

Proxying Emails to Your Local Mail Relay Server

There may be times where you may need to send emails from a machine, that can not do so directly nor through your local SMTP relay and you have no easy way out for mending the causes. Let's examine some scenarios that would allow you to work around the problem.

As far as I know SMTP protocol does not support proxy itself but I can think of several ways to work around the problem. Here are some scenarios with examples on how you could go about working around the problem.

Throughout the scenarios we will be referring to 3 machines:

- Machine A with IP address 192.168.0.2 (cannot send mails directly via internet and cannot relay on Local Mail Relay)
- Machine B with IP address 192.168.1.2
- Local Mail Relay (LMR) with IP address 192.168.2.2

Scenario 1

- Machine A cannot relay on any local or internet MTA
- Machine A can reach Machine B on some unused port (1234 in the example below)
- Machine B can relay through LMR
- Machine B can reach LMR's port 25

We can use nc on Machine B to proxy mail from A to LMR. In this example it will appear to LMR as if it was Machine B that is sending the email.

On machine B

```
mkfifo /tmp/f
nc -l -p 1234 -s 192.168.1.2 < /tmp/f | nc 192.168.2.2 25 >> /tmp/f
```



The above command is a dirty way to get the output from LMR back to A. See NOTES below if it is not clear to you how it works.

On machine A

```
telnet 192.168.1.2 1234
Trying 192.168.1.2 ...
Connected to 192.168.1.2.
Escape character is '^]'.
220 smtp.localnet.com ESMTP Postfix
helo cicciobello
250 smtp.localnet.co
```

```
MAIL FROM:me@localnet.com
250 2.1.0 Ok
RCPT TO:dude@localnet.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
FROM:me@localnet.com
TO:dude@localnet.com
SUBJECT:test sending email from A trough B into smtp.localnet.com
If you get this it works.
.
250 2.0.0 Ok: queued as 1B1189AC
quit
221 2.0.0 Bye
exit
Connection closed by foreign host.
```

Scenario 2

- Machine A cannot reach LMR's port 25
- Machine A is allowed to relay thought LMR but firewall is preventing it from reaching port 25 on LMR
- Machine A can reach the LMR but only on port 22.

We use a ssh tunnel between A and LMR to do the job for us. In this example we are bypassing firewall by using ssh to forward traffic trough a encrypted tunnel. For this to work properly the ssh client that is going to be forwarding needs to be executed by an administrative account.



This time we will be working exclusively on machine A

```
ssh -2 -D localhost:1024 -N 192.168.2.2 &
curl telnet://smtp.localnet.com:25 -p --socks4 localhost:1024
220 merak.invallee.it ESMTP Postfix
helo testme
250 smtp.localnet.co
MAIL FROM:me@localnet.com
250 2.1.0 Ok
RCPT TO:dude@localnet.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
FROM:me@localnet.com
TO:dude@localnet.com
SUBJECT:test using ssh tunnel
if you get this sending email trough ssh tunnel worked
.
250 2.0.0 Ok: queued as 29C549A7
quit
```

```
221 2.0.0 Bye
exit
```

Scenario 3

- Machine B can relay emails through LMR
- Machine B cannot reach LMR's port 25
- Machine B can access LMR via ssh
- Machine A has access to Machine B on whatever port we choose to let ssh listen on for forwarding (1024 in the example).

This is a bit like Scenario 1 but we will be using ssh to forward stuff to LMR and Machine B needs not be able to reach LMR's port 25

We can then start the ssh socks proxy on Machine B listening on a port Machine A can access:

On machine B with a privileged user

```
ssh -2 -D 192.168.1.2:1024 -N 192.168.2.2 &
```

On machine A

```
curl telnet://smtp.localnet.com:25 -p --socks4 192.168.1.2:1024
220 merak.invallee.it ESMTP Postfix
helo testme
250 smtp.localnet.co
MAIL FROM:me@localnet.com
250 2.1.0 Ok
RCPT TO:dude@localnet.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
FROM:me@localnet.com
TO:dude@localnet.com
SUBJECT:test using ssh tunnel
if you get this sending email through ssh tunnel worked
.
250 2.0.0 Ok: queued as 29C549A7
quit
221 2.0.0 Bye
exit
```

Scenario 4

- Machine A cannot reach LMR's port 25
- Machine A is not allowed to relay through LMR
- Machine A cannot proxy on any other machine that is allowed to relay through LMR
- Machine A can reach the LMR but only on port 22.

We can still get the whole command to run on the LMR via ssh:



This time we will be working exclusively on machine A

```
cat << EOF > send_this_mail
echo "helo testme
MAIL FROM:me@localnet.com
RCPT TO:dude@localnet.com
DATA
FROM:me@localnet.com
TO:dude@localnet.com
SUBJECT:test using ssh on remote server
it worked
.
quit
exit
" | curl telnet://localhost:25
EOF
cat send_this_mail |ssh 192.168.2.2
```

NOTES

The command we used in Scenario 1 is probably a good example of bad unixcraft, let me give a brief explanation on how it works. Let's momentarily get rid of the odd use of the fifo by just using this:

```
nc -l -p 1234 -s 192.168.1.2 | nc 192.168.2.2 25
```

If we repeat rest of Scenario 1 we would see the output from LMR's MTA show up on Machine B and no feedback at all making it to Machine A. What we want to do is feed the output of "nc 192.168.2.2 25" (that connects to LMR) back into the nc that is in listening mode so that it can make it back to Machine A. The pipe itself is not bidirectional so we need some unixcraft to work around the problem: we create a fifo and have the output of "nc 192.168.2.2 25" fed into it

```
nc 192.168.2.2 25 >> /tmp/f
```

and we feed the other end of the fifo into the other nc that is listening

```
cat /tmp/f | nc -l -p 1234 -s 192.168.1.2
```

so now the whole command line looks like this:

```
cat /tmp/f | nc -l -p 1234 -s 192.168.1.2 | nc 192.168.2.2 25 >> /tmp/f
```

it is possible to do the same thing without having to use cat, it is a little neater but still bad unixcraft

```
nc -l -p 1234 -s 192.168.1.2 < /tmp/f | nc 192.168.2.2 25 >> /tmp/f
```

Sources

Originally written by [Louigi600](#)

[howtos](#), [SMTP](#), [proxy](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

https://docs.slackware.com/howtos:misc:proxying_smtp

Last update: **2016/06/20 07:28 (UTC)**

