Commit cf158e8

Browse files

dmcgowan authored 5 days ago · 11 / 23 · Verified

```
Merge commit from fork
[release 1.6] validate uid/gid
```

release/1.6 · v1.6.38 2 parents e0c4dd9 + 9639b96 commit cf158e8

Filter files...

- oci
  - spec_opts.go
  - spec_opts_linux_test.go

2 files changed +112 -4 lines changed

Search within code

oci/spec_opts.go +20 -4

```
@@ -22,6 +22,7 @@ import (
        "encoding/json"
        "errors"
        "fmt"
+       "math"
        "os"
        "path/filepath"
        "runtime"

@@ -576,6 +577,20 @@ func WithUser(userstr string) SpecOpts {
                defer ensureAdditionalGids(s)
                setProcess(s)
                s.Process.User.AdditionalGids = nil
+               // While the Linux kernel allows the max UID to be MaxUint32 - 2,
+               // and the OCI Runtime Spec has no definition about the max UID,
+               // the runc implementation is known to require the UID to be <= MaxInt32.
+               //
+               // containerd follows runc's limitation here.
```

```
585 +                   //
586 +                   // In future we may relax this limitation to allow MaxUint32 - 2,
587 +                   // or, amend the OCI Runtime Spec to codify the implementation
        limitation.
588 +                   const (
589 +                           minUserID  = 0
590 +                           maxUserID  = math.MaxInt32
591 +                           minGroupID = 0
592 +                           maxGroupID = math.MaxInt32
593 +                   )
579 594
580 595                   // For LCOW it's a bit harder to confirm that the user actually exists on
        the host as a rootfs isn't
581 596                   // mounted on the host and shared into the guest, but rather the rootfs
        is constructed entirely in the
        @@ -592,8 +607,8 @@ func WithUser(userstr string) SpecOpts {
592 607                   switch len(parts) {
593 608                   case 1:
594 609                           v, err := strconv.Atoi(parts[0])
595       -                   if err != nil {
596       -                           // if we cannot parse as a uint they try to see if it is
        a username
610 +                           if err != nil || v < minUserID || v > maxUserID {
611 +                                   // if we cannot parse as an int32 then try to see if it
        is a username
597 612                                   return WithUsername(userstr)(ctx, client, c, s)
598 613                           }
599 614                           return WithUserID(uint32(v))(ctx, client, c, s)
        @@ -604,12 +619,13 @@ func WithUser(userstr string) SpecOpts {
604 619                           )
605 620                           var uid, gid uint32
606 621                           v, err := strconv.Atoi(parts[0])
607       -                   if err != nil {
622 +                           if err != nil || v < minUserID || v > maxUserID {
608 623                                   username = parts[0]
609 624                           } else {
610 625                                   uid = uint32(v)
611 626                           }
612       -                   if v, err = strconv.Atoi(parts[1]); err != nil {
627 +                           v, err = strconv.Atoi(parts[1])
628 +                           if err != nil || v < minGroupID || v > maxGroupID {
613 629                                   groupname = parts[1]
614 630                           } else {
615 631                                   gid = uint32(v)
```

oci/spec_opts_linux_test.go +92

```
@@ -32,6 +32,98 @@ import (
32 32          "golang.org/x/sys/unix"
33 33  )
34 34
35 +  //nolint:gosec
36 +  func TestWithUser(t *testing.T) {
37 +          t.Parallel()
38 +
39 +          expectedPasswd := `root:x:0:0:root:/root:/bin/ash
40 +  guest:x:405:100:guest:/dev/null:/sbin/nologin
41 +  `
42 +          expectedGroup := `root:x:0:root
43 +  bin:x:1:root,bin,daemon
44 +  daemon:x:2:root,bin,daemon
45 +  sys:x:3:root,bin,adm
46 +  guest:x:100:guest
47 +  `
48 +          td := t.TempDir()
49 +          apply := fstest.Apply(
50 +                  fstest.CreateDir("/etc", 0777),
51 +                  fstest.CreateFile("/etc/passwd", []byte(expectedPasswd), 0777),
52 +                  fstest.CreateFile("/etc/group", []byte(expectedGroup), 0777),
53 +          )
54 +          if err := apply.Apply(td); err != nil {
55 +                  t.Fatalf("failed to apply: %v", err)
56 +          }
57 +          c := containers.Container{ID: t.Name()}
58 +          testCases := []struct {
59 +                  user        string
60 +                  expectedUID uint32
61 +                  expectedGID uint32
62 +                  err         string
63 +          }{
64 +                  {
65 +                          user:        "0",
66 +                          expectedUID: 0,
67 +                          expectedGID: 0,
68 +                  },
69 +                  {
70 +                          user:        "root:root",
```

```
71  +                       expectedUID: 0,
72  +                       expectedGID: 0,
73  +               },
74  +               {
75  +                       user:        "guest",
76  +                       expectedUID: 405,
77  +                       expectedGID: 100,
78  +               },
79  +               {
80  +                       user:        "guest:guest",
81  +                       expectedUID: 405,
82  +                       expectedGID: 100,
83  +               },
84  +               {
85  +                       user: "guest:nobody",
86  +                       err:  "no groups found",
87  +               },
88  +               {
89  +                       user:        "405:100",
90  +                       expectedUID: 405,
91  +                       expectedGID: 100,
92  +               },
93  +               {
94  +                       user: "405:2147483648",
95  +                       err:  "no groups found",
96  +               },
97  +               {
98  +                       user: "-1000",
99  +                       err:  "no users found",
100 +               },
101 +               {
102 +                       user: "2147483648",
103 +                       err:  "no users found",
104 +               },
105 +       }
106 +       for _, testCase := range testCases {
107 +               testCase := testCase
108 +               t.Run(testCase.user, func(t *testing.T) {
109 +                       t.Parallel()
110 +                       s := Spec{
111 +                               Version: specs.Version,
112 +                               Root: &specs.Root{
113 +                                       Path: td,
114 +                               },
```

```
115 +                                   Linux: &specs.Linux{},
116 +                           }
117 +                           err := WithUser(testCase.user)(context.Background(), nil, &c, &s)
118 +                           if err != nil {
119 +                                   assert.EqualError(t, err, testCase.err)
120 +                           }
121 +                           assert.Equal(t, testCase.expectedUID, s.Process.User.UID)
122 +                           assert.Equal(t, testCase.expectedGID, s.Process.User.GID)
123 +                   })
124 +           }
125 + }
126 +
35  127    //nolint:gosec
36  128    func TestWithUserID(t *testing.T) {
37  129            t.Parallel()
```

Comments 0

Please sign in to comment.