Enumeration

Enumeration, in terms of security, is a vulnerability, which enables a potential attacker to guess some hidden system information. There are many types of enumeration threats, but we will discuss only two of them in this topic:

- Username enumeration
- File enumeration

Username enumeration

The username enumeration is an activity in which an attacker tries to retrieve valid usernames from a web application. The web applications are mostly vulnerable to this type of attack on login pages, registration form pages or password reset pages.

If the system is vulnerable to the username enumeration attack, the attacker may be able to obtain a list of existing usernames in the system by submitting input (valid and invalid user names) and analyzing the server response (error messages). The attacker can then run a dictionary attack to further exploit the obtained information.

On this page

Real-world example of a username enumeration attack

There is one well-known username enumeration vulnerability related to previous versions of Apache web server. Some distributions contained a misconfiguration, which enabled potential attackers to identify existing usernames.

When an attacker submits an HTTP request for a possible user's home page, *http://www.example.com/~<username>*, the server responses differently depending on whether the username exists or not:

- If the username exists and does not have a homepage, the server responses with HTTP result code 403, and the server message "*You don't have permission to access /~username on this server.*"
- If the username does not exist, the server responses with HTTP result code 404 and the message "*The requested URL /~username was not found on this server.*"

Because the server responses differently in these cases, the potential attacker can test and enumerate existing usernames. The attacker can then exploit this data for further attacks on the server.

Example of a web application username enumeration vulnerability in web application

When you create a web part in your web application, which enables users to log in, do not reveal information about existing usernames:

Wrong

If the submitted username is incorrect:

```
Entered login does not exist.
```

If the submitted username exists, but the password is incorrect:

```
Entered password is incorrect.
```

This way, the attacker can learn, which usernames exist in the system and which do not.

How to prevent username enumeration attacks in Kentico CMS

The default configuration of Kentico is protected against this type of attack.

If you plan to create a custom login web part, be sure to **show only generic error messages**:

Correct

In both situations (username does not exist or the password is incorrect):

```
Authentication failed.
```

This way, the attacker cannot distinguish between valid and invalid usernames.

In registration web parts, you cannot completely eliminate this vulnerability, because you have to check (and tell the user) if the submitted username already exists. Therefore, **always include CAPTCHA** in these web parts to prevent automatic collecting of the data by scripts.

File enumeration

In this type of enumeration attack, the attacker tries to guess the file names on the server and manage to gain access to them.

In Kentico, the attacker can for example try to guess the name of an export file. The export files are generally located in the *<web project>\CMSSiteUtils\Export* folder. The attacker can learn the structure of the file name and eventually guess an existing one (for example, by trying out different time stamps).

To protect your servers against this type of attack, **forbid access to sensitive directories** in the *web.config* file. See Restricting access to directories.

Summary

- Do not reveal username and password details on you web pages (how long the username/password should be, which characters it should contain, etc.)
- Show only generic error messages (*Login failed.* / *The combination of username and password is invalid.*) in custom login web parts.
- Use CAPTCHA in registration forms.
- Forbid access from the outside to directories, whose names could be guessed by potential attackers.