6.6.7 Delimiting time via program invocations

The TIME_DELIM result type is intended to identify some program whose invocation times will be used to create a set of time ranges. These results, like those from LOG_RANGE differ from other result types in that they define ranges between events. For example, a CONTAINS result set from stdout files would have timestamps reflecting the corresponding program start and stop time, while a TIME_DELIM result would have timestamps reflecting the periods **between** invocations of the program named in the directive.

Consider a lab that directs students to alter iptables on a component. The student is required to demonstrate a desired iptables configuration by running nmap on various other components. The instructor wants to confirm that some set of expected stdout from nmap running on different components all occurred within a single configuration of iptables, delimited by the running of the iptables command. In other words, the student cannot succeed by altering iptables between invocations of nmap on different components.

Note, that to be generally useful, we do not wish to simply look for invocations of iptables by the student. For example, using the command to view the configuration does not represent a change to the configuration. Also, the iptables may be called from a script, e.g., rc.local, and our typical use of stdout files would not see the running of iptables. It is therefore suggested that TIME_DELIM results be tied to files created as an effect of notify events described in 6.1.6. In this example, the notify event would be execution of /sbin/iptables, and the notify_cb.sh script would determine if a change were being made to the configuration.

Then, if the lab results.config were:

```
iptables = firewall:iptables : TIME_DELIM
_remote_nmap_443 = remote_ws:nmap.stdout : CONTAINS : 443/tcp open  https
_remote_nmap_sql = remote_ws:nmap.stdout : CONTAINS : 3306/tcp open  mysql
_local_nmap_443 = ws1:nmap.stdout : CONTAINS : 443/tcp open  https
_local_nmap_sql = ws1:nmap.stdout : CONTAINS : 3306/tcp open  mysql
```

The iptables result set would then include up to N+1 timestamped instances, where N is the quantity of times that iptables was executed to change the configuraion. The first possible timestamp would have a starting time of zero and an ending time of the very first consequential invocation of iptables. The nmap results would each have timestamps corresponding to their times of execution. Note the nmap results include results from two different computers, ws1 and remote_ws.

A goals.config file of:

```
remote_nmap_443 = time_during : _remote_nmap_443 : iptables
remote_nmap_sql = time_during : _remote_nmap_sql : iptables
local_nmap_443 = time_during : _local_nmap_443 : iptables
local_nmap_sql = time_during : _local_nmap_sql : iptables
remote_correct = boolean : ((remote_nmap_443 and_not remote_nmap_sql) \
                            and local_nmap_443 and local_nmap_sql)
```

would generate sets of nmap goals with timestamp ranges corresponding to the iptables results. The remote_correct boolean expression could then be read as: "Was there any single iptables configuration during which the student used nmap to demonstrate that:

- The remote workstation could reach the HTTPS port but not the SQL port, and,
- The local workstation could reach the HTTPS port and the SQL port.

The file identifiers for TIME_DELIM commands can be lists of container:file pairs separated by semicolons. This is useful when configuration changes are delimited by modifications made on more than one component or by more than one program