Official Umbraco Whitepaper

Umbraco Security

At Umbraco, we have processes and precautions in place to ensure high levels of security and testing are continually met across all our products and add-ons. We test and review code internally and externally (via 3rd party audits), we conduct internal security training and follow security experts online to ensure we are always up to date, and have rigorous password and authentication workflows enforced on the Umbraco backoffice to protect access. Security alerts are published publicly on the Umbraco blog as required and if you need to report a security vulnerability you can do so to security@umbraco.com.

Umbraco CMS

We take the following precautions while writing code for Umbraco:

General

- Every six months a security firm helps us do a penetration test to make sure we didn't drop the ball
    - Testing focuses both on new features, and randomly selected areas for the security firm to poke around in to see if it could be vulnerable
    - We will implement any recommended fixes and share what we have fixed with the community for you and your clients to consult
    - If there's any critical issues we will immediately share a blog post on umbraco.com/follow-us and create patches
- Internal code reviews are done consulting the OWASP site for best practices when any code affecting security is updated
    - We use parameterized queries, a standard best practice writing of code, in order to prevent SQL injection attacks on data, such as tampering with, changing, deleting or acquiring unauthorised access to data. This means our code has inherent best practice security measures in place.
- We have several base classes we can inherit from to make sure that code will only be executed if there's a valid backoffice user logged in. This security measure ensure executions are only performed by permitted and authorised users.
- When critical security bugs are found we patch all versions that we can, even back to very old versions, in order to help keep your projects safe regardless of which version of Umbraco CMS you are running
    - Make sure to follow @umbracoproject on Twitter for updates.

  - There's multiple IFTTT recipes available for people not on Twitter to get tweets as emails, RSS feeds, Slack messages etc.
- Umbraco has dependencies on other software, and we keep dependencies updated as much as we can to benefit from any security fixes they might release
- We do internal training at Umbraco HQ to make sure everybody is up to speed on the latest security developments
- At Umbraco HQ we follow security experts on Twitter, like Troy Hunt and Scott Helme. We also religiously listen to the Security Now podcast and follow all of Troy Hunt's excellent security courses on Pluralsight.

Backoffice security

- Since Umbraco version 7.3 we use ASP.Net Identity as the authentication mechanism for back office users. This implementation wraps ASP.Net Membership APIs but can easily be changed to use the native ASP.Net Identity standards and/or be extended to use any custom standards for storing and validating passwords. For the front-end Umbraco uses the ASP.Net Membership APIs for member security but ASP.Net Identity can be used with the UmbracoIdentity plugin if required and again any of this can be extended to suit whatever needs there may be. Passwords by default are hashed with HMAC-SHA256 and the salt is 128bit (unless you have useLegacyEncoding=true , then the standard is lower but security can be ensured if you turn legacy encoding off).
  - Umbraco also supports a full OAuth login system which means if you want to store credentials in a 3rd party system like Azure Active Directory, Identity Server or any OAuth compliant service, this is certainly possible and you can have full control over the OAuth data flow. Of course HTTPS is fully supported and should certainly be used.
  - All of that is available for both front-end membership and the backoffice users
- You can implement any membership / ASP.NET Identity provider you want for both front-end and backoffice if you don't want to rely on our implementation
- By default, 10 incorrect login attempts will lock out the backoffice user to avoid brute force attacks. We currently only support indefinite lock out but currently do not support a lock time frame - ASP.Net Identity supports this so you could extend the current implementation to achieve this. The attempt count is configurable, you can extend this however you like.
- Password rules (length, character diversity, etc.) are configurable from the web.config
  - This MSDN article lists the available configuration options that can be changed from the web.config file
- Password expiry and two factor authentication are currently not support, but could be custom implemented, as an alternative an OAuth provider could help you support that

- Password security questions ("what was the name of your first dog?") are not supported and are considered insecure
- Umbraco 7.5+ includes a password reset option on the login screen
- By default users are logged out of the backoffice after 20 minutes of inactivity, this can be configured to be shorter/longer

Historical security alerts

We've sent out the following very high priority security alerts in the past:

Umbraco CMS

March 2016

https://umbraco.com/follow-us/blog-archive/2016/3/1/major-security-vulnerability-patched-in-umbraco-versions-450-through-4711/

July 2014

https://umbraco.com/follow-us/blog-archive/2014/7/21/security-issues-found-in-umbraco-4-6-and-7/

May 2014

https://umbraco.com/follow-us/blog-archive/2014/5/23/security-update-one-more-major-issue-fixed-in-470-through-4711/

May 2013

https://umbraco.com/follow-us/blog-archive/2013/5/1/security-update-two-major-vulnerabilities-found/

April 2013

https://umbraco.com/follow-us/blog-archive/2013/4/29/security-vulnerability-found-immediate-action-recommended/

November 2012

https://umbraco.com/follow-us/blog-archive/2012/11/14/security-update-for-4100/

ClientDependency

ClientDependency is a plugin that Umbraco uses in all installations that handles CSS and Javascript minification and bundling.

February 2015

https://umbraco.com/follow-us/blog-archive/2015/2/5/security-alert-update-clientdependency-immediately/

Umbraco Forms

Umbraco Forms is an optional plugin for Umbraco that enables editors to build dynamic, custom forms with little technical experience.

January 2016

https://umbraco.com/follow-us/blog-archive/2016/1/27/umbraco-forms-security-notice/

Protecting your sites

- It is becoming easier and easier to run your site on HTTPS, even if your hosting provider does not support it and we strongly recommend doing that. Cloudflare has a free offering that is easy to configure.
    - Our colleague Sebastiaan blogged about securing your site even further. The biggest takeaway: if you run your site over HTTPS, set "umbracoUseSSL" to "true" in your web.config
- Upgrading Umbraco is easier than ever before, keep up to date and benefit from all the smaller security fixes we put in all the time
- You should audit 3rd party plugins you install in Umbraco. Most of them are open source so they can be inspected
    - We don't know of any malicious plugin that currently exists or has ever existed
- Any user that can log into the back office should not have more privileges than they need, so an editor should not have access to the "developer" and "settings" section.
- Make sure that your error handling is not leaking application information - in web.config set compilation debug to false, turn of tracing and turn customErrors on ("remoteOnly" or "on")

3rd party auditing

We receive security / penetration test reports every few weeks from web agencies that have had their site tested by various security firms. We always immediately implement recommendations where needed. If you have a 3rd party auditing your Umbraco site then we're happy to hear from you on security@umbraco.com with any findings so we can fix what's necessary.