containers / **storage** Public

Code Issues Pull requests Actions Projects Security and quality

Commit 935c58f

openshift-merge-bot[bot] authored on Oct 14, 2024 Verified

```
Merge pull request #2134 from mheon/fix_CVE-2024-9676

Use securejoin.SecureJoin when forming userns paths
```

2 parents b97ed7f + 491d72e commit 935c58f

3 files changed +76 -27

Top

Filter files…

- userns.go
- userns_test.go
- userns_unsupported.go

Search within code

userns.go

```
@@ -1,18 +1,21 @@
+ //go:build linux
+
 package storage

 import (
     "fmt"
     "os"
     "os/user"
-    "path/filepath"
```

```
 8   9          "strconv"
 9  10
10  11          drivers "github.com/containers/storage/drivers"
11  12          "github.com/containers/storage/pkg/idtools"
12  13          "github.com/containers/storage/pkg/unshare"
13  14          "github.com/containers/storage/types"
    15  +       securejoin "github.com/cyphar/filepath-securejoin"
14  16          libcontainerUser "github.com/moby/sys/user"
15  17          "github.com/sirupsen/logrus"
    18  +       "golang.org/x/sys/unix"
16  19      )
17  20
18  21      // getAdditionalSubIDs looks up the additional IDs configured for
        @@ -85,40 +88,59 @@ const nobodyUser = 65534
85  88      // parseMountedFiles returns the maximum UID and GID found in the /etc/passwd
            and
86  89      // /etc/group files.
87  90      func parseMountedFiles(containerMount, passwdFile, groupFile string) uint32 {
    91  +       var (
    92  +           passwd *os.File
    93  +           group  *os.File
    94  +           size   int
    95  +           err    error
    96  +       )
88  97          if passwdFile == "" {
89      -           passwdFile = filepath.Join(containerMount, "etc/passwd")
90      -       }
91      -       if groupFile == "" {
92      -           groupFile = filepath.Join(containerMount, "etc/group")
    98  +           passwd, err = secureOpen(containerMount, "/etc/passwd")
    99  +       } else {
   100  +           // User-specified override from a volume. Will not be in
   101  +           // container root.
   102  +           passwd, err = os.Open(passwdFile)
93 103          }
94      -
95      -       size := 0
96      -
97      -       users, err := libcontainerUser.ParsePasswdFile(passwdFile)
```

```
 98   104         if err == nil {
 99         -         for _, u := range users {
100         -             // Skip the "nobody" user otherwise we end up with 65536
101         -             // ids with most images
102         -             if u.Name == "nobody" || u.Name == "nogroup" {
103         -                 continue
104         -             }
105         -             if u.Uid > size && u.Uid != nobodyUser {
106         -                 size = u.Uid + 1
107         -             }
108         -             if u.Gid > size && u.Gid != nobodyUser {
109         -                 size = u.Gid + 1
      105   +         defer passwd.Close()
      106   +
      107   +         users, err := libcontainerUser.ParsePasswd(passwd)
      108   +         if err == nil {
      109   +             for _, u := range users {
      110   +                 // Skip the "nobody" user otherwise we end up with 65536
      111   +                 // ids with most images
      112   +                 if u.Name == "nobody" || u.Name == "nogroup" {
      113   +                     continue
      114   +                 }
      115   +                 if u.Uid > size && u.Uid != nobodyUser {
      116   +                     size = u.Uid + 1
      117   +                 }
      118   +                 if u.Gid > size && u.Gid != nobodyUser {
      119   +                     size = u.Gid + 1
      120   +                 }
110   121             }
111   122         }
112   123     }
113   124
114         -   groups, err := libcontainerUser.ParseGroupFile(groupFile)
      125   +   if groupFile == "" {
      126   +       group, err = secureOpen(containerMount, "/etc/group")
      127   +   } else {
      128   +       // User-specified override from a volume. Will not be in
      129   +       // container root.
      130   +       group, err = os.Open(groupFile)
      131   +   }
```

```
115  132        if err == nil {
116       -         for _, g := range groups {
117       -             if g.Name == "nobody" || g.Name == "nogroup" {
118       -                 continue
119       -             }
120       -             if g.Gid > size && g.Gid != nobodyUser {
121       -                 size = g.Gid + 1
     133  +         defer group.Close()
     134  +
     135  +         groups, err := libcontainerUser.ParseGroup(group)
     136  +         if err == nil {
     137  +             for _, g := range groups {
     138  +                 if g.Name == "nobody" || g.Name == "nogroup" {
     139  +                     continue
     140  +                 }
     141  +                 if g.Gid > size && g.Gid != nobodyUser {
     142  +                     size = g.Gid + 1
     143  +                 }
122  144              }
123  145          }
124  146      }
          @@ -309,3 +331,14 @@ func getAutoUserNSIDMappings(
309  331      gidMap := append(availableGIDs.zip(requestedContainerGIDs),
         additionalGIDMappings...)
310  332      return uidMap, gidMap, nil
311  333  }
     334  +
     335  + // Securely open (read-only) a file in a container mount.
     336  + func secureOpen(containerMount, file string) (*os.File, error) {
     337  +     tmpFile, err := securejoin.OpenInRoot(containerMount, file)
     338  +     if err != nil {
     339  +         return nil, err
     340  +     }
     341  +     defer tmpFile.Close()
     342  +
     343  +     return securejoin.Reopen(tmpFile, unix.O_RDONLY)
     344  + }
```

userns_test.go

```
@@ -1,3 +1,5 @@
+ //go:build linux
+
  package storage

  import (
```

userns_unsupported.go

```
@@ -0,0 +1,14 @@
+ //go:build !linux
+
+ package storage
+
+ import (
+     "errors"
+
+     "github.com/containers/storage/pkg/idtools"
+     "github.com/containers/storage/types"
+ )
+
+ func (s *store) getAutoUserNS(_ *types.AutoUserNsOptions, _ *Image, _ rwLayerStore, _ []roLayerStore) ([]idtools.IDMap, []idtools.IDMap, error) {
+     return nil, nil, errors.New("user namespaces are not supported on this platform")
+ }
```

Comments 0

Please sign in to comment.