

DOI: 10.1145/1610252.1610284

BY GANESH VAIDYANATHAN AND STEVEN MAUTONE

Security in Dynamic Web Content Management Systems Applications

THE PROCESSES BEHIND CORPORATE EFFORTS TO CREATE, manage, publish, and archive Web information has also evolved using Web Content Management Systems (WCMS). WCMS allow teams to maintain Web content in a dynamic fashion through a user friendly interface and a modular application approach. This dynamic “on-the-fly” content creation provides Web site authors several advantages including access to information stored in databases, ability to personalize Web pages according to individual user preferences, and the opportunity to deliver a much more interactive user experience than static Web pages alone.¹¹ However, there are distinct disadvantages as well. Dynamically generating Web content can significantly impact Web server performance, reduce the scalability

of the Web site and create security vulnerabilities or denial of service.¹¹ Organizations are adopting information technology without understanding such security concerns.¹ Moreover, as Mostefaoui⁷ points out, even though many attempts have been made to understand the security architecture, a generic security framework is needed. Recent research amplifies the concerns and benefits of security in open source systems.² However, there is a need for organizations to understand how to evaluate these open source systems and this paper highlights how an evaluation technique in terms of security may be used in an organization to assess a short list of possible WCMS systems. This article focuses on security issues in WCMS and the objective is to understand the security issues as well as to provide a generic security framework.

The contributions of this paper are to:

1. Integrate the goals of security with eight dimensions of WCMS,
2. Specify how to secure the eight dimensions of WCMS,
3. Formulate a framework of security using this integrated view of security goals and security dimensions, and
4. Address the security of the Web architecture at WCMS software application level using the framework and evaluate security features in popular WCMS used in the industry.

Web Content Management Systems

Content management in Web sites is a combination of quality information, formal processes, and supporting systems architecture that help organizations to dynamically contribute, collaborate and control page elements such as text, graphics, and multimedia. This heterogeneous nature of content is managed by WCMS. Currently, there are countless WCMS software packages on the market that provide some sort of content management. The most basic advantage of a WCMS is the consistency that it provides to the design, layout, and organization of

a site, bringing everything together in a uniform package.

The WCMS market is currently divided among two main classifications. The first is open source licensing and the second is owned or leased licensing. Both categories of the market are at comparable stages in development as far as features are concerned. One main difference between the two categories is that open source packages tend to emphasize a more general “global” approach to content management while the licensed packages aim at a specific business goal such as collaboration. Licensed software development is usually under centralized control within a company and has a higher initial cost though it usually has advantages such as support and training in comparison to open source software.⁵ According to the open source organization, open source doesn’t just mean access to the source code, its derived works, and the modifications. It also involves the distribution terms of open-source software. Since open source systems are module based, adding new features to a Web site is very easy. These packages are also supported by several “third party” software that provides extensions, templates, themes, and updates. Third party in open source would mean supporting external open source communities. In organizations, many open source systems operate with extensive communities.

Both types of WCMS packages allow different employees to see different levels of data, therefore taking corporate information management to the next level by meeting some managerial decision making goals as well. The flexibility of Web coding and Web interfaces allows a company to customize the system to their specific user’s needs. It is very important to stress the value of investigating both open source and licensed packages in order to see which strategy and development best suits organizational needs.

Security in WCMS

The main attributes of information security are confidentiality, integrity, authentication, availability, and non-repudiation.^{4,6} Confidentiality is to ensure that information is not accessed by unauthorized persons. Integrity is to ensure that data and applications are safe from modification and modi-

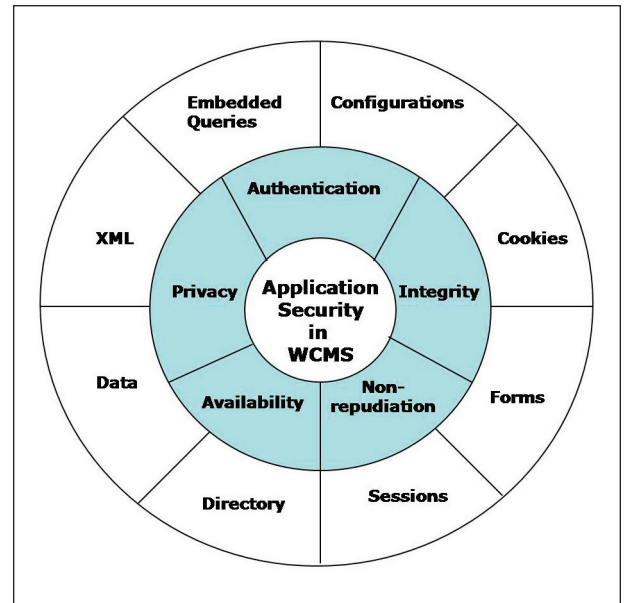
fied only by authorized users. Authentication is to ensure that origin of content is correctly identified with an assurance and that the identity is not false. Availability is to ensure that end systems are available as and when needed to authorized users. Non-repudiation is the security assurance that neither sender nor receiver be able to deny transmission of content. To ensure security, organizations need to integrate these security attributes in system design, software applications as well as other system components and operations.

If a Web system is built without security as one of the main goals, then it will ultimately require more security patches and updates in order to continually fix newly found flaws. It is easy and relatively inexpensive to provide a sufficient level of security for most applications using WCMS due to the basic nature of open Web code.⁸ However, proper security measures must be addressed in programming techniques, server configurations, and programmed module configurations. Security is always best built from the ground up through the design into the implementation.

Security Framework at WCMS application software level

WCMS at the application level has much vulnerability. The vulnerabilities to WCMS include interruption, interception, fabrication, and modification.⁹ Interruption means an asset of the system becomes lost, unavailable, or unusable. Interception is when an unauthorized party gains access to an asset. An attack on the integrity of data and directory by an unauthorized party may constitute modification. An unauthorized party may gain access and fabricate counterfeit objects on a network as well. Data, directory, forms, configurations, and sessions may become vulnerabilities in WCMS security while cookies, embedded queries, and XML communication

Figure 1. Framework of Security in WCMS Applications



may be exploited. These eight dimensions form the functional dimensions in this paper. Security for WCMS software applications need to be treated with a framework in mind. The five goals of security that include integrity, privacy, authentication, availability, and non-repudiation when mapped to the above mentioned functional dimensions of WCMS gives rise to the framework as illustrated in Figure 1.

This framework integrates the eight functional dimensions of WCMS with the five goals of security.

Integrity, the goal that ensures the correctness or accuracy of the application and its contents, addresses the modification and fabrication security threats of data, directories, forms, stored cookies, sessions, embedded queries, configurations, and XML communication. Privacy that assures unauthorized access or viewing ensures no interception threat on the same eight features. Authentication, the goal that ensures the validity of the claimed identities of the entities participating in communication, addresses the fabrication vulnerability of the eight features of WCMS. Availability addresses the interruption vulnerability and ensures that there is no denial of authorized access to data, directory, stored cookies, sessions, and forms. Non-repudiation, the goal that provides proof of the origin of data or the cause of an event or an action, ensures the availability of evidence that can be used to

prove that some kind of event or action has taken place so that the cause of the event or action cannot be repudiated later and addresses fabrication vulnerability. Thus, the eight functional features of security can be mapped to standard five attributes of security.

Configurations. WCMS must be properly configured on a server in order to ensure top level security. It is necessary to secure the operating system with proper patches and security updates as well as the Web server with proper configuration and updates. Global settings need to be incorporated and used in programming current Web system architectures. Specifically *register_globals* and more generally *error logging and reporting* are two areas of configuration directives that pose security vulnerabilities. If the *register_globals* directive is not set within applications then global variables can be changed or affected by an attacker through a basic query. Security measure is to develop and deploy applications with *register_globals* disabled. Error logging and reporting configurations need to be set for both development and production. In the case of displaying errors, *display_errors* need to be set to on for development and set to off for production, so that errors are hidden from the users and potential attackers. The directive that determines whether errors should be written to a log, *log_errors* should be set to on in production. The directive that indicates the location of the log file to which errors are written, *error_log* should be ensured that the Web server has write privileges for the specified file. Both *error_reporting* and *error_reporting* should be set to *E_ALL* in order to enforce the initialization of variables since a reference to an undefined variable generates a notice.

Cookies. Cookies may allow a malicious user to hijack Web sessions and view, modify or otherwise exploit the information related to another user's session by physical access or network sniffing. The security measures include using non-persistent cookies and if persistent cookies are used, short duration of cookie life need to be specified. Another security measure is to transmit cookies in a secure communication and to encrypt the information in the

cookies. When using cross-site scripting (XSS), the compromised output may contain malicious script prepared by an attacker and delivered on behalf of a Web server. The malicious script is granted the same trust level as the Web server, and as a result output integrity is broken.³ Third-party products like AppShield may be used to detect and mitigate this vulnerability.]

Forms. Poorly designed Web applications may contain hidden fields that contain information of users, accounts, and sessions. Manipulation of form by malicious users may result in unexpected application behavior, accessing defunct applications, incomplete database records, or buffer overflow. A good software design is to ensure that data filtering cannot be bypassed and that invalid data cannot be mistaken for valid data, and to identify the origin of data. For example, Web systems need to check customers' addresses before validating the credit card for a purchase, or even more basically validating users' email addresses before accepting them onto a newsletter subscription. Security measures include validation of form input field values and performing referrer checks on the server side.

Embedded Queries. A malicious user may input additional field entries in hijacked forms to receive confidential information about the user and their accounts. Security measures include carefully examining database queries for wild characters, validate and ensure that input fields contain only the relevant user-related information, and ensure proper permissions exist on the database objects accessed by the Web application.

Sessions. Session security is often the target of attack on a system. The most crucial piece of information for an attacker is the session identifier. An attacker can simply initialize a session for a particular session identifier, and then use that identifier to launch the attack. Session identifiers are typically saved in cookies. Session fixation is a method that tricks a victim into using a session identifier chosen by the attacker and if successful, it represents the simplest method with which a valid session identifier can be obtained. Regen-

erating the session identifier whenever there is any change in privilege level will protect against this type of attack and will eliminate the risk of a successful session fixation attack. Session hijacking refers to all attacks that attempt to gain access to another user's session. By requiring a valid session identifier and the correct user-agent header that is associated with the session, security can be made to step up a level. Security measures include ensuring consistent and appropriate session timeouts for the application as malicious users may be able to hijack another user's session under long timeouts.

Directory. Automatic directory listing in servers revealing important application files provide malicious users with a lot of information that may be used against the system. Security measures include disabling directory browsing, establishing and implementing good file management process. To be safe, unwanted files should be removed from document root entirely.

Data. WCMS applications interact with a database. Using a file called *db.inc* to connect to a database with access credentials, authenticating users is very simple. Security measure is to place all modules outside of document root and not make modules accessible. Protecting against structured query language (SQL) injection requires filtering data. Using input filtering method, where all incoming data are validated and any invalid data are prevented from being used by the application, many security concerns may be greatly reduced or eliminated. This command injection attack poses a serious threat to Web application security and can be validated using *SQLCHECK*, an implementation for the setting of SQL command injection attacks.¹⁰

XML Communication

Extensible markup language (XML) is used as a cross-platform and cross-application document format and when coupled with its self-describing attributes, makes it irresistible for solving new data-sharing problems. XML communication has a real security problem and many companies including Entrust, Microsoft, RSA, and VeriSign have tried to layer encryption and digital

Table 1. Evaluation of two WMCS Applications

Functionality	Security Goals	Features	Mambo	vBulletin
Configuration	Av, Au, I, C	Register_globals settings	Set using a php ini file	Managed by the install script
	Av, I	Error_logging functionalities and settings	Email error reporting	Email error reporting
	Av, I	Reporting functionalities and settings	Controlled in the Admin Control Panel list; If operations system emails error to Admin	Controlled by the Admin control panel list
	Av, I	Patches	Patches are downloaded from server and uploaded to the specific site via FTP; No auto uploads; Most 3rd party packages will be notified	Admin control panel shows update notifications with a direct link to the download; No auto uploads; 3rd party packages are not notified
	Av, Au, I, C	Security updates	Security updates are downloaded from server and uploaded to the specific site via FTP	Admin control panel shows update notifications with a direct link to the download
	Av, Au, I, C	Install files (script to create DB tables)	Run installer and delete the installer files; More security notifications than vBulletin	Run installer and delete the installer files; Will notify admin
Cookies	Au, I, C	Non-persistent cookies	Can be changed in php ini file; not integrated into control panel; Edit offline and upload	Cookie settings can be changed in the Admin control panel
	Au, I, C	Duration of cookie life	Default is until browser is restarted; But can be set in the php ini file	Set on install script and can be changed after install through the Admin control panel
	I, NR, C	SSL enabled	Does not utilize SSL, but OSCommerce and other E-Commerce applications can be integrated with SSL	Does not utilize SSL
	Au, C	Encryption facilities	Password hashing	Password hashing
	Au, AV, I, NR, C	Integration to 3rd party security apps	Yes; A module based system and can have many additions made to it's core	Can be integrated at other levels such as the operating system level or web server level
Forms	Av, I, C	Data filtering functionalities	Yes	Yes
	Au, C	Data validation functionalities	Email verification on new accounts, image checking validation for new accounts, password validation	Email verification on new accounts; Login verification "strike out" after three logins within 15 minutes
	Au, C	Referer checks enablement functionalities	None	Post Referrer Whitelist allows data to be submitted via post from within the domain the board is installed on
Embedded Queries	Au, Av, I	Examining database queries	Database can be viewed through SQL or through the Admin control panel	Queries can be viewed created, edited through the Admin control panel
	Au, Av, I	Permissions on the Database objects	Yes	Yes
Sessions	Au, I, C	Modify regenerate session identifier function	No	No
	Au, I, C	User-Agent header functionality	No	No
	Av, Au, I, C	Session timeouts functionality	Set session time for user to be in the system for any period without getting disconnected	Set session time for user to be in the system for any period without getting disconnected
Directory	Au, Av, I, C	Disabling directory browsing functionality	Yes	Yes
	Au, Av, I, C	File Management systems	A full upload, image, content file system is used and accessed through the Admin control panel	Some low level file management is possible; Uploads can have restrictions on file size and type
	Au, Av, I	File Management Control	Controls system files as well as user files	Controls user files only
Data	Au, Av, I, C	User input sanitization functionalities	No clean up data entry; Post anything; No default; Can be added using 3rd party packages	Has built-in facilities; Ban users for spamming based on IP address or username; Can ban block specific words or categories
XML	Au, Av, I, NR, C	XML security	None; Can use 3rd party applications	None

Legend: Au: Authentication; Av: Availability; I: Integrity; NR: Non-Repudiation; C: Confidentiality

signatures on top of XML, but in vain.¹² While some of the security issues may be mitigated by using SSL to hide XML communications over the network, authentication remains to be a bigger security problem with XML communication. Applications that send information via XML documents use the same Internet network protocols that traditional security products monitor. Since XML messages are wrapped in the IP “envelope” that most firewalls are designed to track, corporate networks inspect the envelope but not the contents. Fraudulent XML messages could therefore enter corporate networks undetected. Security gateway appliances, such as Sarvega XML Guardian process encryption of XML files, enforce security policies authorizing access, and generate a log of network activities for auditing purposes, tracking potential hackers.

Evaluation of Framework on WCMS Tools

This framework was evaluated in two of the leading WCMS software applications as illustrated in Table 1. We have chosen Mambo and vBulletin to evaluate the proposed integrated security framework. The reason that these two specific WCMS systems were chosen is because of their overall characteristics in regards to this comparison. Mambo is by far one of the leading WCMS software available today due to its user-friendly and advanced features. vBulletin is a community forum solution which is extremely affordable, reliable, professional, and is effectively instrumental in giving an instant community, a platform where visitors can interact, participate in discussions, get their queries and issues resolved, provide answers and solutions to each other's problems and express opinions. Both Mambo and vBulletin have flourishing communities as well as extremely active sub communities. Their communities consist of mainly smaller organizations and groups.


In general, both applications have most of the security features in place. However, some of the security features need to be implemented using third-party software. Currently, organizations upload content internally and from a centralized location and therefore do not use SSL for uploading content. In the future when organizations encour-

age upload content from various locations directly, more security precautions may become a necessity. In both packages patches and security uploads have no automatic upload capabilities. The PHP files that create database tables, also called the installer files, create the database tables during the first time installation and delete the same files if installed again. As a module based system, Mambo has more features that may be used to install third-party applications to enhance security features. Non-repudiation can be implemented in different ways as long as its properties are satisfied, the two common approaches being one-time password (OTP) dynamic tokens and digital signatures. In general, both packages have most of the features that are needed to satisfy the five goals of security.

Conclusion

This study illustrates a framework for WCMS that can be used to incorporate such systems in organizations. As an example, this study compares two leading WCMS software and firms can use this methodology to understand other software that are available in the market currently and in future. An understanding of these security issues will benefit practically everyone who is interested in open systems. If the Web site is for public use, end users and customers need not be concerned about specific security issues. If the Web site is for internal use, then the security issues discussed in this paper is useful for the system administrators. As corporate Web sites grow, the need to manage their content and features will become exceedingly more difficult. Therefore the need for a better developed dynamic Web content management systems will be necessary. After this analysis using the developed framework, it is evident that all Web site administrators should at least be investigating the possibility of a WCMS to help organize their site. There are so many different packages available, with so many different features, Web development has truly been revolutionized with the growing support for code to help maintain a well developed, user friendly Internet.

All Web architectures need to specifically address the security practices necessary to support the sensitivity of

the data of their particular sites. Security can be implemented at all levels including the application level. This framework and evaluation can provide insights to practitioners of electronic business and information technology managers as well as academicians. 

References

1. Dhillon, G., Backhouse, J. Technical opinion: Information system security management in the new millennium. *Comm. ACM* 43, 7, (July 2000), 125-128.
2. Hoepman, J., Jacobs, B. Increased security through open source. *Comm. ACM* 50, 1, (Jan. 2007), 79-83.
3. Huang, Y., Yu, F., Hang, C., Tsai, C., Lee, D.T., Kuo, S. 2004. Securing Web application code static analysis and runtime protection. *Proceedings of the 13th international conference on World Wide Web*, New York, NY, 40-52.
4. Joshi, J.B.D., Aref, W.G., Ghafoor, A., Spafford, E.H. Security models for Web-based applications. *Comm. ACM* 44, 2, (Feb. 2001), 38-44.
5. Karels, M.J. Features: Commercializing open source software. *Queue* 1.5, (2003), 46-55.
6. Maconachy, W.V., Schou, C.D., Ragsdale, D., Welch, D. 2001. A model for information assurance: An integrated approach. *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, (June 5-6, 2001), West Point, NY: 306-310.
7. Mostefaoui, G.K. Security in pervasive environments, What's next? *Security and Management Journal*, 2003, 93-98.
8. PHP Security Consortium. PHP Security Guide 1.0. <http://phpsec.org/php-security-guide.pdf>, 2005.
9. Pfleeger, C. 1997. *Security in Computing*. Prentice Hall, Upper Saddle River, NJ, 1997.
10. Su, Z., Wassermann, G. 2006. The essence of command injection attacks in Web applications. *33rd ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages*, Charleston, SC, 2006: 372-382.
11. Titchkosky, L., Arlitt, M., Williamson, C. A performance comparison of dynamic Web technologies. *ACM SIGMETRICS Performance Evaluation Review* 31, 3, (2003), 1-11.
12. Vaughan-Nichols, S.J. XML shows promise, but don't underestimate its problems. *News Forge*, June 18, 2003.

Ganesh Vaidyanathan (gvaidyan@iusb.edu) is an associate professor of Decision Sciences at Indiana University South Bend, IN.

Steven Mautone (steven@mautone.us) is a Senior Online Producer for Showtime Networks, Inc. in NY.

© 2009 ACM 0001-0782/09/1200 \$10.00

Copyright of Communications of the ACM is the property of Association for Computing Machinery and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.