

CommuniGate

Integrating with Communigate Pro

There are a number of different way to integrate [SpamAssassin](#) with Communigate Pro. Note that some of these are pretty old and haven't been modified recently.

- CGPSA: <http://www.tffenterprises.com/cgpsa/> (should work on both Linux and Windows)
- CGPAV: <http://program.farit.ru/doc/cgpav.html>
- sa-cgp: <ftp://ftp.vsu.ru/pub/apps/CGP/sa-cgp.c>
- scanspam.sh (Linux only see below)
- cgSpamAssassin: <https://github.com/bret-miller/cgSpamAssassin> (should work on both Linux and Windows)

scanspam.sh

Script to call Spam Assassin from [CommuniGate](#) Pro Rules v1.1 Ed Fang If you have any improvements, drop me a line at edfang@visi.net Thanks for Stefan Seiz for the original template

Script will call [SpamAssassin](#) spamc from [CommuniGate](#). Since SA modifies the actual message, it must be re-delivered via the Submitted directory. Sneaky Header Tagging is used to prevent the message from being scanned again, and/or being caught in a loop is spamd fails.

Source code

```
#!/bin/sh

#####
# Script to call Spam Assassin from CommuniGate Pro Rules v1.1 Ed Fang
# If you have any improvements, drop me a line at edfang@visi.net
# Thanks for Stefan Seiz for the original template
#
# Script will call SpamAssassin spamc from CommuniGate. Since SA modifies
# the actual message, it must be re-delivered via the Submitted directory.
# Sneaky Header Tagging is used to prevent the message from being scanned
# again, and/or being caught in a loop is spamd fails.
#
# DATA
# Communicate Pro Rules
# Any Recipient in *@domain.com
# Message Size less than 32768
# Header Field is not X-Spam-Status*
#
# ACTION
# Execute [STDERR] [FILE] [RETPATH] [RCPT] /var/CommuniGate/scanspam.sh
# Discard
#
# You must discard the message as the script will re-submit the marked message through the
# Submitted directory of CommuniGate. If you don't, you'll get every message
# twice - once scanned, once clear through. (which is a good way to test
# before actually discarding the message. Stalker has asked that messages
# not be modified in the Queue directory, so it's just safer to play by their
# rules.
#
# 1.1 - Changed to run using gawk. Highly advised as awk will occasionally
# crap out with tremendously long html lines.
#
# Note: The global variables aren't used all over the place, so I'll clean that
# up in a later version. I think there might be a more efficient awk script
# so any other awk gurus out there might be able to give me a better awk
# than the one I have (which skips everything until it matches Received
# and then to the end of the file.
#
#####
#### START user configurable variables
# CGate base directory myCgate="/var/CommuniGate" myLogFile="/var/CommuniGate/spam-result.out"

#### END user configurable variables
#Comment out following two lines if you don't want to echo the passed variable info to
#the spam-result.out file.
```

```

myDate=`date +%Y-%m-%d\ %H:%M:%S`
echo "Date $myDate " $@ >> $myLogFile

#Get the fileid of the message file
QueuePath=$2
NewFile=`basename $QueuePath`.tmp'
FinalFile=$NewFile.sub'

# Formulate return-path and Envelope-To addresses from command line args.
# shift out the first 3 arguments, make sure one > to create a new file shift 3

# If you are using Spamprep, the following line should be commented out
echo "Return-Path:" $1 > /var/CommuniGate/Submitted/$NewFile

# If you are using Spamprep, the following line should be uncommented
# ReturnPath=$1

# shift out 5 command arguments. and start appending shift 2
Envelope=$1
shift
while [ $# -gt 0 ] do

    . Envelope=$Envelope', '$1 shift

done

# Formulate the envelope Header file.

# If you are using Spamprep, the following line should be commented out
echo "Envelope-To: " $Envelope >> $myCgate/Submitted/$NewFile

# Append an X-Spam header in there. This is very important. Without this
# tag, if spamc fails to call spamd, it will send the message back without
# a tag and your message will fall into an UGLY loop. This alleviates that.

# If you are using Spamprep, the following line should be commented out
echo "X-Spam-Status: Scanner Called" >> $myCgate/Submitted/$NewFile

# strip out CommuniGate stuff for SpamAssassin first
# using awk, and then send to spamc.

# If you are using Spamprep, the following line should be commented out
gawk '/Received/, /\n/' $myCgate/$QueuePath | /usr/local/bin/spamc -d 127.0.0.1 -f >> /var/CommuniGate/Submitted/$NewFile

# If you are using Spamprep, the following line should be uncommented
#/var/CommuniGate/spam/spamprep "$myCgate/$QueuePath" "$ReturnPath" "$Envelope" | /usr/bin/spamc -d 127.0.0.1 -f >> "$myCgate/Submitted/$NewFile"

#Now submit the file by renameing it to .sub
mv /var/CommuniGate/Submitted/$NewFile /var/CommuniGate/Submitted/$FinalFile

# done.
exit 0;

```