



OBSD.RU

[Home](#)

## Kannel SMPP OpenBSD

[printable page](#)

Авторы: Виктор Гижевский [www.obsd.ru](http://www.obsd.ru), Александр Симаков [www.entropyware.info](http://www.entropyware.info)

Данная статья написана для OpenBSD 3.8, решение применялось на OpenBSD 3.8, 4.0, 4.2, Kannel 1.4.0 и 1.4.1. На более старших версиях, чем описанные в статье, меняются только версии пакетов на более новые.

В настоящее время рынок SMS и мультимедийного контента растет быстрыми темпами. Для построения контент-провайдерского сервера предлагается большое количество коммерческих решений. Но для построения данной \*платформы\* можно воспользоваться и открытыми решениями. Мы рассмотрим пример построения сервера на базе WAP- и SMS-шлюза Kannel ([www.kannel.org](http://www.kannel.org)). Самым распространенным протоколом для связи с СМС-центром на сегодняшний день является протокол SMPP 3.4 (Short messages point-to-point). Мы предполагаем, что вы уже получили сервисный номер и SMPP-подключение у одного из операторов GSM или CDMA стандарта. В принципе, Kannel поддерживает и работу с рядом GSM-модемов для связи с GSM-сетью, но данная возможность не будет рассмотрена в данной статье.

Необходимые исходные данные:

С оператором обязательно согласовываем использование протокола SMPP версии 3.4 (не 4.0). У провайдера сотовой связи должны быть получены сервисный номер (не обязательно короткий, может быть и федеральный), IP-адрес и порт СМС-центра, login и password, кодировка передаваемых оператором в Ваш адрес сообщений, кодировка принимаемых оператором сообщений, параметры source address ton, source address npi, destination address ton, destination address npi.

Если у оператора СМЦЦ российского производителя, то он может поддерживать кодировку KOI8-R или CP1251. Это упростит Вашу задачу. Мы же рассмотрим более универсальный вариант, когда используется кодировка UCS-2 Big Endian, являющаяся стандартом в том числе и для кириллических шрифтов в Восточной Европе при отправке СМС-сообщений. Данную кодировку СМЦЦ Вашего оператора наверняка поддерживает. Также Вам необходимо узнать у оператора, поддерживает ли его СМЦЦ "склейку" и какое количество СМС позволяет "склеивать", разрешит или нет оператор Вам подписываться каким-либо именем сервиса (как правило, оператор заменяет Вашу подпись на свой вариант, все зависит от Ваших отношений с оператором).

Русифицируем консоль согласно <http://www.obsd.ru/8/?q=node/1172>

Перед непосредственной установкой Kannel нам необходимо выкачать и сложить в рабочую папку (например, /tmp) следующие пакеты: bash, gmake, libiconv, p5-HTML-Parser, p5-MD5, gettext, libxml, p5-Compress-Zlib-1.35, p5-HTML-Tagset, p5-URI, glib2, libghttp, p5-Срипт-SSL, p5-HTTP-GHHTTP, p5-libwww. Напоминаю, что сборка из портов в OpenBSD не приветствуется из соображений безопасности. Начинаем установку пакетов.

```
# cd /tmp
# pkg_add libiconv-1.9.2p1.tgz
# pkg_add gettext-0.10.40p3.tgz
# pkg_add glib2-2.6.4.tgz
# pkg_add libxml-2.6.16p5.tgz
# pkg_add bash-3.0.16p1.tgz
# pkg_add p5-libwww-5.803
```

Переходим непосредственно к установке Kannel. Выкачиваем с сайта [www.kannel.org](http://www.kannel.org) стабильную версию kannel. Мы не рекомендуем использовать девелоперскую версию, поскольку на момент написания статьи в девелоперскую версию были добавлены дополнительные проверки, которые привели к невозможности работы с кириллицей. Чтобы сделать девелоперскую версию рабочей, нам пришлось исключить из исходного кода kannel часть кода. В стабильной версии данные изменения вводить не требуется.

```
# wget -c http://www.kannel.org/download/1.4.0/gateway-1.4.0.tar.gz
# tar -zxvf gateway-1.4.0.tar.gz
# cd gateway-1.4.0.tar.gz
# ./configure --disable-ssl
# /usr/local/bin/gmake
# /usr/local/bin/gmake install
```

Добавляем группу \_kannel и пользователя \_kannel

```
# groupadd _kannel
# useradd -s /sbin/nologin -d /dev/null -g _kannel _kannel
```

Создаем старт-стоповый скрипт /usr/local/sbin/kannelctl следующего содержания:

```
#!/bin/sh

# Configuration file
CONF=/etc/kannel/kannel.conf

# Kannel boxes
BEARERBOX=/usr/local/sbin/bearerbox
SMSBOX=/usr/local/sbin/smsbox
WAPBOX=/usr/local/sbin/wapbox

# Debug level: 0..4
DEBUGLEVEL=0

# Kannel user account
USER=_kannel

# Check that boxes and config are ok
[ -x $BEARERBOX ] || exit 0
[ -x $SMSBOX ] || exit 0
[ -f $CONF ] || exit 0

case "$1" in
start)
echo -n "Starting bearerbox... "
$BEARERBOX -v $DEBUGLEVEL --daemonize --user $USER $CONF &
echo "done"
echo -n
sleep 3

echo -n "Starting smsbox... "
$SMSBOX -v $DEBUGLEVEL --daemonize --user $USER $CONF &
echo "done"
;;
stop)
echo -n "Shutting down kannel... "
/usr/bin/pkill -9 -U $USER
echo "done"
;;
restart)
$0 stop
sleep 3
$0 start
;;
*)
echo "Usage: kannelctl {start|stop|restart}"
exit 1
esac

exit 0
```

Не забываем установить права на запуск скриптов. Создаем каталоги для лог-файлов и конфигурационного файла и приступаем к редактированию конфига.

```
# mkdir /var/log/kannel
# chown _kannel /var/log/kannel
# mkdir /etc/kannel
```

```
# cp /tmp/gateway/doc/examples/kannel.conf /etc/kannel
```

Редактируем файл и приводим его к следующему виду:

```
group = core
admin-port = 13000
admin-password = bar
admin-deny-ip = "*.*.*.*"
admin-allow-ip = "127.0.0.1"
```

IP для администрирования

```
smsbox-port = 13012
wdp-interface-name = "*"
log-file = "/var/log/kannel/bearerbox.log"
box-deny-ip = "*.*.*.*"
box-allow-ip = "127.0.0.1"
```

IP, с которых разрешена отправка СМС

```
group = smsc
smsc = smpp
host = 111.222.333.444
```

IP СМСЦ провайдера

```
*port = 2001
```

порт отправки СМСЦ провайдера

```
receive-port = 2001
```

порт получения СМСЦ провайдера, может быть тем же, что и port\*

```
smsc-username = "smsc_username"
```

выданный провайдером login

```
smsc-password = smsc_password
```

выданный провайдером пароль

```
system-type = "KANNEL"
```

любое имя

```
address-range = ""
source-addr-ton = 0
```

параметр выдан провайдером

```
source-addr-npi = 1
```

параметр выдан провайдером

**dest-addr-ton = 1**

параметр выдан провайдером

**dest-addr-npi = 1**

параметр выдан провайдером

**group = smsbox**

**bearerbox-host = localhost**

Ваш IP для отправки СМС (IP Вашего сервера, на котором будет слушать Kannel)

**sendsms-port = 13013**

Порт на котором слушает Kannel

**sendsms-chars = "0123456789+"**

Допустимые символы в номере абонента

**access-log = "/var/log/kannel/sms-access.log"**

**log-file = "/var/log/kannel/smsbox.log"**

**log-level = 0**

**group = sendsms-user**

**username = your\_user**

используя этот логин можно отправлять СМС с помощью http-запроса

**password = your\_password**

используя этот пароль можно отправлять СМС с помощью http-запроса

**# TEST**

**group = sms-service**

**keyword = default**

**text = "INCORRECT REQUEST"**

**concatenation = true**

разрешена или нет склейка

**max-messages = 3**

максимальное количество склеиваемых СМС

**# HELLO WORLD**

**group = sms-service**

**keyword = 123**

**text = "HELLO WORLD"**

здесь может быть только латиница

**concatenation = true**

разрешена или нет склейка

```
max-messages = 3
```

максимальное количество склеиваемых СМС

```
# HELLO WORLD XML-CGI
```

```
group = sms-service  
keyword = service  
get-url =  
"http://localhost:80/cgi-bin/service.pl?otkogo=%p&keyword=%a&komu=%P&time=%t"
```

```
# HELLO WORLD XML-HTML
```

```
group = sms-service  
keyword = 234  
get-url = "http://localhost:80/test.html"
```

Добавляем в /etc/rc.local следующие строки:

```
# Kannel SMS gateway  
if [ -f /etc/kannel/kannel.conf ]; then  
/usr/local/sbin/kannelctl start  
fi
```

Для отправки кириллических СМС из командной строки нам потребуется небольшая утилита, откомпилированную версию для OpenBSD 3.8 которой можно взять [<http://unixman.ru/bin2hex> <http://unixman.ru/bin2hex>] . Ниже приведен исходный текст на С. Откомпилированную версию программы кладем в /usr/local/bin . Не забываем про права на исполнение.

```
#include  
  
int main()  
{  
int ch;  
  
while ((ch = getchar()) != EOF) {  
if (ch == '\n')  
continue;  
else  
printf("%%02X", ch);  
}  
}
```

Теперь перейдем непосредственно к программе для отправки СМС из командной строки. Создаем файл sendsms.pl в каталоге /usr/local/sbin

следующего содержания:

```
#!/usr/bin/perl -w  
  
use warnings;  
use strict;  
use HTTP::Request;  
use HTTP::Response;  
use LWP;  
  
sub send_sms {  
my ($from, $to, $text) = @_;  
  
$text =~ s/\"/\"/g;
```

```
my $url = 'http://127.0.0.1:13013/cgi-bin/sendsms?username=your_user& \
password=your_password&validity=1440&coding=2&to=%2B_TO_&from=_FROM_&text=_TEXT_';
```

Параметр "validity" - срок жизни СМС в минутах. В течении этого времени СМЦ будет делать попытки доставки абоненту.

```
$text = `echo "$text" | iconv -f koi8-r -t ucs-2be | /usr/local/bin/bin2hex`;
```

```
$url =~ s/_FROM_/$from/;
```

```
$url =~ s/_TO_/$to/;
```

```
$url =~ s/_TEXT_/$text/;
```

```
my $ua = LWP::UserAgent->new;
```

```
my $request = HTTP::Request->new(GET => $url);
```

```
my $response = $ua->request($request);
```

```
}
```

```
Main();
```

```
sub Main {
```

```
if (@ARGV != 2) {
```

```
print "Usage: MSISDN TEXT\n";
```

```
exit(1);
```

```
}
```

```
send_sms("Provider", $ARGV[0], $ARGV[1]);
```

```
}
```

Запускаем Kannel командой:

```
# /usr/local/sbin/kannelctl start
```

Можем проверить связь с СМЦ

```
# tail -f /var/log/kannel/bearerbox.log
```

```
# ^C
```

Логинимся в систему от пользователя с кириллической консолью и отправляем СМС командой:

```
bash-3.0# cd /usr/local/sbin
```

```
bash-3.0# ./sendsms.pl 79XXXXXXXX "Проверка отправки кириллицы"
```

Если все сделано верно, Вам придет на указанный номер СМС-сообщение.

Отправляем с мобильного телефона на Ваш сервисный номер СМС-сообщение с текстом "123" - Вам в ответ должно прийти "HELLO WORLD".

Попробуем отправить СМС с произвольным текстом.

Следующим этапом для интерактивной обработки СМС-запросов абонентов нам потребуется установленный http-сервер с cgi-bin или php, например

apache, установку и настройку www-сервера в этой статье мы рассматривать не будем. Отметим лишь, что последние версии апача распространяются

под неприемлимой для OpenBSD лицензией, а в комплекте OpenBSD присутствуют и другие весьма достойные серверы. Штатный уже установленный апач в

комплекте OpenBSD запускается в чруте. Для работы cgi-bin потребуется копирование всех необходимых апачу библиотек в чрут, но временно для

тестов апач можно запустить не в чруте командой httpd -u . В целях безопасности настраиваем pf Даниэля Хартмайера - разрешаем доступ на порты

kannel и http-сервера только с необходимых нам IP.

Принцип взаимодействия в случае нашего конфигурационного файла следующий - kannel при получении СМС-сообщения с ключевым словом "service"

формирует http-запрос на указанный в конфиге URL <http://localhost:80/cgi-bin/service.pl?otkogo=%p&keyword=%a&komu=%P&time=%t>. В ответ httpd

при обработке скрипта выдает xml-код (ВНИМАНИЕ!) в кодировке UTF-8. В этом случае будет сформировано кириллическое сообщение. В данном случае

скрипту будут переданы параметры `otkogo=(MSISDN абонента)`, `keyword=service`, `komu=(Ваш сервисный номер)`, `time=(время запроса)`.

Для примера ответного XML-сообщения Вы можете создать файл <http://localhost:80/test.html> в кодировке UTF-8 следующего содержания:

```
<?xml version="1.0"?>
<message>
<submit>
<ud>ВАШ ТЕКСТ</ud>
<dc>
<coding>2</coding>
</dc>
</submit>
</message>
```

А, например, данный код

```
<?xml version="1.0"?>
<message>
<submit>
<ud>Если слышна мелодия поддерживается EMS</ud>
<udh>820C8000424547494E3A494D454C4F444590D0A56455253494F4E3A312E300D0A464F524D41543A434C415353312E3
00D0A4D454C4F444593A2A336633663366332363312364332364332364333633172336633663366332363332366332
3663366332A342363332363332363332A332361310D0A454E443A494D454C4F444590D0A</udh>
<dc>
<coding>2</coding>
</dc>
<from>TEST</from>
</message>
```

сформирует EMS-сообщение и телефон проиграет мелодию (на телефонах, поддерживающих EMS. Завесно, но Nokia до сих пор игнорирует этот

стандарт). Информацию о кодировании udh-заголовков и другие справочные материалы по sms можно найти на сайте <http://www.isms.ru>

Данный xml-код может быть находиться не только в статическом файле, но также и может быть сформирован динамическим интерактивным

cgi-приложением или результатом работы php-скрипта.

Kannel обладает большими возможностями. Не забывайте, есть документация на Kannel. Это, конечно, далеко не лучшая документация из всех нами

виденных, но, думаем, данная статья поможет Вам на начальном этапе с ней разобраться.

Использованная литература:

1. Документация OpenBSD 3.8 <http://www.openbsd.org/faq/index.html>
2. Документация Kannel <http://www.kannel.org/download/1.4.0/userguide-1.4.0/userguide.html>
3. Udh код для мелодии взят с сайта <http://http://www.isms.ru>