

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 users paths

main (#2134) · v1.59.1 ... v1.56.0

2 parents [b97ed7f](#) + [491d72e](#) commit 935c58f

3 files changed

+76 -27

Top

Filter files...



- users.go
- users\_test.go
- users\_unsupported.go

Search within code



users.go



```

@@ -1,18 +1,21 @@
1 + //go:build linux
2 +
1 3 package storage
2 4
3 5 import (
4 6     "fmt"
5 7     "os"
6 8     "os/user"
7 -     "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    15 +     securejoin "github.com/cyphar/filepath-securejoin"
14    16     libcontainerUser "github.com/moby/sys/user"
15    17     "github.com/sirupsen/logrus"
18    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    91 +     var (
92    92 +         passwd *os.File
93    93 +         group  *os.File
94    94 +         size   int
95    95 +         err    error
96    96 +     )
88    97     if passwdFile == "" {
89    97 -         passwdFile = filepath.Join(containerMount, "etc/passwd")
90    98 -     }
91    99 -     if groupFile == "" {
92    99 -         groupFile = filepath.Join(containerMount, "etc/group")
98    100 +         passwd, err = secureOpen(containerMount, "/etc/passwd")
99    101 +     } else {
100   102 +         // User-specified override from a volume. Will not be in
101   103 +         // container root.
102   104 +         passwd, err = os.Open(passwdFile)
93   103     }
94   105 -
95   106 -     size := 0
96   107 -
97   108 -     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
110     +
111     +         defer passwd.Close()
112     +
113     +         users, err := libcontainerUser.ParsePasswd(passwd)
114     +         if err == nil {
115     +             for _, u := range users {
116     +                 // Skip the "nobody" user otherwise we end up with 65536
117     +                 // ids with most images
118     +                 if u.Name == "nobody" || u.Name == "nogroup" {
119     +                     continue
120     +                 }
121     +                 if u.Uid > size && u.Uid != nobodyUser {
122     +                     size = u.Uid + 1
123     +                 }
124     +                 if u.Gid > size && u.Gid != nobodyUser {
125     +                     size = u.Gid + 1
126     +                 }
127     +             }
128     +         }
129     +     }
130     +
131     +     groups, err := libcontainerUser.ParseGroupFile(groupFile)
132     +     if groupFile == "" {
133     +         group, err = secureOpen(containerMount, "/etc/group")
134     +     } else {
135     +         // User-specified override from a volume. Will not be in
136     +         // container root.
137     +         group, err = os.Open(groupFile)
138     +     }
```

```

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
122     +         defer group.Close()
123     +
124     +         groups, err := libcontainerUser.ParseGroup(group)
125     +         if err == nil {
126     +             for _, g := range groups {
127     +                 if g.Name == "nobody" || g.Name == "nogroup" {
128     +                     continue
129     +                 }
130     +                 if g.Gid > size && g.Gid != nobodyUser {
131     +                     size = g.Gid + 1
132     +                 }
133     +             }
134     +         }
135     +     }
136     + }
137     + }
138     + }
139     + }
140     + }
141     + }
142     + }
143     + }
144     + }
145     + }
146     + }
147     + }
148     + }
149     + }
150     + }
151     + }
152     + }
153     + }
154     + }
155     + }
156     + }
157     + }
158     + }
159     + }
160     + }
161     + }
162     + }
163     + }
164     + }
165     + }
166     + }
167     + }
168     + }
169     + }
170     + }
171     + }
172     + }
173     + }
174     + }
175     + }
176     + }
177     + }
178     + }
179     + }
180     + }
181     + }
182     + }
183     + }
184     + }
185     + }
186     + }
187     + }
188     + }
189     + }
190     + }
191     + }
192     + }
193     + }
194     + }
195     + }
196     + }
197     + }
198     + }
199     + }
200     + }
201     + }
202     + }
203     + }
204     + }
205     + }
206     + }
207     + }
208     + }
209     + }
210     + }
211     + }
212     + }
213     + }
214     + }
215     + }
216     + }
217     + }
218     + }
219     + }
220     + }
221     + }
222     + }
223     + }
224     + }
225     + }
226     + }
227     + }
228     + }
229     + }
230     + }
231     + }
232     + }
233     + }
234     + }
235     + }
236     + }
237     + }
238     + }
239     + }
240     + }
241     + }
242     + }
243     + }
244     + }
245     + }
246     + }
247     + }
248     + }
249     + }
250     + }
251     + }
252     + }
253     + }
254     + }
255     + }
256     + }
257     + }
258     + }
259     + }
260     + }
261     + }
262     + }
263     + }
264     + }
265     + }
266     + }
267     + }
268     + }
269     + }
270     + }
271     + }
272     + }
273     + }
274     + }
275     + }
276     + }
277     + }
278     + }
279     + }
280     + }
281     + }
282     + }
283     + }
284     + }
285     + }
286     + }
287     + }
288     + }
289     + }
290     + }
291     + }
292     + }
293     + }
294     + }
295     + }
296     + }
297     + }
298     + }
299     + }
300     + }
301     + }
302     + }
303     + }
304     + }
305     + }
306     + }
307     + }
308     + }
309     + }
310     + }
311     + }
312     + }
313     + }
314     + }
315     + }
316     + }
317     + }
318     + }
319     + }
320     + }
321     + }
322     + }
323     + }
324     + }
325     + }
326     + }
327     + }
328     + }
329     + }
330     + }
331     + }
332     + }
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     +     return securejoin.Reopen(tmpFile, unix.O_RDONLY)
343     + }
344     + }

```



```
@@ -309,3 +331,14 @@ func getAutoUserNSIDMappings(
```

```

309 331     gidMap := append(availableGIDs.zip(requestedContainerGIDs),
310 332     additionalGIDMappings...)
311 333     return uidMap, gidMap, nil
312 334 }
313 335 // Securely open (read-only) a file in a container mount.
314 336 func secureOpen(containerMount, file string) (*os.File, error) {
315 337     tmpFile, err := securejoin.OpenInRoot(containerMount, file)
316 338     if err != nil {
317 339         return nil, err
318 340     }
319 341     defer tmpFile.Close()
320 342     return securejoin.Reopen(tmpFile, unix.O_RDONLY)
321 343 }
322 344 }

```

usersns\_test.go



```
... @@ -1,3 +1,5 @@
1 + //go:build linux
2 +
1 3   package storage
2 4
3 5   import (
...
↓
```

```
usersns_unsupported.go
... @@ -0,0 +1,14 @@
1 + //go:build !linux
2 +
3 + package storage
4 +
5 + import (
6 +     "errors"
7 +
8 +     "github.com/containers/storage/pkg/idtools"
9 +     "github.com/containers/storage/types"
10 + )
11 +
12 + func (s *store) getAutoUserNS(_ *types.AutoUserNsOptions, _ *Image, _
    rwLayerStore, _ []roLayerStore) ([]idtools.IDMap, []idtools.IDMap, error) {
13 +     return nil, nil, errors.New("user namespaces are not supported on this
    platform")
14 + }
```

## Comments 0



Please [sign in](#) to comment.