CMS SECURITY TIPS AND TRICKS

L.Petkova. PhD
University of Library an0d Information technology (UNIBIT)

lilyanapetkova92@gmail.com

Abstract: *Content management systems become more popular and due to the speed provided during the development the number of websites developed with a CMS growns extremely. This is the reason the sites developed with a CMS become a more attractive target for cybercriminals. The CMSes are divided in 2 categories: free open-source and paid. According to that categorization the security level of the system is limitless. As part of my research, we are investigating only the free tools supporting the application development process and the presented tools and softwares are only from personal experience! So for the present article, we are going to present the Umbraco CMS with its integrated security tools and some other free to use tricks in order for the system we develop to be secure.*

Keywords: SECURITY, VULNERABILITY, UMBRACO, CONTENT MANAGEMENT SYSTEM

1. Introduction

It is not enough to just have a website, but to adapt with the emerging technologies and security flaws every single day.

With the discovery of content management system(CMS) solutions the web development becomes faster and more flexible through money, time and support. Which makes the choice of working with CMS inevitable. The main goal of our research is to focus on the free perspective of software development.

Understanding the sources of vulnerabilities and learning to program securely is imperative to protecting the Internet and ourselves from attack. [11][12][13]

The third and most comprehensive method integrates security and risk management into the development cycle, either with involvement of security team members or by developing a core security focus on the part of developers. [13]

The present article demonstrates secure websites' development with an open source CMSs in the .NET world. All of the provided samples are gained through personal experience and helped preventing several different attacks or having options to increase the security level of a site.

As a result we provide security analyses on websites developed by us with the help of one open source CMS - Umbraco CMS.

2. Content Management System

A content management system, or abbreviated CMS, is a software that provides speed in developing a website without the need of specialized technical knowledge. But having the necessary skills as a developer we can build powerful and enough secure websites. [4]

Content management systems provide an automated control on the collection, management and the distribution of information.

The functionalities of a CMS are: Collecting (authoring process, acquisition process, automatic format conversion, aggregation process, information reusability and consistent content segmentation into metadata); Management (system security, content storage in a neutral format, automatic indexing, simple and rapid research system, utilisation of standard technologies and languages, simple content modifying, document versioning, content management through a single interface, management of all type of formats, possibility of automatic updating, possibility of automatic deletion, compatibility with other applications, content condivision with other applications, system scalability, automatic workflow, system simplicity and flexibility, security of all content approval processes); Publication (template utilisation to create publications, automatic content conversion, simultaneous creation of more publications, advanced personalisation system, multiple-language interface). [4][5]

In our personal experience we are working on ASP.NET platforms, so the CMS analysed in the current research is also based on .NET: Umbraco CMS.

3. Security Options For Umbraco CMS

The security options list provides most of the details necessary for securing a website developed with Umbraco CMS. [2][3]

Due to the different versions of that CMS, there might be differences in the settings necessary for version 7 and version 8 of Umbraco CMS. As we are analysing the Umbraco CMS we are going to provide not only the setting but the difference too.

3.1. SSL/HTTPS

One step in securing a website is by enabling the HTTPS protocol. There are a lot of benefits in using HTTPS, such as: protection against phishing attacks or "Man in the middle" attacks (which incepts a communication between two systems[1][2]). Another benefit of using HTTPS protocol on the website is the "preference" of Google search engine of the HTTPS over HTTP. As we already have investigated SEO perspective in website security, the ranking of sites with HTTPS protocol enabled is higher.[9][10]

Sample 1: *SSL/HTTPS configuration Umbraco 8 CMS*

```
<add key="Umbraco.Core.UseHttps" value="false" />
```

Sample 2: *SSL/HTTPS configuration Umbraco 7 CMS*

```
<add key="umbracoUseSSL" value="false" />
```

Once the site is HTTPS configured, there should be a redirect rule attached in order to redirect all requests to HTTPS. That way all requests, services, backoffice handlers will be denied.

Sample 3: *HTTPS force rule*

```
<rule name="HTTPS force" enabled="true" stopProcessing="false">
  <match url="(.*)" />
  <conditions>
     <add input="{HTTPS}" pattern="^OFF$" />
  </conditions>
  <action type="Redirect"
url="https://{HTTP_HOST}{REQUEST_URI}"
redirectType="Permanent" />
</rule>
```

3.2. HTTP headers

In one of our previous researches we have presented more detailed the purpose of the HTTP headers. Their use globally protects the entire website from a user-end perspective. But the Umbraco backoffice needs additional protection.

According the Security Headers researcher Scott Helme 7 out of 10 are the most valuable headers which should be included in a site: X-Frame-Options; Strict-Transport-Security; Content-Security-Policy; X-XSS-Protection; X-Content-Type-Options; Referrer-Policy and Feature- Policy! [7]

As the backoffice provides the website's content management, the headers we should definitely use is the Content Security Policy header, which as we have already investigated "reduce the harm caused by content injection attacks". [8]

So in addition to the global site HTTP headers in order to protect the Umbraco backoffice we need to add additional configuration. To set

it up we need to add location related to the Umbraco backoffice access and use the already known code structure!

Sample 4: *HTTP headers security*

```
<location path="umbraco">
  <system.webServer>
    <httpProtocol>
      <customHeaders>
        <remove name="X-Frame-Options" />
        <add name="X-Frame-Options" value="SAMEORIGIN" />
        <remove name="Content-Security-Policy" />
        <add name="Content-Security-Policy" value="default-src 'self'
www.gravatar.com player.vimeo.com *.vimeocdn.com
packages.umbraco.org our.umbraco.org;script-src 'self' 'unsafe-inline'
'unsafe-eval';style-src 'self' 'unsafe-inline';img-src 'self' data:
www.gravatar.com umbraco.tv;font-src 'self';" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</location>
```

3.3. Backoffice users

As Umbraco is an open source content management system, developers all around the world can contribute to that CMS with various additional code packages.

3.3.1. Password rules configuration

Every application using password for some reason, needs to configure the password a complex structure to be hard to guess. In Umbraco CMS websites this is managed through the web.config. Websites using Umbraco CMS though need to make this configuration for users which use the front end website, called members, and for users which use the Umbraco backoffice. From a programming point of view each of them is managed with different security providers which are not that different except for the reason that are used for different website sides (*UsersMembershipProvider for users* and *UmbracoMembershipProvider for members*).

For the structure we can define the minimum characters need to be used in the password (***minRequiredPasswordLength***), the need of symbol characters (***minRequiredNonalphanumericCharac ters***), also we can define a regular expression for mapping the password to some kind of model (***passwordStrengthRegularExpression***), and so on (*see Sample 5*).

Sample 5: *User Password configuration*

```
<add name="UsersMembershipProvider"
type="Umbraco.Web.Security.Providers.UsersMembershipProvider,
Umbraco" minRequiredNonalphanumericCharacters="1"
passwordStrengthRegularExpression="^(?=.*\d)(?=.*[a-
z])(?=.*[@#$%^&amp;+=])(?=.*[A-Z]).{12,}$"
minRequiredPasswordLength="12" useLegacyEncoding="false"
enablePasswordRetrieval="false" enablePasswordReset="true"
requiresQuestionAndAnswer="false" passwordFormat="Hashed" />
```

3.3.2. Two steps authentication

The two steps authentication is always a plus in order to protect the application you use or develop. Each well secure software provides that additional authentication proof! As an open source developers can develop such plugins or use already developed packages.

Such packages we have used in several websites developed by us, is the Umbraco 2FA package (*see Figure 1*)!



Figure 1: *Umbraco 2FA*

After its quick backoffice installation, the set up is even quicker than this.

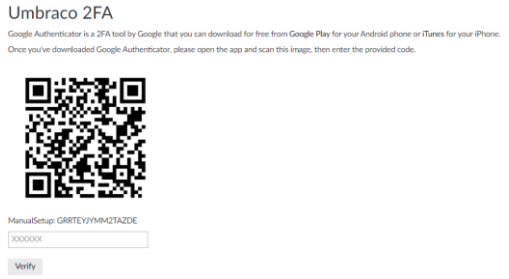


Figure 2: *Umbraco 2FA setup*

By just pressing the Setup button from the backoffice (*see Figure 2*) and installing the Google Authenticator mobile application (*see Figure 5*), after entering the provided code or just scanning the QR code, the setting up is done.



Figure 3: *Google Authenticator code*

And whenever the user tries to login to the backoffice, the second step is just to enter the code from the Google Authenticator application (*see Figure 3*) and the user is logged in (*see Figure 4*).

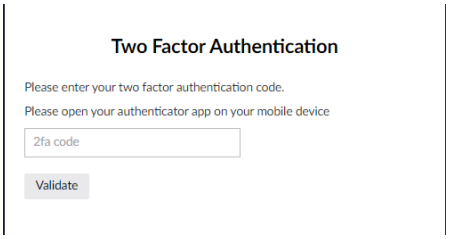


Figure 4: *Two factor authentication Umbraco form*

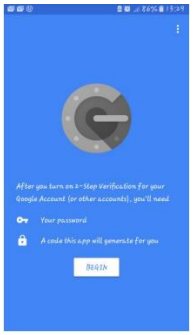


Figure 5: *Google Authenticator application*

The best part of using the Google Authenticator application is that the code is updated in every 45 seconds, so remember it is not an option.

3.3.3. Health Checks

Health Checks is a plugin of Umbraco backoffice which provides a list of areas to check to see if Umbraco is configured and working according to the best practices.

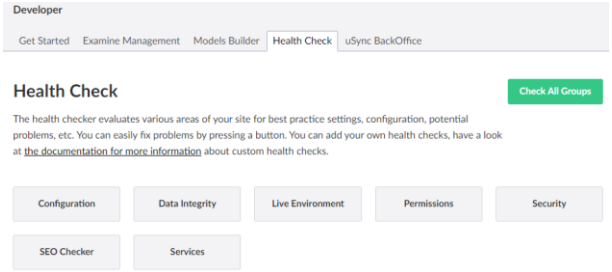


Figure 6: *Health Check plugin*

In the security section the Health Check evaluates potential vulnerabilities and flaws and provides options to fix them, such as *sniffing vulnerabilities*, *cross-site scripting* , *HTTPS configuration*, and other checks (*see Figure 7*).

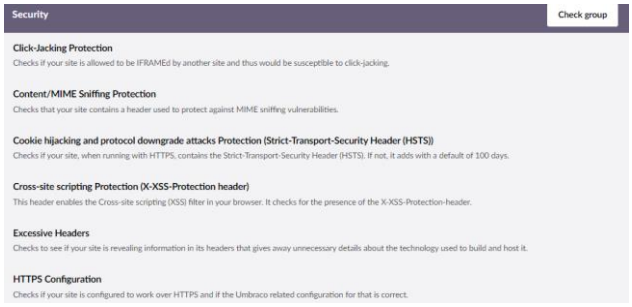


Figure 7: *Health Check security area overview*

4. Results

Our research is based on analyses on several security checks on a few preselected web applications. In our previous articles we have used the ALEXA analytical insight to choose six websites from an excerpt of 50 sites based on a selected Country category - Bulgaria. [7][8][9][10]

But as part of this current research we are looking for websites developed by Umbraco CMS. For that purpose the current analyses are based on websites developed from our personal experience. We are not going to present the name of sites due to law and policies restrictions! [6]

Table 1: *Security statistic*

	site A	**site B**	**site C**	**site D**	**site E**	**site F**
HTTP headers	+	-	-	+	-	-
SSL//HTTPs	+	+	+	+	+	+
Health Checks	+	+	+	+	+	+
Password configuration	+	+	+	+	+	+

The results from Table 1 show that the web applications developed from us are not fully secure according to the options we have reviewed so far. Most of the reasons for the lack of security are related to the clients requirements or the fact that we have not still reached the software development phase [6] for integrating the necessary security level in the website.

5. Conclusion

As part of web development, the security perspective takes a major role in it. Being up-to-date with the newelest technologies is one of the biggest steps which can help protect our applications and keep the attackers away.

From the analytics we have provided we have to consider that the use of CMS for web development can give the necessary level of protection. Which as always if not properly integrated, its purpose stays unachieved.

Our entire research is based on personal experience!

References

[1] OWASP Secure Headers Project
[2] Umbraco community, https://our.umbraco.com/
[3] Wahlberg,, N., Sterling, P., Umbraco User's Guide, 2011
[4] Canavan, T., CMS Security Handbook: The Comprehensive Guide for WordPress, Joomla, Drupal, and Plone, April 2011
[5] Benevolo, C., Evaluation of Content Management Systems (CMS): a Supply Analysis, January 2017
[6] Petkova, L., SECURITY STANDARDS in software development, 2017
[7] Petkova, L., HTTP SECURITY HEADERS, 2018
[8] Petkova, L., CONTENT SECURITY POLICY VALIDATION, 2019
[9] Petkova, L., SEO SPAM AND MALWARE, 2019
[10] Petkova, L., SECURITY'S LEAKS IN SEO SPAMMING, 2019
[11] ICT Security Trends, Willian Dimitrov, Sofia, 2017, Avangard, ISBN 978-619-160-766-2
[12] Software testing, Willian Dimitrov, Sofia, 2017, Avangard, ISBN 978-619-160-765-5
[13] ICT Security Model, Willian Dimitrov, Sofia, 2018, Avangard, ISBN 978-619-160-950-5