

Привязка IP+MAC

В стандартной реализации связки Billing и XGE (включая Softrouter) отсутствует привязка ip+mac. Если она требуется, необходимо отредактировать скрипт управления XGE в биллинге и, при необходимости, добавить синхронизатор MAC привязок.



IP+MAC привязки реализованы через создание статических записей в ARP таблице

- Скрипт событий
- Синхронизация привязок MAC+IP на XGE по БД биллинга
- Синхронизация IP+MAC по ARP таблице

Скрипт событий

1. Создайте папку для измененного скрипта:

```
mkdir -p /app/asr_billing/cfg/var/lib/event/  
touch /app/asr_billing/cfg/var/lib/event/xge_router.sh  
chmod a+x /app/asr_billing/cfg/var/lib/event/xge_router.sh
```

2. Создайте файл /app/asr_billing/cfg/var/lib/event/xge_router.sh со следующим содержанием:

```
#!/bin/bash  
  
. /usr/local/lib/carbon.shlib  
  
trap __exit EXIT  
  
TMPDIR=/tmp/${0##*/}  
mkdir -p $TMPDIR  
  
__exit() {  
    local ret=$?  
    rm -f $TMPDIR/*.$$  
    return $ret  
}  
  
# set -eux  
  
burst_in=''  
burst_out=''  
ceil_in=${ceil_in:-1000}  
rate_in=${rate_in:-1000}  
ceil_out=${ceil_out:-1000}  
rate_out=${rate_out:-1000}  
  
__xge_coa_send() {  
    echo "Filter-Id=\"$@" | radclient -x $nas_ip coa $coa_psw &>$TMPDIR/radclient.$$  
    ret=$?  
    # exit 254 , todo busy context  
    grep "no response from server" $TMPDIR/radclient.$$ && exit 254  
    cat $TMPDIR/radclient.$$  
    return $ret  
}  
  
user_add() {  
    [ "$auth_type" = "1" ] && __xge_coa_send session $ip start IPOE  
    __xge_coa_send session $ip mac set $mac  
}  
  
user_del() {  
    __xge_coa_send session $ip stop "user_del"  
    __xge_coa_send session $ip remove
```

```

}

user_accept() {
    [ "$auth_type" = "1" ] && __xge_coa_send session $ip start IPOE
    __xge_coa_send session $ip redirect blocked cancel
    __xge_coa_send session $ip nat $snatip
    __xge_coa_send session $ip mac $mac
}

user_drop() {
    __xge_coa_send session $ip redirect blocked
}

user_redirect() {
    __xge_coa_send session $ip redirect negbal
}

user_redirect_cancel() {
    __xge_coa_send session $ip redirect negbal cancel
}

user_rate_set() {
    __xge_coa_send session $ip rate set in $rate_in $ceil_in $burst_in out $rate_out $ceil_out $burst_out
}

user_rate_set_cancel() {
    __xge_coa_send session $ip rate remove
}

user_info() {
    __xge_coa_send session $ip info
}

user_test() {
    __xge_coa_send session $ip test
}

user_event_before() {
    :
}

user_event_after() {
    :
}

user_disconnect() {
    __xge_coa_send session $ip disconnect
}

__xge_list_local(){
    ipset -o save -l $4 | grep add | cut -d ' ' -f 3
}

__xge_ssh_send(){
    echo -e "chroot /app/xge $@\nexit\n" | ssh_send --port ${telnet_port:-33} -u
    ${telnet_login:-root} -p ${telnet_password:-servicemode} ${nas_ip:-127.0.0.1}
}

users_from_nas() {
    # coa
    local SYNCNCDIR="/var/lib/event/sync/$nas_ip"
    mkdir -p $SYNCNCDIR
    if [ "$nas_ip" != "169.254.18.12" ]; then
        __xge_ssh_send xgesh show list xge_blocked_list | grep '^[[0-9]].*' >
        $SYNCNCDIR/blocked_list.nas
        __xge_ssh_send xgesh show list xge_negbal_list | grep '^[[0-9]].*' >
        $SYNCNCDIR/negbal_list.nas
        __xge_ssh_send xgesh show list xge_auth_list | grep '^[[0-9]].*' >
        $SYNCNCDIR/auth_list.nas
    fi
    #
    if [ "$nas_ip" = "169.254.18.12" ]; then
        __xge_list_local xgesh show list xge_blocked_list | grep '^[[0-9]].*' >
        $SYNCNCDIR/blocked_list.nas
        __xge_list_local xgesh show list xge_negbal_list | grep '^[[0-9]].*' >

```

```

$SYNCDIR/negbal_list.nas
__xge_list_local xgesh show list xge_auth_list | grep '^[0-9].*' >
$SYNCDIR/auth_list.nas
fi
}

user_info(){
    echo '<pre>' > /tmp/${user_id}_user_info.new
    __xge_coa_send session $ip test human | grep "Reply-Message" | sed -e
's/Reply-Message =//g; s/^\s\+//g; s/^\s//g; s/"$//g' >> /tmp/${user_id}_user_info.new
    echo '</pre>' >> /tmp/${user_id}_user_info.new

    mv -f /tmp/${user_id}_user_info.new /tmp/${user_id}_user_info
    chown apache:apache /tmp/${user_id}_user_info
}

user_get_mac() {
    local TMPDIR=/tmp/nas_event_daemon/$nas_ip/user_get_mac/
    mkdir -p $TMPDIR/

    __xge_coa_send session $ip mac get | grep "Reply-Message" | sed -e 's/Reply-Message =//g;
s/^\s\+//g; s/^\s//g; s/"$//g' >> ${TMPDIR}/${user_id}

```

```
chmod 777 -R /tmp/nas_event_daemon/  
}
```

3. Перезапустите контейнер биллинга:

```
/app/asr_billing/service restart
```

Синхронизация привязок MAC+IP на XGE по БД биллинга

Описано создание привязок на стороне XGE в случае если Вы решили внедрить привязку IP+MAC, у абонентов заведены MAC-адреса в учетных записях и Вы хотите назначить привязки на стороне XGE.

1. Создайте файл скрипта

```
touch /app/base/cfg/set_xge_ip_mac.sh  
chmod a+x /app/base/cfg/set_xge_ip_mac.sh
```

Заполните скрипт /app/base/cfg/set_xge_ip_mac.sh

```
#!/bin/bash  
  
sqlexec "set heading off; select uf_ip2string(ip),mac from users where mac!='' and ip!=0 and  
deleted=0 and opt82=0 and abonent_id>1" | grep -v "^$" | sort -u | awk '$1$2{print $1;"$2}'  
> /tmp/ipmac.list  
  
while IFS=';' read ip mac; do  
    if [[ $1 == "remove" ]]; then  
        chroot /app/xge xgesh session $ip mac remove  
    elif [[ $1 == "set" ]]; then  
        chroot /app/xge xgesh session $ip mac set $mac  
    else  
        echo xgesh session $ip mac set $mac  
    fi  
done <<< "$(</tmp/ipmac.list)"
```

Скрипт ожидает передачи параметров:

- set - установить привязки
- remove - удалить привязки
- без параметров - напечатать в терминал команды для установки привязок по пользовательским сессиям без выполнения

2. Запустите синхронизацию:

```
bash -x /app/base/cfg/set_xge_ip_mac.sh set &>>/var/log/xge_ipmac_set.log
```

Лог синхронизации можно будет найти по пути /var/log/xge_ipmac_set.log

Синхронизация IP+MAC по ARP таблице

Синхронизация выполняется вручную при необходимости, например при внедрении функции если абоненты уже заведены, но ранее привязка не использовалась.

Скрыты получают MAC-адреса абонентов из ARP-таблицы XGE.

1. Создайте файлы скриптов и установите права на выполнение

```
touch /app/xge/cfg/mac_autoget.sh
touch /app/base/cfg/mac_process.sh
chmod a+x /app/xge/cfg/mac_autoget.sh
chmod a+x /app/base/cfg/mac_process.sh
```

2. Создайте указанные ниже файлы и со следующим содержимым:

/app/xge/cfg/mac_autoget.sh

```
#!/bin/bash

>/tmp/true_ipmac
# IP ,
# arp , c , ip mac
for ip in $(</tmp/users_ip_where_no_mac_in_db.list); do
    current_mac="$(xgesh session $ip mac get)"
    if [ "$current_mac" == "00:00:00:00:00:00" ]; then
        continue
    else
        echo $ip $current_mac >> /tmp/true_ipmac
        xgesh session $ip mac set $current_mac
    fi
done
```

/app/base/cfg/mac_process.sh

```
#!/bin/bash

# db IP , ip
get_list() {
    sqlxec "set heading off; select uf_ip2string(ip) from users where mac='' and ip!=0 and
    deleted=0 and opt82=0 and abonent_id>1" | grep -v "^$" | sort -u >
    /app/xge/tmp/users_ip_where_no_mac_in_db.list
}
#
mac_get() {
    chroot /app/xge/ /cfg/mac_autoget.sh
}

# , /app/xge/cfg/mac_autoget.sh
mac_save() {
    [ ! -s /app/xge/tmp/true_ipmac ] && exit 11
    while read ip mac ; do
        curl -XPOST 'http://169.254.80.82:8082/rest_api/v2/Users/' -d 'method1=objects.get&arg1={
        "ip":"'$ip'" }&method2=set&arg2={"mac":"'$mac'" }&method3=save&arg3={}'
    done <<< "$(cat /app/xge/tmp/true_ipmac)"
}

main() {
    get_list
    mac_get
    mac_save
}

main
```

3. Запустите синхронизацию:

```
bash -x /app/base/cfg/mac_process.sh &>>/var/log/xge_ipmac_arp_sync.log
```

Лог синхронизации можно будет найти по пути /var/log/xge_ipmac_arp_sync.log