TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

New Path w/ Commandline Arguments Filtering

Robert McLay

March 16, 2023

XALT: Outline

- Based on last month's presentation
- I have implemented "Approach 2"
- Namely a built-in pattern matching
- It is not perfect
- But it will work well with python
- It is currently available under the testing branch

Some issue to deal with

- Many commands take command line argument
- This includes python
- Below is a list of *some* of them (via zsh tab completion)
- Python: no options with values
- But some programs may take options with values

```
% python3 -<tab>
option
-B   -- don't write .py[co] files on import
-E   -- ignore PYTHON* environment variables (such as PYTHONPATH)
-I   -- isolate Python from the user's environment
-O   -- optimize generated bytecode slightly
-OO  -- remove doc-strings in addition to the -O optimizations
-x   -- skip first line of source, allowing use of non-Unix forms of #!cmd
```

TACC

Approach 2: XALT does the filtering for you

- This consists of two parts in your site Config.py file
- Changes to path_patterns
- Add in a new group of patterns: path_arg_patterns

Adding the CUSTOM tag to path_patterns

```
path_patterns = [
    ['CUSTOM',  r'.*\/python[0-9][^/][^/]*'],
    ...
]
```

- Only paths that have the “CUSTOM” tag will get further filtering
- A site can have as many “CUSTOM” tags as they like

The new **`path_arg_pattern`**

```
path_arg_patterns = [
  ['SKIP', r'.*\/python[0-9][^/;][^/;]*;.*\/share\/.*'],
  ['PKGS', r'.*\/python[0-9][^/;][^/;]*;.*\/data\/.*'],
  ['PKGS', r'.*\/python[0-9][^/;][^/;]*;'],
]
```

- The pattern is path + “;” + arg as shown above
  1. SKIP any python scripts that have /share/ as part of the path
  2. KEEP any python scripts that have /data/ as part of the
  3. KEEP any python scripts that have neither of the above
- Note the change in how the executable pattern is written!

Rules

- For “CUSTOM” tags the arguments are each processed
- Arguments that start with minus [-] are ignored
- All other arguments would be abspath and checked for existance before being run through path_arg_patterns
- Not perfect but reasonably safe

Could filter on option arguments

```
['CONTINUE, r'.*\/python[0-9][^/;][^/;]*;-.*'],
```

- Ignoring options could be under Site control
- With the pattern above.
- I don’t this supporting this would be useful.

Final patterns

```
['PKGS', r'.*\/python[0-9][^/;][^/;]*;'],
```

- This pattern is there to handle when things do not match
- If this pattern is not provided then the default final pattern is a SKIP

False matching

```
$ python -z /my/path/share/foo /my/path/data/hello.py
```

```
or
```

```
$ python -z /my/path/data/foo /my/path/share/hello.py
```

- Suppose you have either of the above command lines
- The first one would be SKIP'ed
- The second one would be KEEP'ed
- I see no general way to get this to work perfectly
- *This is NOT a current issue with python*
- See "feature" described later

a SKIP match

```
$ XALT_TRACING=run python /my/path/share/hello_world.py

    track_executable():
    -> arg: 1: value: "../../share/hello_world.py"
    -> pattern: "/usr/bin/python3.10;/my/path/share/hello_world.py",
       track_executable token: 3: SKIP
```

- This is what happens when a pattern matches a SKIP
- This can be shown when XALT_TRACING=run

a PKGS match

```
$ XALT_TRACING=run python /my/path/data/hello_world.py

    track_executable():
    -> arg: 1: value: "../../data/hello_world.py"
    -> pattern: "/usr/bin/python3.10;/my/path/data/hello_world.py",
    track_executable token: 1: PKGS

myinit(0/1,LD_PRELOAD,/usr/bin/python3.10){
...
}
```

- This is what happens when a pattern matches a PKGS
- This can be shown when XALT_TRACING=run
- This tag causes the execution to be TRACKED

Final pattern

```
$ XALT_TRACING=run python2 /my/path/pkg_tracking.py --seq 22

   track_executable():
    -> arg: 1: value: "/my/path/pkg_tracking.py"
    -> pattern: "/usr/bin/python2.7;/my/path/pkg_tracking.py",
       track_executable token: 5: CONTINUE
    -> arg: 2: value: "--seq"
    -> arg: 3: value: "22"
    -> pattern: "/usr/bin/python2.7;", track_executable token: 1: PKGS

myinit(0/1,LD_PRELOAD,/usr/bin/python2.7){
  ...
}
```

- This last pattern is called after the command line has been processed
- In this case the tag is PKGS
- Without this pattern it is SKIP

Partial support for arguments w/ values

```
path_patterns = [
    ['CUSTOM', r'.*fakePrgm'],

path_arg_patterns = [
  ['SKIP',      r'.*fakePrgm;.*\/share\/.*'],
  ['KEEP',      r'.*fakePrgm;.*\/data\/.*'],
  ['JUMP_1',    r'.*fakePrgm;--opt'],
  ['CONTINUE', r'.*fakePrgm;--opt=.*'],
  ['KEEP',      r'.*fakePrgm;'],
]
```

- Suppose fakePrgm is some python like program
- Want to KEEP /data/; SKIP /share/ like before
- Want to jump over fakePrgm –opt /share/... by using JUMP_1 tag

Partial support for arguments w/ values (II)

```
$ fakePrgm --seq 21 --opt ../share/ ../../data/file.txt
    track_executable():
    -> arg: 1: value: "--seq"
    -> pattern: "/my/path/fakePrgm;--seq", track_executable token: 5: CONTINUE
    -> arg: 2: value: "21"
    -> arg: 3: value: "--opt"
    -> pattern: "/my/path/fakePrgm;--opt", track_executable token: 6: JUMP_1
    -> arg: 5: value: "../../data/file.txt"
   -> pattern: "/my/path/fakePrgm;/my/path/data/file.txt", track_executable token: 2: KEEP

myinit(0/1,LD_PRELOAD,/my/path/fakePrgm){
 ...
}
```

- Suppose fakePrgm is some python like program
- Want to KEEP /data/; SKIP /share/ like before
- Want to jump over fakePrgm –opt /share/... by using JUMP_1 tag

Show this in action

- xalt_configuration_report output
- example_run.txt

Conclusions

- Available now in the testing branch of XALT
- It works with the cases I have tested with
- Some support for skipping over arguments.
- Please test it out.

Future Topics?

- No Meeting in April. I'll be out of town.
- Next Meeting will be on May 18, 2023 at 10:00 am U.S. Central (15:00 UTC)