

Intégration de mot de passe d'invité ISE avec la passerelle SMS basée sur l'exemple de suffixe et de configuration de Kannel

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Configurez](#)

[Avantages de la passerelle SMS](#)

[Schéma de réseau et circulation](#)

[Configurations](#)

[ISE](#)

[Suffixe](#)

[Maildrop avec Mailfilter](#)

[Kannel](#)

[Vérifiez](#)

[ISE](#)

[Suffixe](#)

[Maildrop](#)

[Mailfilter](#)

[Kannel](#)

[Téléphone d'invité](#)

[Dépannez](#)

[ISE](#)

[Informations connexes](#)

Introduction

Ce document décrit comment intégrer des solutions open sources (suffixe, Maildrop, Kannel) avec le Logiciel Cisco Identity Services Engine (ISE) afin de fournir un message du service de messages courts (SMS) aux utilisateurs avec des comptes d'invité.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Cisco ISE et accès invité
- Script de Linux et de shell

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Version 1.2 ou ultérieures de Cisco ISE
- Version 2.10 de suffixe
- Version 2.6.0 de Maildrop
- Version 1.5.0 de Kannel

Remarque: Veuillez être au courant que le suffixe, le Maildrop, et le Kannel sont des solutions open sources, et Cisco ne prend en charge pas ces Produits. Cet exemple de configuration présente simplement comment ISE peut être intégré avec des autres produits afin de fournir une solution de bout en bout.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Informations générales

ISE te permet pour créer l'invité explique l'accès au réseau provisoire, typiquement pour des invités, des visiteurs, des sous-traitants, des consultants, et des clients. De tels comptes sont créés par des utilisateurs de sponsor par l'intermédiaire du portail de sponsor. Quand vous créez le compte, il est possible d'envoyer un mot de passe d'accès dynamique-généré avec un SMS directement au téléphone portable d'utilisateur d'invité.

Cisco ISE peut envoyer ces qualifications par l'intermédiaire de l'email avec le Protocole SMTP (Simple Mail Transfer Protocol) à la passerelle Mail2SMS. Cette passerelle est responsable de la livraison SMS.

Configurez

Avantages de la passerelle SMS

Il y a de plusieurs solutions d'accès Mail2SMS sur le marché. Ils peuvent habituellement recevoir des données avec l'utilisation de différents protocoles, tels que le SMTP, le peer-to-peer court de message (SMPP), le FTP, HTTP (protocole simple d'Access d'objet (SAVON), services Web), et

envoient un message SMS au téléphone portable spécifique.

Il pourrait être le meilleur de construire votre propre passerelle SMS. Il tient compte de :

- Meilleure flexibilité
- La capacité d'établir des règles composées au sujet du routage (basé sur temps, basé sur la politique, basé sur contenu)
- Intégration avec des bases de données locales (par exemple, différentes stratégies de routage pour différents groupes de Répertoire actif)
- Des coûts d'exploitation potentiellement plus à prix réduit (aucun besoin de payer un service externe)
- La possibilité pour utiliser cette solution également pour des alertes de santés générées par ISE et envoyées comme emails

Il pourrait être intéressant d'avoir un déploiement mélangé - une passerelle personnelle SMS qui est également intégrée avec un service externe.

Schéma de réseau et circulation

Voici l'écoulement :

1. L'utilisateur de sponsor crée un compte d'invité avec une notification SMS, et fournit le nombre de téléphone portable pour l'utilisateur. ISE envoie un email au serveur SMTP configuré. L'adresse source (**de**) appartient à un utilisateur spécifique de sponsor, tandis que l'adresse de destination (**à**) est configurée globalement sur ISE (dans cet exemple, **sms@test-cisco.com**). Tous les détails au sujet de l'utilisateur de création récente, tel que le nom d'utilisateur et mot de passe, sont à l'intérieur du corps de cet email.
2. L'email arrive sur le serveur de suffixe, qui est configuré avec le maildrop comme agent de livraison local. Juste avant la livraison au répertoire de maildir d'utilisateur SMS, le maildrop recherche le mailfilter dans le répertoire home pour l'utilisateur. Le script de mailfilter analyse l'email, et si toutes les données nécessaires sont trouvées, il emploie le **wget** afin d'envoyer la requête HTTP GET au smsbox de Kannel. Ce les requêtes HTTP GET contient le message texte avec le nom d'utilisateur et mot de passe, et le nombre de téléphone portable de l'utilisateur. Le smsbox de Kannel est le frontal de Kannel qui est utilisé afin de recevoir toutes les demandes des utilisateurs afin d'envoyer le SMS (pour le passer au bearerbox de Kannel).
3. Le smsbox de Kannel envoie cette demande au bearerbox de Kannel, qui a la responsabilité d'envoyer le SMS.
4. Il pourrait y avoir de plusieurs règles et centres de service de service de messages courts (SMSCs) configurés sur le bearerbox. Cet exemple utilise un serveur externe SMPP. La configuration pour un téléphone portable local-relié est facile et est présentée plus tard.

Chaque module de cette solution (smsbox de suffixe, de Kannel, et bearerbox de Kannel) peut être installé sur un serveur distinct. Pour la simplicité dans cet exemple, il est configuré sur le même serveur.

Configurations

ISE

Terminez-vous ces étapes afin de configurer l'ISE.

1. Configurez l'utilisateur de portail de sponsor. Dans cet exemple, la configuration du par défaut ISE est utilisée, et l'utilisateur est placé dans le groupe de **SponsorAllAccount** :

L'email pour l'utilisateur de sponsor peut être configuré plus tard du portail de sponsor.

2. Afin de pouvoir envoyer des notifications SMS, éditez les privilèges par défaut pour le groupe de **SponsorAllAccount** :

Par défaut, le privilège de l'**envoi SMS** est désactivé.

3. Configurez le serveur SMTP, et assurez-vous que les configurations de DN sont correctes.

Tous les emails de notification sont envoyés à l'hôte de **smtp.test-cisco.com**. ISE n'essaye pas de vérifier les enregistrements MX de DN pour les domaines configurés (ce serveur SMTP est traité comme relais).

4. Personnalisez l'email qui est envoyé comme notification SMS.

5. Configurez l'adresse e-mail de destination, qui est la seule configuration qui n'est pas laissée en tant que par défaut. Toutes les notifications sont envoyées par l'intermédiaire d'un serveur SMTP configuré plus tôt avec pour mettre en place le positionnement comme **sms@test-cisco.com**.

Remarque: Il est possible de configurer ISE afin d'envoyer des notifications vigilantes par l'intermédiaire de l'email. Ceci peut également être intégré avec la solution proposée afin d'envoyer les alertes comme SMS aux téléphones portables. Cisco informe que vous utilisez un fond de placement distinct sur le serveur de suffixe pour ceci (par exemple, **alert@test-cisco.com**).

Suffixe

Le suffixe est un serveur SMTP qui reçoit des emails d'ISE. La configuration par défaut est utilisée

excepté quelques modifications mineures. Terminez-vous ces étapes afin de le configurer.

1. Configurez le suffixe afin d'être la destination locale pour le domaine de **test-cisco.com**. Il est important de configurer également un agent de livraison local : maildrop. Voici les modifications nécessaires dans le main.cf :

```
myhostname = smtp.test-cisco.com
mydomain = test-cisco.com
mydestination = $myhostname, $mydomain, localhost
local_transport = maildrop
```

2. L'étape suivante est de lancer le maildrop dans le master.cf. Changez la ligne correcte dans le master.cf :

```
maildrop unix -      n      -      -      pipe
  flags=DRhu user=vmail argv=/usr/bin/maildrop -d ${user}
```

Puisque c'est un déploiement simple sans domaines virtuels, **{le paramètre d'utilisateur}** est utilisé au lieu du paramètre **{réceptif}** par défaut.

3. Configurez le compte local SMS qui est utilisé afin de recevoir les emails :

```
neptun ~ # useradd sms
neptun ~ # passwd sms
New password:
BAD PASSWORD: it is too simplistic/systematic
Retype new password:
passwd: password updated successfully
neptun ~ # chown -R sms:sms /home/sms/
```

En ce moment, tous les emails devraient être correctement fournis à l'utilisateur SMS. La structure de maildir est créée automatiquement quand elle reçoit d'abord l'email.

Maildrop avec Mailfilter

Juste avant la livraison, le maildrop recherche **.mailfilter** dans le répertoire home pour l'utilisateur. Si ce fichier est trouvé, le script est exécuté. Les privilèges pour le fichier devraient être limités à l'utilisateur seulement :

```
neptun sms # touch /home/sms/.mailfilter
neptun sms # chmod go-rwx /home/sms/.mailfilter
```

Voici le contenu du fichier :

```
# Mailfilter script for parsing ISE SMS messages
# Author: Michal Garcarz at cisco.com
# Date: 1 Dec 2013

#DEFAULT="$HOME/.maildir/"
DATE=`date`
SHELL="/bin/bash"

# Our log file
logfile "/home/sms/maildrop.log"

# Our verbosity in the log file
```

```
VERBOSE="5"
```

```
log "-----SMS MAILFILTER LOG-----"  
log "Email received at: $DATE"
```

```
if (/^Subject:. *Guest.*Text.*Notification.*//)  
{
```

```
    log "Email processed by script sending SMS via Kannel"
```

```
    USERNAME=""
```

```
    PASSWORD=""
```

```
    TO=""
```

```
    if (/^text:Username:(.*)/:b)
```

```
    {
```

```
        log "Username exists $MATCH1"
```

```
        USERNAME=$MATCH1
```

```
    }
```

```
    if (/^text:Password:(.*)/:b)
```

```
    {
```

```
        log "Password exists $MATCH1"
```

```
        PASSWORD=$MATCH1
```

```
    }
```

```
    if (/^to:(.*)/:b)
```

```
    {
```

```
        log "Mobile phone exists $MATCH1"
```

```
        TO=$MATCH1
```

```
    }
```

```
    if ($USERNAME ne "" && $PASSWORD ne "" && $TO ne "")
```

```
    {
```

```
        log "Sending via HTTP to kannel username=$USERNAME password=$PASSWORD to=$TO"
```

```
        DATA="ISE Guest portal Username: $USERNAME Password: $PASSWORD"
```

```
        #also curl can be used instead of wget
```

```
        xfilter "wget -O/dev/null \"http://192.168.112.100:13013/cgi-bin/sendsms?username=  
tester&password=foobar&to=$TO&text=$DATA\" >> /tmp/mailedrop-kannel.log 2>>  
/tmp/mailedrop-kannel.log"
```

```
    }
```

```
    #deliver to maildir (not used since xfilter returns !=0)
```

```
    to $DEFAULT/
```

```
}
```

Le script :

- Vérifie si le sujet est identique comme ce qui est configuré sur l'ISE
- Lit le nom d'utilisateur et mot de passe aux champs pour le corps d'email (le modèle par défaut de l'ISE est utilisé)
- Appelle un programme externe si tous les champs existent : **le wget** afin d'envoyer le HTTP OBTIENNENT au smsbox de Kannel avec tous les paramètres. Notez que des qualifications spécifiques sont utilisées dans l'URL (username=tester&password=foobar). Ce sont les qualifications de l'utilisateur configuré dans Kannel avec les privilèges d'envoyer le SMS.

Il y a deux fichiers journal ici :

- **/home/sms/mailedrop.log** - logs de l'exécution du script
- **/tmp/mailedrop-kannel.log** - logs d'exécution de wget

Kannel

Le smsbox et le bearerbox peuvent être configurés à partir du fichier unique. Cette configuration utilise le serveur externe SMPP pour la livraison. Il est facile de trouver des plusieurs services sur le Web si vous recherchez l'expression de **fournisseur de services de sms de smpp**. La configuration est simple, parce qu'il n'y a aucun besoin de recevoir et conduire des messages SMS. Cette solution est seulement pour envoyer et utilise un fournisseur SMPP.

Voici un extrait de `/etc/kannel/kannel.conf` :

```
#bearerbox
group = core
admin-port = 13000
admin-password = bar
smsbox-port = 13001
log-level = 0
log-file = "/var/log/kannel/kannel.log"
access-log = "/var/log/kannel/access.log"

# SMSC SMPP
group = smsc
smsc = smpp
host = ****.com
port = 1775
smsc-username = 41jt7wi3
smsc-password = *****
system-type =
address-range =

# SMSBOX SETUP
group = smsbox
bearerbox-host = localhost
sendsms-port = 13013
sendsms-chars = "0123456789 +-"
global-sender = 12345
log-file = "/var/log/kannel/smsbox.log"
log-level = 0
access-log = "/var/log/kannel/access.log"
white-list =
black-list =
reply-couldnotfetch =
reply-couldnotrepresent =
reply-requestfailed =
reply-emptymessage =

# SEND-SMS USERS, this credentials has been used in wget script
group = sendsms-user
username = tester
password = foobar
user-deny-ip = "*. *.*.*"
user-allow-ip = "192.168.*.*"

# SMS SERVICE Default
# there should be default always (this is for receiving SMS messages - not used)
group = sms-service
keyword = default
text = "No service specified"
```

Il est possible de relier un téléphone portable par l'intermédiaire de l'USB et de configurer GSM SMSC :

```
group = smsc
```

```
smc = at #type = GSM
smc-id = usb0-modem
my-number = 1234
modemtype = auto #types: wavcom, siemens, siemens-tc35, falcom,
nokiaphone, ericsson
device = /dev/ttyUSB0 #phone device seen on server
```

À la plupart des téléphones, on l'exige également pour lancer la fonctionnalité de modem ; par exemple, dans la version 2.2 et ultérieures androïde, il est activé dans les configurations/attachement et le Hotspot/USB portatif attachant.

Souvenez-vous pour exécuter le bearerbox et le smsbox. Voici un exemple :

```
neptun ~ # /etc/init.d/kannel-bearerbox start
* Starting kannel bearerbox ... [ ok ]
neptun ~ # /etc/init.d/kannel-smsbox start
* Starting kannel smsbox ... [ ok ]
```

```
neptun ~ # netstat -atcpn
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 0.0.0.0:13013 0.0.0.0:* LISTEN 24170/smsbox
tcp 0 0 0.0.0.0:13000 0.0.0.0:* LISTEN 24151/bearerbox
tcp 0 0 0.0.0.0:13001 0.0.0.0:* LISTEN 24151/bearerbox
```

Bearerbox doit avoir au moins un SMSC configuré afin de commencer.

Vérifiez

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

ISE

Le port par défaut pour le portail de sponsor est utilisé (8443). Le sponsor se connecte dans <https://ise.test-cisco.com:8443/sponsorportal/>.

Assurez-vous que le sponsor fait assigner une adresse e-mail dans **mes configurations** :

Créez le compte d'invité avec une notification SMS :

Vous recevez la confirmation que le compte d'invité a été avec succès créé :

ISE devrait envoyer un email au serveur SMTP configuré.

Suffixe

Le serveur SMTP reçoit le message et emploie le maildrop afin de le fournir au compte local (sms@test-cisco.com). Voici un extrait de `/var/log/messages` :

```
Nov 30 22:39:47 neptun postfix/smtpd[18460]: connect from unknown[192.168.112.1]
Nov 30 22:39:47 neptun postfix/smtpd[18460]: 2B36030B32: client=unknown
[192.168.112.1]
Nov 30 22:39:47 neptun postfix/cleanup[18463]: 2B36030B32: message-id=
<563762958.941385847586377.JavaMail.root@ise2>
```

```
Nov 30 22:39:47 neptun postfix/qmgr[32658]: 2B36030B32: from=<sponsor@test-cisco.com>,
size=689, nrcpt=1 (queue active)
Nov 30 22:39:47 neptun postfix/pipe[18464]: 2B36030B32: to=<sms@test-cisco.com>,
relay=maildrop, delay=0.18, delays=0.14/0/0/0.04, dsn=2.0.0, status=sent (delivered
via maildrop service)
```

Maildrop

Avant que vous envoyiez l'email au SMS, le répertoire de maildir exécute `/home/sms/.mailfilter`, qui exécute une action spécifique.

Voici un extrait du `/home/sms/maildrop.log` :

```
-----SMS MAILFILTER LOG-----
Email received at: Sat Nov 30 22:39:47 CET 2013
Email processed by script sending SMS via Kannel
Username exists jsmith02
Password exists t6ub79_6r
Mobile phone exists 4850xxxxxxx
Sending via HTTP to kannel username= jsmith02 password= t6ub79_6r to=4850xxxxxxx
```

Mailfilter

Le script de mailfilter lit toutes les données et exécute le `xfilter`, qui appelle le `wget` afin de passer tous les paramètres à Kannel.

Voici un extrait de `/tmp/maildrop-kannel.log` :

```
--2013-11-30 22:39:47-- http://192.168.112.100:13013/cgi-bin/sendsms?username=
tester&password=foobar&to=4850xxxxxxx&text=ISE%20Guest%20portal%20Username:
%20%20jsmith02%20Password:%20%20t6ub79_6r
Connecting to 192.168.112.100:13013... connected.
HTTP request sent, awaiting response... 202 Accepted
Length: 24 [text/html]
Saving to: `/dev/null'
```

OK

100% 1.14M=0s

```
2013-11-30 22:39:47 (1.14 MB/s) - `/dev/null' saved [24/24]
```

La requête HTTP GET est reçue.

Kannel

Les signaler de smsbox de Kannel qu'il a reçu une demande de HTTP du `wget` et lui ont envoyé cette demande au bearerbox afin de livrer le SMS.

Voici un extrait de `/var/log/kannel/smsbox.log` :

```
2013-11-30 22:39:47 [18184] [5] INFO: smsbox: Got HTTP request </cgi-bin/sendsms>
from <192.168.112.100>
2013-11-30 22:39:47 [18184] [5] INFO: sendsms used by <tester>
2013-11-30 22:39:47 [18184] [5] INFO: sendsms sender:<tester:12345>
(192.168.112.100) to:<4850xxxxxxx> msg:<ISE Guest portal Username:
jsmith02 Password: t6ub79_6r>
2013-11-30 22:39:47 [18184] [5] DEBUG: Stored UUID fd508632-9408-49e1-9eda-
```

3ce8d4b939d4

```
2013-11-30 22:39:47 [18184] [5] DEBUG: message length 57, sending 1 messages
2013-11-30 22:39:47 [18184] [5] DEBUG: Status: 202 Answer: <Sent.>
2013-11-30 22:39:47 [18184] [5] DEBUG: Delayed reply - wait for bearerbox
2013-11-30 22:39:47 [18184] [0] DEBUG: Got ACK (0) of fd508632-9408-49e1-9eda-
3ce8d4b939d4
```

Le bearerbox de Kannel se connecte au serveur du distant SMPP et envoie le message avec succès.

Voici un extrait de `/var/log/kannel/kannel.log` :

```
2013-11-30 22:39:47 [18165] [8] DEBUG: boxc_receiver: sms received
2013-11-30 22:39:47 [18165] [8] DEBUG: send_msg: sending msg to box: <127.0.0.1>
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP[SMPP:*****.com:1775/0:41jt7wi3:]:
throughput (0.00,0.00)
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP[SMPP:*****.com:1775/0:41jt7wi3:]:
Sending PDU:
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP PDU 0x2056bf0 dump:
2013-11-30 22:39:47 [18165] [6] DEBUG:   type_name: submit_sm
2013-11-30 22:39:47 [18165] [6] DEBUG:   command_id: 4 = 0x00000004
2013-11-30 22:39:47 [18165] [6] DEBUG:   command_status: 0 = 0x00000000
2013-11-30 22:39:47 [18165] [6] DEBUG:   sequence_number: 5 = 0x00000005
2013-11-30 22:39:47 [18165] [6] DEBUG:   service_type: NULL
2013-11-30 22:39:47 [18165] [6] DEBUG:   source_addr_ton: 2 = 0x00000002
2013-11-30 22:39:47 [18165] [6] DEBUG:   source_addr_npi: 1 = 0x00000001
2013-11-30 22:39:47 [18165] [6] DEBUG:   source_addr: "12345"
2013-11-30 22:39:47 [18165] [6] DEBUG:   dest_addr_ton: 2 = 0x00000002
2013-11-30 22:39:47 [18165] [6] DEBUG:   dest_addr_npi: 1 = 0x00000001
2013-11-30 22:39:47 [18165] [6] DEBUG:   destination_addr: "4850xxxxxxxx"
2013-11-30 22:39:47 [18165] [6] DEBUG:   esm_class: 3 = 0x00000003
2013-11-30 22:39:47 [18165] [6] DEBUG:   protocol_id: 0 = 0x00000000
2013-11-30 22:39:47 [18165] [6] DEBUG:   priority_flag: 0 = 0x00000000
2013-11-30 22:39:47 [18165] [6] DEBUG:   schedule_delivery_time: NULL
2013-11-30 22:39:47 [18165] [6] DEBUG:   validity_period: NULL
2013-11-30 22:39:47 [18165] [6] DEBUG:   registered_delivery: 0 = 0x00000000
2013-11-30 22:39:47 [18165] [6] DEBUG:   replace_if_present_flag: 0 = 0x00000000
2013-11-30 22:39:47 [18165] [6] DEBUG:   data_coding: 0 = 0x00000000
2013-11-30 22:39:47 [18165] [6] DEBUG:   sm_default_msg_id: 0 = 0x00000000
2013-11-30 22:39:47 [18165] [6] DEBUG:   sm_length: 57 = 0x00000039
2013-11-30 22:39:47 [18165] [6] DEBUG:   short_message:
2013-11-30 22:39:47 [18165] [6] DEBUG:     Octet string at 0x205ec70:
2013-11-30 22:39:47 [18165] [6] DEBUG:       len: 57
2013-11-30 22:39:47 [18165] [6] DEBUG:       size: 58
2013-11-30 22:39:47 [18165] [6] DEBUG:       immutable: 0
2013-11-30 22:39:47 [18165] [6] DEBUG:       data&colon; 49 53 45 20 47 75 65 73
74 20 70 6f 72 74 61 6c   ISE Guest portal
2013-11-30 22:39:47 [18165] [6] DEBUG:       data&colon; 20 55 73 65 72 6e 61 6d
65 3a 20 20 6a 73 6d 69   Username: jsmi
2013-11-30 22:39:47 [18165] [6] DEBUG:       data&colon; 74 68 30 32 20 50 61 73
73 77 6f 72 64 3a 20 20   th02 Password:
2013-11-30 22:39:47 [18165] [6] DEBUG:       data&colon; 74 36 75 62 37 39 11 36
72   t6ub79.6r
2013-11-30 22:39:47 [18165] [6] DEBUG:     Octet string dump ends.
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP PDU dump ends.
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP[SMPP:*****.com:1775/0:41jt7wi3:]:
throughput (1.00,0.00)
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP[SMPP:*****.com:1775/0:41jt7wi3:]:
throughput (1.00,0.00)
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP[SMPP:*****.com:1775/0:41jt7wi3:]:
Got PDU:
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP PDU 0x2056bf0 dump:
2013-11-30 22:39:47 [18165] [6] DEBUG:   type_name: submit_sm_resp
2013-11-30 22:39:47 [18165] [6] DEBUG:   command_id: 2147483652 = 0x80000004
```

```
2013-11-30 22:39:47 [18165] [6] DEBUG:   command_status: 0 = 0x00000000
2013-11-30 22:39:47 [18165] [6] DEBUG:   sequence_number: 5 = 0x00000005
2013-11-30 22:39:47 [18165] [6] DEBUG:   message_id: "4128473611307259"
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP PDU dump ends.
2013-11-30 22:39:47 [18165] [6] DEBUG: SMPP[SMPP:*****.com:1775/0:4ljt7wi3:]:
throughput (1.00,0.00)
```

Notez que l'adresse source est placée en tant que **12345**. Cette configuration n'importe pas. Les réécritures externes de serveur SMPP qui évaluent. Il est possible d'acheter le service supplémentaire afin de pour être présenté différemment.

Téléphone d'invité

L'utilisateur d'invité reçoit un SMS :

Dépannez

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

ISE

Vous pourriez rencontrer cette erreur quand vous créez un compte d'invité : Incapable d'envoyer un message texte aux utilisateurs suivants d'invité : xxxx. Vous devez ajouter votre adresse e-mail à la page de configurations. Si vous recevez ce message d'erreur, vérifiez l'adresse e-mail de sponsor.

[Informations connexes](#)

- [Logiciel Cisco Identity Services Engine - Guide des utilisateurs 1.2](#)
- [Documentation de Kannel](#)
- [Documentation de suffixe](#)
- [Support et documentation techniques - Cisco Systems](#)