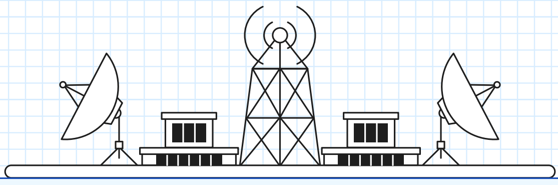
storyblok

Experts Explain: The Blueprint of a Secure CMS

Your CMS is the first line of content security. It's essential – but it's also complex. That's why we turned to the experts.

Read through this infographic to learn the blueprint of a secure CMS and hear tips from our Storyblok Partners on how you can build a safe system too.



The Comms System

The Guard Handbook: Clearly Defined Roles, Permissions, and Workflows

"A CMS needs clear, well-defined roles that map directly to how content is created, reviewed, and published. If permissions aren't tightly tied to workflows, it's easy for people to have more access than they should."

Braden Mears, Senior Solution Architect, Aaxis Digital

"Look for the ability to create customizable review workflows with configurable gates along with warnings about potentially insecure content. Plus, quick and precise communication when any security issue arises with the platform itself is a must."

Dane Morgridge, Principal Architect, RDA

Guard Education: Team Training

"We reduce human error with approval workflows, separation of duties, and targeted training for admins and editors. We prevent workflow bypass with governed release pipelines and read-only production for routine content changes. The biggest difference comes from treating content governance as a security control: custom workflows with required review and approval steps, role-based permissions mapped to real responsibilities, clear release pipelines that prevent policy bypass in production, and regular training, access audits, and cross-team incident readiness across content, engineering, and security teams."

Kile Lindgren, Solutions Architect & Storyblok MVP, BizStream

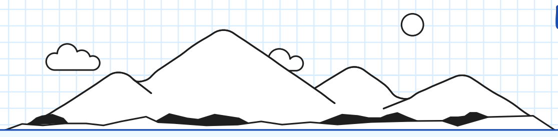
"Equally important [to continuous monitoring] are clearly defined responsibilities, documented processes, and a strong security awareness across the entire team — from developers to content editors. Only when security is treated as an ongoing discipline can a CMS be operated safely and sustainably over the long term."

Manuel Bolai, Manager, Frontend Engineering with diva-e

The Alarm: Incident Reporting Paths

"Seek out operational transparency through auditability, monitoring, and clear incident reporting & escalation paths."

Kile Lindgren, Solutions Architect & Storyblok MVP, BizStream



Beyond the Fortress

External Contractors: Plugins & Integrations

"Most issues don't come from the CMS itself, but from how it's configured and extended over time. Permissions tend to sprawl as teams grow, plugins get added and forgotten, and integrations are set up quickly to meet deadlines. Plugins are a common risk because they often inject JavaScript and can be manipulated if they're out of date. Webhooks and publish events are another area where small misconfigurations can lead to data being exposed or triggered in unintended ways. These risks are usually reduced through regular access reviews, keeping extensions tightly controlled and up to date, and treating integrations as part of the security surface - not just plumbing."

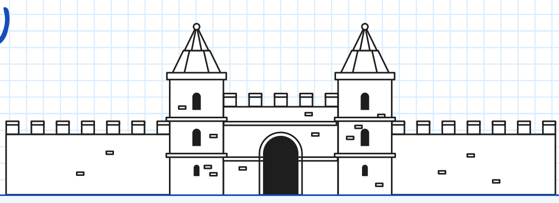
Braden Mears, Senior Solution Architect, Aaxis Digital

"We reduce integration and dependency risk through integration review standards, token and secret rotation, dependency monitoring, and recurring penetration testing with follow-through remediation."

Kile Lindgren, Solutions Architect & Storyblok MVP, BizStream

"Not applying critical security patches is almost guaranteed to result in the compromise of a CMS, sooner or later. Some popular CMSs leverage many third-party plugins or modules and have had a long history of compromises. It is best to leverage automated upgrades and trusted partners wherever possible."

Adam Pupek, Nick Svanascini, and Wes McChristian, IT Security Team, American Eagle



The Gatehouse

The Drawbridge: APIs

“Properly securable APIs are crucial for headless operations. Enabling options like non-interactive application OAuth flows for systems that must be accessed over the public internet. Users also need the ability to lock down API access to specific IP addresses and the ability to revoke those OAuth sessions if a given connected system is compromised.”

Dane Morgridge, Principal Architect, RDA

“Secure access control via access tokens is essential. In a headless setup, access tokens are the primary mechanism for protecting content APIs. Since all content is delivered through these APIs, improper token handling can easily expose large parts of the content repository.”

Manuel Bolai, Manager, Frontend Engineering with diva-e

The Credentials Confirmation: Authentication & Input Security

“Uploading malicious script files, which are often called webshells, is a common way to compromise websites and CMSs. Any CMS that accepts file uploads, especially from non-administrators, needs to have a hardened upload mechanism. This is required to prevent malicious actors from uploading software to take control of the web server. Defense-in-depth, such as leveraging anti-malware or endpoint detection and response software is also useful here in case a malicious script is uploaded. Those tools can help prevent the malicious files from ever executing and track any activity for those that could not be immediately prevented.”

Adam Pupek, Nick Svanascini, and Wes McChristian, IT Security Team, American Eagle

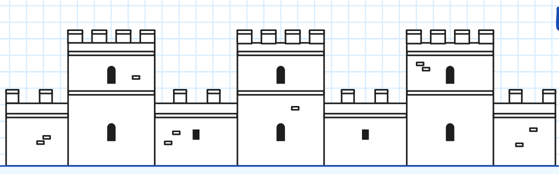
“Inability to properly lock down inputs for validation or sanitization is a risk. This is especially critical for WYSIWYG (what you see is what you get) editors. Warnings about JavaScript in the WYSIWYG and the ability to lock down based on tags, etc. is often something I see missing.“

Dane Morgridge, Principal Architect, RDA

The Guards: Team Vigilance

“What really keeps a CMS secure is how people work with it every day. Teams that take the time to explain why permissions exist, how workflows are supposed to be used, and what to watch out for tend to avoid a lot of problems. It also helps when there’s a clear owner who understands both the technical side and the content side of the CMS. Many issues show up during migrations, new integrations, or staffing changes, so having plans, checklists, and a way to roll back changes makes those moments far less risky. When something does go wrong, teams that have talked through how they’ll respond usually recover faster and with less disruption.”

Braden Mears, Senior Solution Architect, Aaxis Digital



The Outer Wall

Thorough ID Check: Role-Based Access Control (RBAC) and Permissions...

“No real administration of users and roles is a risk. You know it is not truly a secure CMS when every previous vendor appears in the list of users. And they are all active.”

Dane Morgridge, Principal Architect, RDA

“The main risks we keep seeing are over-privileged/misconfigured roles, insecure integrations, and gaps in logging/monitoring that let incidents slip through. We solve these by applying a strong, standardized SDLC with security-by-default patterns rather than one-off fixes.”

Nicolas Hotterbeekx, Principal and Architect Lead at AE

“One of the most critical security features is a granular permission model. This includes permissions on folder and page level, as well as field-level permissions. This ensures that users only have access to the content and functionality they actually need, reducing the risk of accidental changes and protecting sensitive data from unauthorized access.”

Manuel Bolai, Manager, Frontend Engineering with diva-e

“Enable MFA for account verification and authentication; granular RBAC policies and a POLP strategy when assigning access; automated updates to core functionality, especially when security flaws are being addressed.”

John Reiner, Principal Architect, RDA

...and Multi-Factor Authentication (MFA) & Single Sign-On (SSO)

“Prioritize MFA for all direct users and fine grain roles with full audit history. The bigger a system gets, the more there is a need to give a certain set of users a very specific level of access. Having support for things like passkeys is great also. The ability to use external identity providers like Microsoft/Google is important. Don’t forget clearly outlined support processes for if a user account is compromised including that policy, the kinds of information that would be required to recover an account in the event of a breach.”

Dane Morgridge, Principal Architect. RDA

“MFA is becoming a necessity out of the box. This drastically reduces the risk of compromised accounts and phishing efforts.”

Rob Sanders, Director of Digital Experience & Commerce, RDA

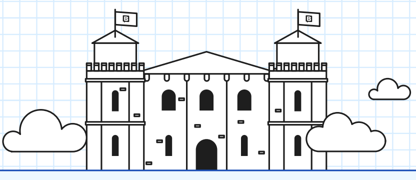
“We mitigate identity risk by enforcing SSO and MFA, applying least-privilege RBAC, recertifying access regularly, and deprovisioning users quickly.”

Kile Lindgren, Solutions Architect & Storyblok MVP, BizStream

The Keys: Credential and Token Management

“One of the most common risks is the exposure of access tokens in client-side applications. These tokens often have permission to read large amounts of content, so once exposed, they can be abused to extract data. To mitigate this, access tokens should never be used directly in the frontend. Instead, they should be secured behind a middleware or a Backend-for-Frontend (BFF). The frontend communicates only with this layer, which controls access, applies additional authorization logic, and returns only the data that is actually required. This approach ensures that content remains protected even if the client application is compromised.”

Manuel Bolai, Manager, Frontend Engineering with diva-e



The Command Center

The Records Room: Logs & Backups

“A CMS must provide detailed, exportable audit logs for traceability. Backups need to be reliable, with proven restore testing.”

Nicolas Hotterbeekx, Principal and Architect Lead at AE

“Not having a recovery plan is a risk. For a large corporation or site providing valuable on-demand information, your site going down could be more costly than having a recovery plan. Mitigate with automated backups, disaster recovery, but also schedule and stick to it for regular restoration testing so you’ll be ready when it happens.”

Rob Sanders, Director of Digital Experience & Commerce, RDA

“You need to have detailed security logs from a load balancer or web application firewall. Also, logs from the operating system and/or an endpoint detection and response solution. However, good logging at the application layer completes the picture. These logs should include not only user activity, but API activity as well. Without good logging, you cannot do adequate monitoring.”

Adam Pupek, Nick Svanascini, and Wes McChristian, IT Security Team, American Eagle

The Vault: Encryption

“On the data side, encryption in transit and at rest is table stakes, especially when the CMS is handling PII or content flowing across multiple systems through APIs.”

Braden Mears, Senior Solution Architect, Aaxis Digital

“Encryption and resilience are essential. This includes encryption in transit and at rest, mature incident response, backups, and continuity planning.”

Kile Lindgren, Solutions Architect & Storyblok MVP, BizStream

The Rule Book: Documentation & Change Management

“Have as much documentation about how the CMS security is designed as possible. We get questions like this from clients from time to time, and it’s important that we are able to answer those questions quickly and factually. Having it also helps us have confidence that the system we are recommending to our clients is secure.”

Dane Morgridge, Principal Architect, RDA

“Make sure all changes go through review and testing. A majority of the issues we see come from rushed deployments or misconfigurations leading to major frustration.”

Rob Sanders, Director of Digital Experience & Commerce, RDA

The Stamp of Approval: Certifications

“One non-negotiable is a proven security posture, including SOC 2 and ISO 27001 certifications, plus willingness to complete security questionnaires, evidence reviews, and penetration testing.”

Kile Lindgren, Solutions Architect & Storyblok MVP, BizStream

“It’s critical to maintain industry-approved security standards (e.g. ISO, SOC 2). These certifications signal to the market that the CMS technology has undergone rigorous process examination and testing. It lets your users know that you take security seriously and are willing to submit to the diligence required to validate best practices.”

Adam Pupek, Nick Svanascini, and Wes McChristian, IT Security Team, American Eagle

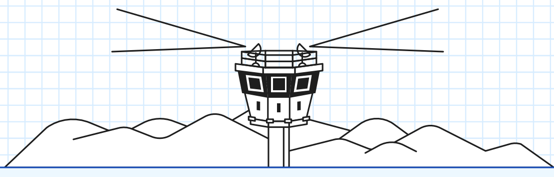
Central Management: Environment Configuration

“Having a central security area that can manage roles for users along multiple environments is helpful too, rather than a basic one at a central point (are you user/admin and what instances can you use) and then more specific on the instance itself…it creates a decoupling that can be harder to manage if you need to remove someone’s access.”

Ken McAndrew, Senior Consultant, RDA

“Ensuring that debugging features like exposing the YSOD are not turned on in production. Ensure that features that might allow someone to explore GraphQL, for example, are only available via the admin. This is probably all handled best by environment variables.”

Ken McAndrew, Senior Consultant, RDA



The Watchtower

The Searchlights: Access Reviews and Security Scans

“Regular access reviews can detect unexpected accounts. Those accounts could potentially be added by an external attacker or an existing administrator creating a backdoor account. This process can also detect accounts that should have been removed such as for a past administrator that no longer should maintain access to the CMS. The fewer administrative accounts exist in the CMS, the better. You can also try to compromise it with regular security scanning and penetration testing….The CMS should also be included in any annual penetration testing activities so that it is tested in a more focused manner.”

Adam Pupek, Nick Svanascini, and Wes McChristian, IT Security Team, American Eagle

“Making the CMS unavailable from the public Internet is key. Utilizing a VPN is a good start. If that is not possible, excluding unauthorized IP Addresses would be second. If the CMS is publicly available, securing it with MFA is a must.”

David Gregory, Senior Consultant, RDA

The Repair Team: Patches and Updates

“Lack of patching is one of the biggest risks to the security of a CMS. Once security notifications, or patches are released, it starts an arms race between attackers and defenders. A mature process that implements patches to the core CMS and any plugins or modules makes a huge difference in the overall security of the CMS.”

Adam Pupek, Nick Svanascini, and Wes McChristian, IT Security Team, American Eagle

“Security patching has to be timely with clear SLAs and a solid vulnerability process.”

Nicolas Hotterbeekx, Principal and Architect Lead at AE

The Mindset: Constant Vigilance

“CMS security is not a one-time task but a continuous process. Even the best technical safeguards require ongoing maintenance and review. Key practices include regular monitoring and continuous adjustments of security measures, keeping dependencies up to date, and consistently using the latest Storyblok APIs.”

Manuel Bolai, Manager, Frontend Engineering with diva-e

The Foundation

“When we talk to enterprises about CMS security, one thing becomes clear very quickly: security isn’t a feature, it’s a foundation. If it’s not built in from day one, everything else is a risk multiplier.”

Manuel Bolai, Manager, Frontend Engineering with diva-e

Create with Joy.
Scale with Intelligence.

Discover Storyblok’s ISO27001 certified CMS today.

VIEW OUR INTERACTIVE DEMO

CREATE A FREE ACCOUNT

Copyright © 2026 Storyblok. All rights reserved.