knadh / **listmonk** Public

Code | Issues 81 | Pull requests 12 | Discussions | Actions | Projects

Commit 347f597

knadh committed last week · ✓ 4 / 4

```
Fix serveral missing permission checks across multiple handlers.

- List permission check in campaign update.
- List permission check in CSV import.
- Granular list/subscriber permission checks in bulk subscriber
  handlers.
- Subscriber JSON export endpoint.
- Test mail endpoint.

Damn, I can't believe this was all missed ;(
```

master · v6.1.0 nightly

1 parent 171a597 commit 347f597

4 files changed +92 -18 lines changed

Top

Filter files…

- cmd
  - campaigns.go
  - import.go
  - subscribers.go
- internal/auth
  - models.go

4 files changed +92 -18 lines changed

Search within code

cmd/campaigns.go

```
@@ -330,6 +330,10 @@ func (a *App) UpdateCampaign(c echo.Context) error {
330  330            return err
```

```
331 331         }
332 332
    333 +     // Filter lists against the current user's permitted lists.
    334 +     user := auth.GetUser(c)
    335 +     o.ListIDs = user.FilterListsByPerm(auth.PermTypeGet|auth.PermTypeManage,
        o.ListIDs)
    336 +
333 337         if c, err := a.validateCampaignFields(o); err != nil {
334 338             return echo.NewHTTPError(http.StatusBadRequest, err.Error())
335 339         } else {
```

```
@@ -549,6 +553,16 @@ func (a *App) TestCampaign(c echo.Context) error {
```

```
549 553             return err
550 554         }
551 555
    556 +     // Check if the user has permission to access the subscribers.
    557 +     user := auth.GetUser(c)
    558 +     subIDs := make([]int, len(subs))
    559 +     for i, s := range subs {
    560 +         subIDs[i] = s.ID
    561 +     }
    562 +     if err := a.hasSubPerm(user, subIDs); err != nil {
    563 +         return err
    564 +     }
    565 +
552 566         // Get the campaign from the DB for previewing.
553 567         tplID, _ := strconv.Atoi(c.FormValue("template_id"))
554 568         camp, err := a.core.GetCampaignForPreview(id, tplID)
```

cmd/import.go

```
@@ -7,6 +7,7 @@ import (
```

```
7  7          "os"
8  8          "strings"
9  9
   10 +       "github.com/knadh/listmonk/internal/auth"
10 11         "github.com/knadh/listmonk/internal/subimporter"
11 12         "github.com/knadh/listmonk/models"
12 13         "github.com/labstack/echo/v4"
```

```
@@ -27,6 +28,14 @@ func (a *App) ImportSubscribers(c echo.Context) error {
```

```
27  28              a.i18n.Ts("import.invalidParams", "error", err.Error()))
28  29          }
29  30
    31  +       // Filter list IDs against the current user's permitted lists.
    32  +       user := auth.GetUser(c)
    33  +       opt.ListIDs = user.FilterListsByPerm(auth.PermTypeManage, opt.ListIDs)
    34  +       if len(opt.ListIDs) == 0 {
    35  +           return echo.NewHTTPError(http.StatusForbidden,
    36  +               a.i18n.Ts("globals.messages.permissionDenied", "name", "lists"))
    37  +       }
    38  +
30  39          // Validate mode.
31  40          if opt.Mode != subimporter.ModeSubscribe && opt.Mode !=
            subimporter.ModeBlocklist {
32  41              return echo.NewHTTPError(http.StatusBadRequest,
            a.i18n.T("import.invalidMode"))
```

cmd/subscribers.go

```
    @@ -303,8 +303,14 @@ func (a *App) UpdateSubscriber(c echo.Context) error {
303 303
304 304     // SubscriberSendOptin sends an optin confirmation e-mail to a subscriber.
305 305     func (a *App) SubscriberSendOptin(c echo.Context) error {
    306 +       user := auth.GetUser(c)
    307 +
306 308         // Fetch the subscriber.
307 309         id := getID(c)
    310 +       if err := a.hasSubPerm(user, []int{id}); err != nil {
    311 +           return err
    312 +       }
    313 +
308 314         out, err := a.core.GetSubscriber(id, "", "")
309 315         if err != nil {
310 316             return err
    @@ -320,8 +326,14 @@ func (a *App) SubscriberSendOptin(c echo.Context) error
            {
320 326
321 327     // BlocklistSubscriber handles the blocklisting of a given subscriber.
322 328     func (a *App) BlocklistSubscriber(c echo.Context) error {
    329 +       user := auth.GetUser(c)
```

```
                330 +
323             331         // Update the subscribers in the DB.
324             332         id := getID(c)
                333 +       if err := a.hasSubPerm(user, []int{id}); err != nil {
                334 +           return err
                335 +       }
                336 +
325             337         if err := a.core.BlocklistSubscribers([]int{id}); err != nil {
326             338             return err
327             339         }
                    @@ -331,6 +343,8 @@ func (a *App) BlocklistSubscriber(c echo.Context) error
                    {
331             343
332             344     // BlocklistSubscribers handles the blocklisting of one or more subscribers.
333             345     func (a *App) BlocklistSubscribers(c echo.Context) error {
                346 +       user := auth.GetUser(c)
                347 +
334             348         var req subQueryReq
335             349         if err := c.Bind(&req); err != nil {
336             350             return echo.NewHTTPError(http.StatusBadRequest,
                    @@ -341,6 +355,10 @@ func (a *App) BlocklistSubscribers(c echo.Context)
                    error {
341             355                 a.i18n.Ts("globals.messages.errorInvalidIDs", "error", "ids"))
342             356         }
343             357
                358 +       if err := a.hasSubPerm(user, req.SubscriberIDs); err != nil {
                359 +           return err
                360 +       }
                361 +
344             362         // Update the subscribers in the DB.
345             363         if err := a.core.BlocklistSubscribers(req.SubscriberIDs); err != nil {
346             364             return err
                    @@ -384,6 +402,10 @@ func (a *App) ManageSubscriberLists(c echo.Context)
                    error {
384             402             return echo.NewHTTPError(http.StatusBadRequest,
                    a.i18n.T("subscribers.errorNoListsGiven"))
385             403         }
386             404
                405 +       if err := a.hasSubPerm(user, subIDs); err != nil {
                406 +           return err
```

```
        407 +     }
        408 +
387     409       // Filter lists against the current user's permitted lists.
388     410       listIDs := user.FilterListsByPerm(auth.PermTypeGet|auth.PermTypeManage,
                  req.TargetListIDs)
389     411
                  @@ -414,8 +436,14 @@ func (a *App) ManageSubscriberLists(c echo.Context)
                  error {
414     436
415     437   // DeleteSubscriber handles deletion of a single subscriber.
416     438   func (a *App) DeleteSubscriber(c echo.Context) error {
        439 +     user := auth.GetUser(c)
        440 +
417     441       // Delete the subscribers from the DB.
418     442       id := getID(c)
        443 +     if err := a.hasSubPerm(user, []int{id}); err != nil {
        444 +         return err
        445 +     }
        446 +
419     447       if err := a.core.DeleteSubscribers([]int{id}, nil); err != nil {
420     448           return err
421     449       }
                  @@ -425,6 +453,8 @@ func (a *App) DeleteSubscriber(c echo.Context) error {
425     453
426     454   // DeleteSubscribers handles bulk deletion of one or more subscribers.
427     455   func (a *App) DeleteSubscribers(c echo.Context) error {
        456 +     user := auth.GetUser(c)
        457 +
428     458       // Multiple IDs.
429     459       ids, err := parseStringIDs(c.Request().URL.Query()["id"])
430     460       if err != nil {
                  @@ -436,6 +466,10 @@ func (a *App) DeleteSubscribers(c echo.Context) error {
436     466               a.i18n.Ts("globals.messages.errorInvalidIDs", "error", "ids"))
437     467       }
438     468
        469 +     if err := a.hasSubPerm(user, ids); err != nil {
        470 +         return err
        471 +     }
        472 +
439     473       // Delete the subscribers from the DB.
```

```
440  474          if err := a.core.DeleteSubscribers(ids, nil); err != nil {
441  475              return err
     @@ -473,8 +507,11 @@ func (a *App) DeleteSubscribersByQuery(c echo.Context)
     error {
473  507          }
474  508      }
475  509
     510  +      // Filter list IDs against the current user's permitted lists.
     511  +      listIDs := user.GetPermittedListIDs(req.ListIDs)
     512  +
476  513      // Delete the subscribers from the DB.
477       -      if err := a.core.DeleteSubscribersByQuery(req.Search, req.Query,
     req.ListIDs, req.SubscriptionStatus); err != nil {
     514  +      if err := a.core.DeleteSubscribersByQuery(req.Search, req.Query, listIDs,
     req.SubscriptionStatus); err != nil {
478  515              return err
479  516      }
480  517
     @@ -509,8 +546,11 @@ func (a *App) BlocklistSubscribersByQuery(c
     echo.Context) error {
509  546          }
510  547      }
511  548
     549  +      // Filter list IDs against the current user's permitted lists.
     550  +      listIDs := user.GetPermittedListIDs(req.ListIDs)
     551  +
512  552      // Update the subscribers in the DB.
513       -      if err := a.core.BlocklistSubscribersByQuery(req.Search, req.Query,
     req.ListIDs, req.SubscriptionStatus); err != nil {
     553  +      if err := a.core.BlocklistSubscribersByQuery(req.Search, req.Query,
     listIDs, req.SubscriptionStatus); err != nil {
514  554              return err
515  555      }
516  556
     @@ -544,7 +584,7 @@ func (a *App) ManageSubscriberListsByQuery(c
     echo.Context) error {
544  584      }
545  585
546  586      // Filter lists against the current user's permitted lists.
```

```
547      -      sourceListIDs :=
         user.FilterListsByPerm(auth.PermTypeGet|auth.PermTypeManage, req.ListIDs)
     587 +      sourceListIDs := user.GetPermittedListIDs(req.ListIDs)
548  588        targetListIDs :=
         user.FilterListsByPerm(auth.PermTypeGet|auth.PermTypeManage, req.TargetListIDs)
549  589
550  590        // Run the action in the DB.
         @@ -587,6 +627,12 @@ func (a *App) ExportSubscriberData(c echo.Context)
         error {
587  627        // list subscriptions, campaign views, and link clicks. Names of
588  628        // private lists are replaced with "Private list".
589  629        id := getID(c)
     630 +
     631 +      // Check if the user has access to at least one of the lists on the
         subscriber.
     632 +      if err := a.hasSubPerm(auth.GetUser(c), []int{id}); err != nil {
     633 +          return err
     634 +      }
     635 +
590  636        _, b, err := a.exportSubscriberData(id, "", a.cfg.Privacy.Exportable)
591  637        if err != nil {
592  638            a.log.Printf("error exporting subscriber data: %s", err)
         @@ -670,23 +716,10 @@ func (a *App) filterListQueryByPerm(param string, qp
         url.Values, user auth.User)
670  716                return nil, echo.NewHTTPError(http.StatusBadRequest,
         a.i18n.T("globals.messages.invalidID"))
671  717            }
672  718
673      -          listIDs = user.FilterListsByPerm(auth.PermTypeGet|auth.PermTypeManage,
         ids)
674      -      }
675      -
676      -      // There are no incoming params. If the user doesn't have permission to get
         all subscribers,
677      -      // filter by the lists they have access to.
678      -      if len(listIDs) == 0 {
679      -          if _, ok := user.PermissionsMap[auth.PermSubscribersGetAll]; !ok {
680      -              if len(user.GetListIDs) > 0 {
681      -                  listIDs = user.GetListIDs
682      -              } else {
```

```
683     -                // User doesn't have access to any lists.
684     -                listIDs = []int{-1}
685     -            }
686     -        }
        719 +        listIDs = ids
687 720       }
688 721
689     -     return listIDs, nil
        722 +     return user.GetPermittedListIDs(listIDs), nil
690 723   }
691 724
692 725   // formatSQLExp does basic sanitisation on arbitrary
```

internal/auth/models.go

```
@@ -309,3 +309,21 @@ func (u *User) FilterListsByPerm(types PermType,
listIDs []int) []int {
309 309
310 310       return out
311 311   }
        312 +
        313 + // GetPermittedListIDs filters the given list IDs by the user's get/manage
        314 + // permissions and returns the filtered set. Unlike `FilterListsByPerm()`, if
no
        315 + // IDs are present (empty input or 0 permitted lists), it falls back to the
user's
        316 + // permitted list IDs if any. This is useful for endpoints which accept a few
IDs
        317 + // or the lack of which implies "all".
        318 + func (u *User) GetPermittedListIDs(listIDs []int) []int {
        319 +     listIDs = u.FilterListsByPerm(PermTypeGet|PermTypeManage, listIDs)
        320 +     if len(listIDs) == 0 {
        321 +         if _, ok := u.PermissionsMap[PermSubscribersGetAll]; !ok {
        322 +             if len(u.GetListIDs) > 0 {
        323 +                 return u.GetListIDs
        324 +             }
        325 +             return []int{-1}
        326 +         }
        327 +     }
        328 +     return listIDs
```

```
329 + }
```

Comments 0

Please sign in to comment.