report. On the other hand, when the student runs `checkwork`, the context is clear and we can provide feedback to the student about the current state of the system relative to the goals. Now the questions are better formed, e.g., does John currently have access to the expected table columns?.

The above discussion is not intended to dissuade lab designers from informing instructors about partial success. If goals can be defined to show the student was able to provide most of the desired access controls though unable to enforce the entire policy, that is to be encouraged. But that can also be hard to do. It is often far easier to provide the student with information about partial goal achievement because the context is *now*.

6.6 Assessment examples

The following examples illustrate some typical assessment operations as they would be defined in the results.config and goals.config files.

6.6.1 Did a program output an expected answer?

Often, the easiest approach to such an assement is to simply use a `FILE_REGEX` field_type within the results.config – and not bother with the goals.config.

```
got_x = *.stdout : FILE_REGEX : X is:.*347
```

The lab goals will include a boolean named `got_x`, which will be true if any stdout file contained a string matching that REGEX.

6.6.2 Do artifact files contain one of two specific strings?

Consider the labs/formatstring/instr_config/results.config file for a few examples. The first non-comment line defines a result having the symbolic name "_crash_sig":

```
_crash_sig = vul_prog.stdout : CONTAINS : program exit, segmentation
_crash_smash = vul_prog.stdout : CONTAINS : *** stack smashing detected
```

This result is TRUE for each timestamped stdout file resulting from running the vul_prog program in which the file contains the string "program exit, segmentation". The goals.config includes this goal:

```
crash =  boolean : ( _crash_smash or  _crash_sig )
```

The value of the crash goal is TRUE if either result was ever true. Use of the `count_greater` operator in the above example would also provide the desired assessment. Note that the boolean operator only assesses values within timestamped sets. For example, if the result values came from different program outputs, then they may not be within the same timestamp, and thus would not compare. In such a case, the `count_greater` operator should be used.

6.6.3 Compare value of a field from a selected line in an artifact file

Again refrence the labs/formatstring/instr_config/results.config file. The third non-comment line defines a result having the symbolic name "origsecret1value":

```
origsecret1value = vul_prog.stdout : 6 : STARTSWITH : The original secrets:
newsecret1value = vul_prog.stdout : 6 : STARTSWITH : The new secrets:
```