# Using Condor effectively in the presence of Personal Firewalls

John Kewley

October 7, 2004

## Summary

## 1 Problem

In a Pool comprising personal workstations there may be individual firewalls on the machines in that pool.

For condor to be able to send jobs to such a machine, the following must be done:

- Open ports 9614, 9618 and an arbitrary port range (say 65000-65255) in that machine's firewall to EVERY (either directly or indirectly using subnet entries) machine in that pool!
- Restrict condor on that machine to use the above mentioned port range

So far, so good (although the "EVERY" might worry some people).

The problem comes when another machine is added which is not included in the EVERY clause above, ie it is not individualy named nor is it implied by one of the permitted subnets.

If such a machine submits a job to the pool and condor matches it with a firewalled machine, the job will keep timing out when the firewall gets in the way. While it is timing out, querying the job queue can take some time, also when eventually it gives up, other jobs may instead be sent to the same firewalled machine. It is important then that either all machines are updated frequently (either an administration nightmare, or a worrying automatic update to your security mechanisms) or we need to find a way to prevent jobs on new machines going to the firewalled resources.

## 2 Possible Solutions

- Every time a new machine is added, send an email to owners of machines with firewalls asking them to check their settings and update if neccessary. In the interim, jobs getting matched for those machines will get sent in error to that machine, and will clog up the system by timing out and having to be resent elsewhere. When a pool is being steadily expanded, this could cause a lot of hassle for the resource owners havign to keep changing their firewalls.

- Restrict the machines that can submit jobs and therefore that need adding to the firewall entries. This could be a bit restrictive, and may be a bottleneck in the future. It may also place a restriction on the firewalled machine that it cannot submit jobs.
- Have a cron job which periodically checks the central nodes for condor nodes in the pool that will need adding to its firewall. Most Linuxes seem to be using iptables, but there are other firewalls around, in particular on the WIN boxes, so any script which was to translate these for inclusion in the firewall tables would have to be multi-lingual. Also, automatic updating of firewall settings may not be accepted by the people who are concered enough about security to have a workstation firewall in the first place.
- Allow machines with firewalls to update the list of machines and/or subnets that they allow access from "in their own time". They could update their tables when they felt like it. In the meantime, jobs from machines outside this range would need a mechanism in place so that they could avoid sending jobs to those machines that had not yet opened their firewalls to them.

Of these, I believe, the final one seems preferable. As there is no current support in Condor for this, how can it be achieved?

It needs a bit of rigour (and macro magic) to get a reasonable model:

1. All machines with firewalls add the following entries to their `condor_config.local` file:

   ```
   HAS_FIREWALL = TRUE
   STARTD_EXPRS = HAS_FIREWALL
   ```

   Those without firewalls can also add

   ```
   HAS_FIREWALL = FALSE
   STARTD_EXPRS = HAS_FIREWALL
   ```

   but that is not essential (because of condor ClassAd's fuzzy matching.

2. For each subnet that is opened, in addition to the firewall changes, add a line to the `condor_config.local` as follows:

   ```
   FW_ALLOWS_113 = TRUE
   ```

   and add this new Macro to the `STARTD_EXPRS` line.

   ```
   STARTD_EXPRS = HAS_FIREWALL, FW_ALLOWS_113
   ```

   [Note that it would be nice to utilise the Subnet value that is available through condor, but macro names cannot contain '.'] [For subnets with only a few users, we could also have machines as well as subnets named - `FW_ALLOWS_rjavig6`, for instance - both machine AND subnet would need to be referred to in the user's submit file. This should parallel the firewall setup: if you want less hassle, go for the subnet option; if you want max security, name each machine separately in both firewall file and config file]

3. Now, the user's submit file needs to be able to utilise these macros.

   This can be setup in the `condor_config.local` of the submitting machine by firstly defining the subnet and machine name.

   ```
       MY_SUBNET = 113
       MY_HOST = condor
   ```

   (remembering that the characters for `MY_HOST` have to be from the appropriate subset for macro names)

   Then define `ADDITIONAL_REQUIREMENTS` which are automatically added to the end of all user's `REQUIREMENTS` for that machine as follows:

   ```
       ADD_REQUIREMENTS = \
   ((HAS_FIREWALL =!= TRUE) || \
    (FW_ALLOWS_$(MY_HOST) == TRUE) || \
    (FW_ALLOWS_$(MY_SUBNET) == TRUE))
   ```

   [Note the fuzzy match for `HAS_FIREWALL`, also the use of `MY_SUBNET` and `MY_HOST` to generate ClassAd names]

So, if a new user comes along, he gets an initial set of machines to play with (those with no firewalls or which already include his subnet) and (assuming the owners of the firewalled machines are informed), other machines when appropriate ports are opened to them, and corresponding macros defined.