ALLAN THRAEN | 1 years ago | PDF |

.NET Development CMS Optimizely (Episerver)

CHRISTMAS COUNTDOWN: #5 SURE, OUR SERVERS ARE LOCKED UP TIGHT IN THE BASEMENT!

Securing your website is as important a topic as it is large and complex. In this post I will not go into too many details, but highlight a few problems I often see in Optimizely/EPiServer CMS implementations.

The 2023 Christmas Countdown: 12 Common Pitfalls in Optimizely CMS - and how to avoid them

> *This blogpost is part of the 2023 Christmas Countdown series where I each day for the last 12 days before Christmas go through my Top 12 list of the most common and dangerous pitfalls I typically see in Optimizely (EPiServer) CMS 12 Implementations. If you want to learn more and perhaps have your own site evaluated feel free to* reach out to us *here at CodeArt.*
> Here are links to all the posts in this series as they are published:
> 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

Season's greetings, festive developers! Today, as we trim the digital tree, let's illuminate the hidden vulnerabilities within Optimizely (EPiServer) CMS implementations. Grab your holiday cheer and join us on a journey to fortify our digital gingerbread castles against the grinches of cyber threats!
Here are some of the most common (and easily fixable) security issues I often see:

1. **Optimizely Roles: The Admin Access Jingle**

   - *Pitfall:* Decking the halls with full admin access for all can turn our digital wonderland into a chaotic winter market.
   - *Example:* Sir Trusts-a-Lot gifts unlimited power to every elf in the workshop. Now, even the littlest helper can cause havoc in the North Pole. Establish a hierarchy of roles and permissions, limiting access to production, for a silent night of order.

   While its great to show your staff that you trust them, It's not ideal that everybody has the power to publish content in production, or that unexperienced content creators can access many admin features they dont fully understand - just as developers as a rule should be limited to development/testing sites - as the temptation to do live testing on production can be too big.

2. **Connecting CMS to Company AD: The Active Defense Sleigh**

   - *Pitfall:* Relying on default AspNet authentication is like leaving the chimney open for unexpected visitors.
   - *Example:* Lord Forgetful forgets to revoke access when a loyal reindeer changes allegiance. Integrating the CMS with the company's Active Directory ensures a synchronized and fortified defense against unauthorized access, keeping our digital sleigh rides secure.

   If your website is using the default authentication, do yourself a favor; log in to admin mode and list all the users - and see how up to date that list is.

3. **Custom Editor Tools: The Temptation of Quick Fixemas**

   - *Pitfall:* Creating debug or editorial tools on unauthorized controllers is like leaving Santa's secret workshop unguarded.
   - *Example:* Lady Haste hastily deploys a debug tool accessible via a 'secret' URL. Little does she know, the security obscurity is no match for the prying eyes of digital grinches. Keep your tools locked away from unauthorized access to avoid unintended bah-humbugs.

   Keep your tools locked away from unauthorized access to avoid unintended consequences. Ideally by securing them in the /EPiServer/ (or whatever the UIUrl is) and placing the views in _protected modules.

4. **Missing Content Security Policy: Strengthening the Nutcracker Suite**

   - *Pitfall:* Neglecting server-side settings like CORS and proper HTTP headers leaves our Nutcracker Suite vulnerable to a digital snowstorm.
   - *Example:* Count Oversight skips the setup of Content Security Policy, leaving the Nutcracker Suite susceptible to cross-site scripting attacks. Fortify your defenses with the right server-side configurations to prevent a dance of malicious sugarplum fairies.

A good idea can be to head over to securityheaders.com - remember, it's ok to not follow all their recommendations - but think about which applies to you and make a conscious decision.

5. **Form Submissions and Uploads: Safeguarding the Royal Jingle Bells**

- *Pitfall:* Overlooking the security of form-upload folders is like leaving the royal jingle bells exposed on the front porch.
- *Example:* Sir Neglect-a-Lot forgets to secure the folder housing sensitive letters to Santa. Resumes and wish lists now float around like confetti at a royal holiday ball. Implement robust security measures to ensure these treasures remain within the castle walls, safe from digital Grinches.

And even more concerning - for an experienced hacker, the ability to potentially upload code/html that can then be called is an open invitation.

Obviously I'm not even touching on the basic stuff here like avoiding SQL injection vulnerabilities, XSS vulnerabilities and regularly have the site scanned/checked by a pentesting service.

Conclusion: As we navigate the intricate landscape of Optimizely (EPiServer) CMS implementations, let us not forget the specifics that can make or break our digital defenses. By addressing these often-overlooked security pitfalls, we fortify our castles, ensuring they stand tall against the relentless onslaught of cyber threats. Onward, noble developers, to a realm where security reigns supreme!

.NET Development CMS Optimizely (Episerver)

CodeArt ApS
Teknikerbyen 5, 2830 Virum, Denmark
Email: info@codeart.dk
Phone: +45 26 13 66 96
CVR: 39680688