10    if (t_was_cancelled()) {
11        exit;
12    }
13
14    if ( ( t_local_replied("all") || t_local_replied("last") ) && t_check_status("408") ||
15        t_check_status("3[0-9][0-9]|600|603") ) {
16        t_reply(181, "Call is Being Forwarded");
17        $T_fr_inv_timeout = 60;
18        $var(i) = 0;
19        while ( $avp(ruri_set)[$var(i)] != NULL ) {
20            #До OpenSIPS 3.4 флаг 'branch' был 'r'
21            lookup("location", "branch", $avp(ruri_set)[$var(i)]);
22            $var(i) = $var(i) + 1;
23        }
24    }
25    route(relay);
26    exit;
27 }
```

Шаблонизация конфигурации



```
1 log_level=3
2 server_header="{{ uas_header }}"
3 user_agent_header="{{ uac_header }}"
4 auto_aliases=no
5
6 udp_workers=8
7 socket=udp:{{ local_ip }}:5060 frag
8 socket=hep_udp:{{ local_ip }}:6061
9
10 mpath="{{ local_mpath }}"
11
12 ...
13
14 route {
15     ...
16     {% if node_id == 0 %}
17     topology_hiding("U", "zeronode");
18     {% else %}
19     if ( $(fU{s.int}) != 0 && $(fU{s.len}) == $(fU{s.int}{s.len}) ) {
20         topology_hiding("U", "/NonZero$fU/NonZero$tU");
21     } else {
22         topology_hiding("U", "$tU");
23     }
24     {% endif %}
25     ...
26 }
```

Демонстрация



Итоги



- **Переменные из dialplan:** оптимизация конфигурации
- **switch-case:** вместо множества if-else
- **RFC 2445 / RFC 5545:** iCalendar правила для учета даты и времени
- **Комбинирование добавления branch-ей** и failure route-ов
- **Встроенные трансформаторы** позволяют не заниматься придумыванием regex-ов и т.п.
- **Шаблонизация** при помощи M4 и Jinja2
- **Neovim** в качестве IDE с конфигурациями
- **Исходники и примеры конфигурации** требуют предварительного изучения документации

KAMAILIO
CONF/25

Спасибо за внимание!

Буду рад ответить на все ваши вопросы
сейчас или свяжитесь со мной в будущем:



Контур

Артём Друзь

Старший инженер

https://t.me/devvoipops_notes

