2022

4(1), 44-65

İçerik Yönetim Sistemleri ve Veri Koruma Çerçevesinde WordPress Güvenliğinin İncelenmesi

Hüseyin Çakır[1], Murat Taşer*[2]

Anahtar Sözcükler

İçerik Yönetim Sistemleri
WordPress
İçerik
Veri
Güvenlik

Makale Hakkında

Gönderim Tarihi
03 Mart.2022
Kabul Tarihi
19 Nisan 2022
Yayın Tarihi
29 Haziran 2022
Makale Türü
Araştırma Makalesi

Öz

Günümüzde küçük ve orta ölçekli kurumlar ağırlıklı olmak üzere, birçok kurum ve şirket tarafından İçerik Yönetim Sistemleri (İYS) yaygın bir şekilde kullanılmaktadır. Bunun altında yatan gerekçeler incelendiğinde, maliyet-performans ilişkisine dayalı verilerle karşılaşılmaktadır. Fakat bu sistemlerin yaygın bir şekilde kullanılması dijital dünyada dikkat çekmeye başladığı andan itibaren tehdit unsurları da artmış bulunmaktadır. Bu sistemler, bağlantılı güvenlik açıklarından yararlanmak isteyen bilgisayar korsanlarının toplu saldırı hedefleridirler. Düzenli bakım yapılmadığında, risk analizlerine göre bir güvenlik çerçevesi çizilmediğinde, teknik anlamda yeterli olmayan bir şirket veya serbest çalışanlar (freelance) tarafından geliştirildiğinde; WordPress sitesi için aynı sağlamlık ve güvenilirlik geçerli olmayabilir. Bu çalışma için örnek bir Web İYS oluşturulmuş, içerik ve verileri bekleyen tehlikeler ele alınmış, oluşan problemlere yanıt aranmaya çalışılmış, verileri korumak ve riskleri en aza indirgemek için bu yanıtlar ışığında elde edilen bulgulardan faydalanılarak bir sonuç değerlendirmesi yapılmıştır. Özellikle güvenlik çerçevesi üzerinde durularak, veri koruma noktasında neler yapılması gerektiğine dair ana hatlar belirlenmiştir.

Content Management Systems and Review of WordPress Security in Terms of Data Protection

Keywords

Content Management Systems
WordPress
Content
Data
Security

Article Info

Received
March 03, 2022
Accepted
April 19, 2022
Published
June 29, 2022
Article Type
Research Paper

Abstract

Today, Content Management Systems (CMS) is widely used by many institutions and companies, mainly small and medium-sized institutions. When it examine the reasons behind this, It be come across data based on the cost-performance relationship. However, since the widespread use of these systems started to attract attention in the digital world, threats have also increased. These systems are targets of mass attack by hackers who want to exploit the associated vulnerabilities. When it is not maintained regularly, a security framework is not drawn according to risk analysis, it is developed by a company or freelancers who are not technically competent; The same robustness and reliability may not apply to the WordPress site. For this study, a sample Web CMS was created, the dangers waiting for the content and data were discussed, an answer was tried to be sought for the problems that occurred, a result evaluation was made by making use of the findings obtained in the light of these answers in order to protect the data and minimize the risks. With particular emphasis on the security framework, the main lines of what should be done at the point of data protection have been determined.

Atıf: Çakır, H. & Taşer, M. (2022). İçerik Yönetim Sistemleri ve Veri Koruma Çerçevesinde WordPress Güvenliğinin İncelenmesi. *Bilgi ve İletişim Teknolojileri Dergisi, 4*(1), 44-65. https://doi.org/10.53694/bited.1082095

Cite: Cakir, H. & Taser, M. (2022). Content Management Systems and Review of WordPress Security in Terms of Data Protection. *Journal of Information and Communication Technologies*, *4*(1), 44-65. https://doi.org/10.53694/bited.1082095

* **Sorumlu Yazar/Corresponding Author:** mtaser@pau.edu.tr

[1] Assoc. Prof. Dr., Gazi University, Faculty of Education, Department of Computer Education and Instructional Technologies, Ankara, Turkey, hcakir@gazi.edu.tr, https://orcid.org/0000-0001-9424-2323

[2] Phd. Hospital General Manager, Pamukkale University Hospitals, Denizli, Turkey, mtaser@pau.edu.tr, https://orcid.org/0000-0001-6871-4171

Extended Abstract

Introduction

A CMS allows to manage content from a user interface. There are a number of CMS software available with a few clicks of downloads. This software make it easy for a non-technical person to use and navigate within the panel. This eliminates the need for the CMS world to be proficient in JavaScript, HTML, CSS, PHP and MySQL and other similar components. Developing a website with CMS is easy enough to let you choose which parts fit best, write text, add images and graphics directly from the control panel. Websites are created with databases with a secure and easy-to-use interface (Collins, 2003).

Most CMS are managed and updated through new features as web technologies evolve. Web CMS build platform options are plentiful. The most popular is WordPress. The most basic feature of this platform is that it does not require a lot of time to create dynamic websites. Free and open-source WordPress has a template system and plugin architecture. Templates allow developers to build a site on a predefined structure instead of building it from scratch. Plugins provide additional functionality to extend tools. The founders of WordPress were guided by the philosophy that the best software should run with minimal setup. They aimed to make the tools easy to use and intuitive for developers and designers to devote their time to site-building.

CMSs are software platforms that allow its users to perform functions such as creating, editing, archiving, reporting and publishing content. Today, they are widely used by many institutions and companies, mainly small and medium-sized institutions. This widespread use in the digital world has also increased the threat to these systems. In particular, they are the target of mass attacks by hackers who want to exploit related security vulnerabilities.

For most users, it is a question to be answered whether these systems are safe or not, and if it is not a safe environment, whether this situation can be corrected. In addition, "are they easy targets for hackers, what are the risks of cyber-attacks to these sites, which elements should be observed?" appear in questions such as.

From a security standpoint, major CMS systems have the advantage of being open source. Security researchers can access it and test it, which makes it possible to identify vulnerabilities. However, on the other hand, the fact that hackers are in a position to access it can lead to the discovery and malicious use of these security vulnerabilities. Communities around large-scale CMS systems develop plug-ins that add functionality to these systems. While this is an advantage in terms of usefulness, it also comes with risks associated with the integration, configuration and maintenance of these plugins.

Another of the main risks associated with CMS is updates. Numerous plugins and themes are available for core CMS systems. So the updates are not only for the versions of the CMS system, but also for the versions of the various plugins and themes used. New vulnerabilities are constantly being discovered and are immediately fixed with a patch. Therefore, it is necessary to install updates as soon as possible and to frequently check for available patches. Negligence in this matter will seriously increase the risk of data breach on the system used.

The last big risk with CMS that should be mentioned is the competence of the developers. Most CMS-based websites are passed down not only to configuration but also to custom-made developments by an in-house software development team or a service provider. In this sense, issues such as the security skills of the developers and whether penetration tests are carried out should also be questioned.

Method

As the method of the research, the qualitative model was used in general and the document analysis method was used. Documents generally refer to articles, papers, theses, forums on web pages, and articles on the web page related to the subject, which were recorded without any intervention by the researcher. In this context, first of all, the literature was scanned, web pages related to the subject were examined and analyzed in detail.

In the study, security studies were carried out with the help of simulations on a sample WordPress Web CMS and it was aimed to prepare a sample directive to reduce existing risks by using the document review method. At the same time, it is aimed to help individuals and institutions using the CMS platform to identify the main risks, to present the important points that they should pay attention to in order to strengthen the security level of the system, and to create a resource that they will constantly refer to in the relevant processes.

In line with the purpose of the study, Wordpress security has been examined based on IYS and Data Protection. Regarding the measures that can be taken, five main topics and 21 sub-headings have been determined in line with the literature and expert opinions.

Findings

The applications made within the scope of the study show that; Contrary to popular belief, WordPress CMS sites are as secure as an on-demand site. If a well-known and widespread CMS is used around the world, it can be considered a solid solution that is constantly evolving. But when regular maintenance is not done, a security framework is not drawn according to risk analysis, when it is developed by a company or freelance people who are not technically competent; The same robustness and reliability may not apply to the WordPress site.

However, CMS systems run a greater risk of being hacked. These systems are targets of mass attack by hackers who want to exploit the associated vulnerabilities. Major CMS platforms such as WordPress, Joomla, Drupal, Shopify make up 46,8% of all websites. Therefore, attackers are more likely to come across vulnerable sites. The security of the WordPress system and plugins depends on the community, which is often very active for known CMS systems. However, it is necessary to be careful with the add-on maintenance of the system. There is always a risk that the solution will become less popular, less protected or even abandoned at some point.

Discussion and Conclusion

As with any proprietary/known product, security depends on the importance given to it by the company and the knowledge of the development team. The advantage of familiar CMS systems like WordPress is that a team is directly responsible for their development and security. That's why that there is a roadmap we have planned for updates, features to test, items to fix, etc. However, it is also possible that the product will at some point not be supported by the developer and is no longer maintained.

Whether it's an open source or commercially licensed CMS, the choices should be made based on needs. The security of the site depends first and foremost on how that system is managed, how it is configured, and how it is maintained, rather than whether it is based on an open source CMS.

The risks faced by websites, such as data theft, service interruptions, or hosting illegal content, vary depending on the size and functionality of the CMS system. If a well-known and widespread CMS is used around the world, it can be considered a solid solution that is constantly evolving. However, the same robustness and reliability may not be valid if regular maintenance is not performed, a security framework is not drawn according to risk analysis, and when it is developed by a technically incompetent company or freelance people. The solution is to initially perform a web test that can only focus on the major risks, detecting and then detailing step-by-step to fix the vulnerabilities. As a result, sites built with WordPress are also safe, but they have certain requirements to be considered to ensure their security.

Giriş

İYS; kullanıcılarının içerikler oluşturmasına, düzenlemesine, arşivlemesine, işbirliği yapmasına, raporlamasına, yayınlamasına, dağıtmasına ve bilgilendirmesine izin veren bir yazılım platformudur (Vetch & White, 2006). Grafik Kullanıcı Ara yüzü (Graphical user interface-GUI) vasıtasıyla, bir web sitesinin veri tabanıyla kullanıcı dostu etkileşimini sağlar. Rockley, Kostur ve Manning (2003) İYS'yi işbirliğine dayalı bir ortamda iş akışını yönetmek için kullanılan ve prosedürler koleksiyonu sağlayan bir yazılım platformu olarak tanımlar. Bu teknoloji, son otuz yıl içinde ortaya çıkan bir projenin varyasyonlarıdır ve WCMS (Web Content Management System), ECM (Enterprise Content Management), DAM (Digital Asset Management) gibi türevleri mevcuttur. Bu türevler arasında belirgin farklılıklar vardır. Bugün neredeyse herkes sınırlı teknik bilgi birikimine sahip olarak bir web sayfası oluşturabilmektedir.

Web siteleri, sayfalarını oluşturmak ve tasarlamak için HTML (Hypertext Markup Language) ve CSS (Cascading Style Sheets) kullanır. HTML sayfanın yapısını CSS ise görsel ve işitsel düzeni sağlar. Bunlar Web sayfası oluşturmak için temel bazı bileşenlerden sadece ikisidir. (Boiko, 2005; Paulsen, 2012). İYS, herhangi bir kodlama bilgisi olmayan kullanıcıların, bir WYSIWYG (what you see is what you get) ara yüzü kullanarak web sitelerinin içeriğini değiştirmesine ve düzenlemesine olanak tanır. İYS yazılımına girilen veriler, web sayfasını bir şablon aracılığıyla işleyen bir veri tabanında saklanır ve bu sayfanın CSS'i çıktıyı kontrol edebilir (Amsler & Churchville, 2021). Daha büyük bir marka veya kuruluş olduğunda, şirketin kullanım senaryosu için doğru yazılımı bulmak biraz daha zor olabilir. Bu durum kurumu, İYS veya WCMS kullanma noktasında tereddütte bırakabilir.

İYS, genellikle WCMS ve ECM'nin bir üst evreni olarak kabul edilir. Yukarıda belirtildiği gibi İYS, içeriği geliştirmek, düzenlemek, yönetmek ve yayınlamak için kullanılan bir yazılımdır. Bir İYS, belgeler veya veri tabanı kayıtları gibi yapılandırılmış içerikle en iyi şekilde çalışır, aynı zamanda video ve ses dosyaları gibi içerikleri yönetmek için de kullanılabilir (Amsler & Churchville, 2021).

ECM (Enterprise Content Management – Kurumsal İçerik Yönetimi): Bir kuruluşun iş süreçlerini içeriğiyle birleştirme stratejisinin öncülüğünde yazılımsal araçlarını birleştirir. Yapılandırılmış ve yapılandırılmamış içeriği yönetebilir. Kurumsal İçerik Yönetimi, bir şirketin içerik stratejilerinin, yazılımlarının, araçlarının ve ekibinin içeriği etkili bir şekilde yönetmek için bir araya gelmesidir.

WCMS (Web Content Management System – Web İçerik Yönetim Sistemi): Web İYS, birçok sektör uzmanı tarafından İYS'nin bir diğer alt kümesi olarak kabul edilir. WCMS ve ECMS arasındaki çizgi incedir. Birincil farklılaştırıcı faktör, bir WCMS'nin web içeriği için daha uygulanabilir iken bir ECMS' in işlevi ise bütünsel bir iş süreçlerini ifade etmesidir (Kinsta, 2020).

Bir İYS, içeriği bir kullanıcı ara yüzünden (dashboard) yönetmeye olanak tanır. Birkaç basit fare hareketiyle yapılabilen yüklemeyle kullanılabilen çok sayıda İYS yazılımı vardır. Bu yazılımlar, teknik olmayan bir kişinin kullanımını ve panel içinde gezinmesini kolaylaştırır. Bu durum, İYS dünyasının JavaScript, HTML, CSS, PHP ve MySQL vb. gibi diğer bileşenlerde uzman olma ihtiyacını ortadan kaldırır. İYS ile bir web sitesi oluşturmak bir legonun parçalarıyla oynamaya benzer. Sitenizi oluşturmak için hangi parçaların en iyi şekilde yerleşeceğini seçebilirsiniz. Örneğin doğrudan kontrol panelinden metin yazmanıza, resimler ve grafikler eklemenize olanak

tanır. Web siteleri, güvenli ve kullanımı kolay bir ara yüze sahip veri tabanlarıyla oluşturulur (Kohan, 2010).

Çoğu İYS, web teknolojileri geliştikçe yeni özellikleri aracılığıyla yönetilir ve güncellenir. Web İYS oluşturma platformu seçenekleri çoktur ve bunlardan en yaygını WordPress'tir. Wordpress, çok yönlü özellikler, temalar, şablonlar ve eklentiler içeren açık kaynak kodlu bir platformdur ve dinamik web siteleri oluşturmak için çok zaman ayırmak gerektirmez.

Collins (2003), içerik yönetimini yapılandırılmış (veri tabanları) ve yapılandırılmamış (belgeler, e-postalar, video dosyaları vb.) bilgi kaynaklarını sürdürmek, düzenlemek ve aramak olarak tanımlar. Tarihsel olarak, içerik yönetimi kavramı kuruluşlar tarafından üstlenilen web içeriğini yönetme çabalarından kaynaklanmıştır (Paivarinta & Munkvold, 2005). ECM kavramı, düz web içeriği yönetiminin ötesinde, "tüm ön yüz (frontend) uygulamaların ve cihazların arka yüz (backend) belge/dosya yönetim sistemleri ve veri tabanları ile ilintilendirilmesi" yöntemini ele alır. ECM yalnızca teknik sistemleri içermez, aynı zamanda "bir kuruluşun yaşam döngüleri boyunca bilgi varlıklarını yönetmek için ihtiyaç duyduğu stratejiler, araçlar, süreçler ve becerileri" de (Smith & McKeen, 2003) içerir.

Günümüz anlayışına göre, bir WCMS, HTML, XML, PDF gibi web formatlarında yapılandırılmış içerik oluşturmayı ve yayınlamayı destekler. Bir WCMS ayrıca içeriği gözden geçirmek ve onaylamak, içeriği arşivlemek ve bazı durumlarda içeriğin sürümünü oluşturmak için işlevler sunar (Hallikainen, Kivijarvi, & Nurmimaki, 2002). Bu işlevleri kullanarak, örneğin yazar, eleştirmen ve kullanıcı gibi farklı ayrıcalıklara sahip birkaç rolü içeren bir editörlük sürecini uygulamak mümkündür.

Bir WCMS kurumsal web siteleri, çevrimiçi mağazalar veya topluluk portalları oluşturmak için kullanılabilir. Bu sistemlerin oluşturulması için bir WCMS kullanmanın en büyük avantajı, içeriklerin değiştirilmesinin herhangi bir teknik bilgi gerektirmemesi, yani HTML kodunu düzenlemeden yapılabilmesidir. İçerik tipik olarak veri tabanlarında saklanır. İçeriğin yayınlanması, değiştirilmesi ve kaldırılması grafik kullanıcı ara yüzleri aracılığıyla yapılabilir. Bu roller açısından bakıldığında, bir WCMS iki ana bölüme ayrılabilir: bir ön yüz ve bir arka yüz kısmı. Ön yüz, tüketicilere mevcut içeriği sunar. Ön yüz kullanıcılarının içeriği değiştirme veya düzenleme izinleri yoktur. Bazı durumlarda (bir makaleye yorum göndermek gibi), ön yüz kullanıcılara özel izinler verilebilir. Yayınlama ve düzenleme, uygulamanın arka yüz kısmı aracılığıyla yapılır, yazarlar ve gözden geçirenler için çalışma yeri bu kısımdır. Genel olarak yazarlar, içeriklerini zengin bir metin düzenleyicisi kullanarak girerler. Düzenleme, CSS veya XSL gibi stil sayfaları uygulanarak oluşturulur. Bu stil sayfaları, örneğin bir WCMS'in kurumsal bir tasarıma uymasını sağlamak için arka yüz kullanıcılar tarafından uyarlanabilir.

WordPress WCMS

Ücretsiz ve açık kaynaklı olan WordPress, en fazla tercih edilen İYS'dir. Ana özellikleri şablon sistemi ve eklenti mimarisidir. Şablonlar, geliştiricilerin bir siteyi sıfırdan oluşturmak yerine önceden tanımlanmış bir yapı üzerine inşa etmelerine olanak tanır. Eklentiler ise araçları genişletmek için ek işlevsellik sağlar. WordPress'in kurucuları, en iyi yazılımların minimum kurulumla çalışması gerektiği felsefesiyle hareket etmişlerdir. Geliştiricilerin ve tasarımcıların zamanlarını site oluşturmaya ayırmaları için araçları kullanımı kolay ve sezgisel hale getirmeyi amaçlamışlardır. 2005 yılında logosunun oluşturulmasıyla WordPress'in markalaşması gerçekleşmiştir. WordPress 2008'de, Tema Dizini'ni başlatarak kullanıcıların temalar geliştirmesine ve yüklemesine izin vermiştir. Bu sayede bugün binlerce ücretsiz tema mevcuttur. WordPress Vakfı'nın oluşturulması 2010 yılında

tamamlanmıştır. WordPress'in ticari haklarını elinde tutan şirket Automattic, bazı kurucu katılımcıların amaçlarına göre temeli oluşturmuştur. Buradaki amaç, onları şirketten ayrı tutmak ve ticari marka koruyucusunun devralınması durumunda suistimal ve seyreltmeyi önlemektir. WordPress 2015 yılında başka önemli adımlar daha atarak, REST API altyapısıyla WordPress çekirdeğini birleştirmiştir. Bugün WordPress, kendi kendine barındırılan birçok web sitesinin arka planındaki güçtür. Mimarisi çoğunlukla b2/cafelog yapısının başlangıcına benzer. Web geliştirme için sunucu taraflı dili PHP'dir. MySQL ise açık kaynaklı ilişkisel bir veri tabanı yönetim sistemidir. Dolayısıyla WordPress oldukça esnek ve özelleştirilebilir bir yapıya sahiptir.

WordPress Bünyesinde Veri Korumaya Dair Sorunlar, Bulgular ve Değerlendirmeler

Pek çok WCMS hem ergonomik hem de verimli siteler oluşturmak için kullanıcılara kolaylıklar sunar. Ancak bu web altyapılarının güvensiz olduğu ve kötü niyetli bilgisayar korsanları için kolay hedefler oldukları düşüncesi de hâkimdir. Bu noktada cevaplanması gereken bazı sorular ortaya çıkmaktadır: "Gerçekten bu sistemler güvensiz midir, bu güvensizlikleri ortadan kaldırmak ne kadar mümkündür, bu sistemler bilgisayar korsanları için kolay hedefler midir, siber saldırı riskleri nelerdir, (özellikle Web İYS'nin) hangi unsurları gözlemlenmelidir?"

Bir diğer problem, İYS platformu kullanıcılarının tipolojisinin on-demand platform sitelere göre daha geniş olmasıdır. Bu da kullanıcıların güvenlik uygulamalarına daha az aşina olduğu ve İYS'ye karşı yapılan saldırıların başarı olasılığının arttığı anlamına gelir. Güvenlik açısından bakıldığında, başlıca İYS'ler açık kaynak olma avantajına sahiptir. Güvenlik araştırmacıları buna erişebilir ve onu test ederler, bu da güvenlik açıklarını belirlemeyi mümkün kılar. Ancak öte yandan kötü niyetli bilgisayar korsanlarının da bu sistemlere erişimi vardır ve bu da onların güvenlik açıklarını bulmalarını ve zafiyetleri istismar etmelerini sağlayabilir. Belli başlı İYS'lerin etrafındaki büyük topluluklar, bu sistemlere işlevsellik katan eklentiler (plugin) geliştirirler. Ancak bu durum eklentilerin entegrasyonu, yapılandırması ve bakımı ile ilgili riskleri de beraberinde ortaya çıkarmaktadır.

İYS ile ilişkili temel risklerden bir diğeri güncellemelerdir. Temel İYS'ler için çok sayıda eklenti, tema vb. mevcuttur. Dolayısıyla güncellemeler yalnızca İYS sürümleri için değil, aynı zamanda kullanılan çeşitli eklentilerin ve temaların sürümleri için de geçerlidir. Yeni güvenlik açıkları sürekli olarak bulunmakta ve hemen bir yama ile düzeltilmektedir. Bu nedenle güncellemeleri mümkün olan en kısa sürede yüklemek ve mevcut yamaları sık sık kontrol etmek gerekir. Bu konuda yaşanabilecek bir ihmal, kullanılan sistem üzerinde veri ihlali yaşama riskini ciddi oranda arttıracaktır.

İYS ile ilgili diğer büyük risk, belirli geliştirmelerle ilgilidir. Çoğu İYS tabanlı web sitesi, yalnızca yapılandırmaya değil aynı zamanda bir şirket içi yazılım geliştirme ekibi veya bir hizmet sağlayıcı tarafından özel olarak hazırlanmış geliştirmelere de aktarılır. Web İYS belirli geliştirmeler içeriyorsa, bunlar *sıfırdan (from scratch)* geliştirilen bir siteyle aynı riskleri sunar. Dolayısıyla bu anlamda sorulması gereken sorular şunlardır:

- Geliştiricilerin güvenlik konularındaki becerileri nelerdir?
- Kullanılan İYS'de sızma testleri yapılmış mıdır?
- Koddaki olası kusurlar nasıl düzeltilebilir?

Yöntem

Araştırmanın yöntemi olarak genel olarak nitel modelden yararlanılmış olup doküman analizi yöntemi kullanılmıştır. Doküman analizi yöntemi, yazılı veya basılı belgelerin içeriğini titizlikle ve sistematik olarak analiz

etmek için kullanılan bir nitel araştırma yöntemlerinden bir tanesidir (Wach, 2013). Bu yöntem, başta basılı ve elektronik dokümanlar olmak üzere tüm belgeleri incelemek ve değerlendirmek için kullanılan sistemli bir modeli ifade etmektedir. Nitel araştırmada kullanılan diğer yöntemler gibi doküman analizi de anlam çıkarmak, ilgili konu hakkında bir anlayış oluşturmak, deneysel bilgi geliştirmek için verilerin incelenmesini ve yorumlanmasını hedeflemektedir (Corbin & Strauss, 2008). Dokümanlar genel olarak araştırmacının herhangi bir müdahalesi olmadan kaydedilmiş makaleler, bildiriler, tezler, web sayfalarındaki forumlar ve konu ile ilgili web sayfasında yer alan yazıları ifade etmektedir. Bu kapsamda, öncelikle literatür taranmış, konu ile ilgili web sayfaları incelenmiş detaylı olarak analiz edilmiştir.

Ayrıca; çalışmada AWS tabanlı bir Wordpress Web İYS ortamı kurularak, bu sistemleri kullanan kurumları ne gibi tehlikelerin beklediği simüle edilmiş ve devamında önerilerde bulunulmuştur. İçerik ve verileri bekleyen tehlikeler ele alınmış, verileri korumak ve riskleri en aza indirgemek için güvenlik sürekliliğini sağlama anlamında neler yapılabileceği incelenmiştir. Çalışmanın amacı doğrultusunda, İYS ve Veri Koruma esas alınarak Wordpress güvenliği incelenmiştir. Alınabilecek önlemlerle ilgili olarak temel beş konu başlığı ve 21 alt başlık, literatür ve uzman görüşleri doğrultusunda belirlenmiştir. Bunlar;

- Wordpress İYS'nin yönetici bazında güvenliğinin incelenmesi
- Sunucu ve içerik bazında güvenliğin incelenmesi
- Wordpress'in yönetim bazında incelenmesi
- Protokol ve anahtar bazlı incelemeler
- Yapılandırma erişimleri bazında güvenlik incelemeleri

Bu temel beş konu başlığı altında 21 alt başlık ele alınmıştır.

- Çift doğrulama çerçevesinde yönetici paneli girişi
- Yönetici hesabı login bilgileri güvenliği
- Wp-admin yönetici panel URL'i
- Dosya ve dizin izinleri
- Barındırma güvenliği seviyesi
- Yedekleme çerçevesinde inceleme
- All in One WP Migration
- BackuUpWordpress
- Veri tabanı güvenliği çerçevesinde inceleme
- Servis kesintisi saldırılarına karşı güvenlik
- WordPress sürümü çerçevesinde inceleme
- Güncellemeler ve otomatikleştirme
- Eklentilerde güvenlik incelemesi
- Gerekli olmayan eklentiler çerçevesinde inceleme
- HTTP header çerçevesinde inceleme
- HTTPS kullanımı çerçevesinde inceleme
- Wp-config.php dosyasındaki güvenlik anahtarları bazında inceleme

- Bazı konfigürasyon dosyalarına erişim çerçevesinde incelenmesi
- Tema düzenleyici kısmının erişim çerçevesinde incelenmesi
- Hotlink çerçevesinde incelenmesi
- XML-RPC özelliği çerçevesinde incelenmesi

Bulgular

Wordpress İYS'nin Yönetici Bazında Güvenliğinin İncelenmesi

Çift doğrulama çerçevesinde yönetici paneli girişi

Parolalar, web'de oturum açmanın standardı olarak kullanılmaktadırlar. Ancak kırılmaları mümkündür. Güçlü parolalar oluşturulup bunlar düzenli olarak değiştirilse bile, oturum açılan her yerde (sunucu, tarayıcı vs.) saklanabilir ve bir sunucu / tarayıcı ihlali ile sızdırabilirler. Bir kişiyi ve dolayısıyla parolasını tanımlamanın üç yolu vardır: Kim olduğu, nelere sahip olduğu ve neler bildikleri. Parola ile oturum açmak, tek adımlı doğrulamadır. Yalnızca *'neler bildiği* 'ne dayanır. İki aşamalı kimlik doğrulama tanım gereği kimliğinizi kanıtlamak için yalnızca bir faktör yerine yukarıdaki olası üç faktörden ikisinin kullanıldığı bir sistemdir. Pratikte mevcut iki aşamalı uygulamalar yine bilinen bir şifreye dayanmaktadır ancak sahip olunan telefon veya başka bir cihazla da kimlik doğrulaması yapılabilir. Giriş sayfasına iki faktörlü kimlik doğrulama (2FA) modülünün tanıtılması önemli bir güvenlik önlemidir. Bu durumda, kullanıcı iki farklı bileşen için oturum açma ayrıntılarını sağlamak zorunda kalır. Web sitesi sahibi bu ikisinin ne olduğuna karar verir. Normal bir parola ve ardından gizli bir soru, gizli bir kod, bir dizi karakter veya daha popüler olan ve telefona bir kod gönderen Google Authenticator gibi uygulamalar olabilir. Bu şekilde, siteye yalnızca telefonun sahibi olan kişi giriş yapabilecektir. Bu sayede güvenlik bir üst seviyeye taşınmış olur (WordPress, 2021a).

Yönetici hesabı login bilgileri güvenliği

Brute Force saldırılar, bir siteye illegal erişim sağlamak için en basit yöntemlerden biridir. Kullanıcı adlarını ve parolalarını sisteme girene kadar tekrar tekrar denemek üzere tasarlanan bu saldırı yöntemi, *123456* gibi zayıf şifreler ve *admin* gibi bilinen kullanıcı adlarının kullanıldığı giriş bilgilerinde çok başarılı olabilir. Kısacası, bu saldırı herhangi bir web sitesinin güvenliğindeki en zayıf halkaya yani insana yapılan bir saldırıdır. Piyasadaki her web uygulamasına yapılabilir ancak popüler olan WordPress, daha sık saldırıya uğramaktadır.

Saldırıların çoğu, WordPress'in ilk sürümlerinin varsayılan olarak ayarlanması nedeniyle kullanıcıların *admin* kullanıcı adını kullandığı varsayarak yapılır. Dolayısıyla ilk yapılacak işlerden biri, bu kullanıcı adı yerine yeni bir hesap oluşturmak, tüm gönderileri o hesaba aktarmak ve *admin*i tamamen silmektir. Kullanıcı adını değiştirmek için Change Username eklentisi de kullanılabilir. Parolanın amacı, diğer insanların tahmin etmesini ve bir brute force saldırısının başarılı olmasını zorlaştırmaktır. Güvenli parolalar oluşturmak için kullanılabilecek birçok otomatize parola oluşturma aracı mevcuttur. WordPress ayrıca, şifreyi değiştirirken gösterilen bir şifre gücü ölçere sahiptir. Oluşturulan parolanın yeterli olduğundan emin olmak için bu araç kullanılabilir. Kullanıcıları güçlü parolalar oluşturmaya zorlamak için Force Strong Password eklentisini kullanılabilir. Bir parola seçerken kaçınılması gerekenler:

1. Gerçek ad, kullanıcı adı, şirket adı veya web sitesi adıyla ilgili herhangi bir değişiklik.

2. Herhangi bir dilde bir sözlükten bir kelime.
3. Kısa bir parola.
4. Sadece sayısal veya yalnızca alfabetik parola.

Wp-admin yönetici panel URL'i

Bazen belirsizliğe dayalı popüler WordPress güvenlik stratejisi site için etkili olabilir. Saldırı ihtimalini azaltmak için bilgisayar korsanlarının tahmin edebileceği arka kapıların bulunması zor hale getirilmelidir. Bu noktada WordPress giriş ve yönetici alanının gizlenmesi ve kilitlenmesi, güvenliği artıracaktır. Bunun için varsayılan wp-admin oturum açma URL'i değiştirilmeli ve oturum açma girişimleri sınırlandırılmalıdır. Varsayılan olarak, WordPress sitesi yönetici giriş URL'i herkesin bilebileceği şekliyle *domain.com/wp-admin* dir.

Yönetici, giriş URL'ini değiştirmek için ücretsiz WPS Hide Login veya Perfmatters eklentisini kullanabilir. Her iki eklentinin de basit bir giriş alanı vardır. URL'i seçerken bir botun veya scriptin tarayabileceği herhangi öngörülebilir bir dizinde yer alabilecek bir isim seçilmemelidir (Duò, 2021). Bu çözüm, illegal giriş denemelerinin azaltılmasına yardımcı olur. Ek olarak giriş sayısını sınırlandırmak da etkilidir. Cerber Limit Login Attempts oturum açma girişimlerini, kilitleme sürelerini ve IP kara- beyaz listelerini yönetmek için kullanılan, etkili ve ücretsiz bir eklentidir. Başka bir basit alternatif de yine ücretsiz Login Lockdown eklentisidir. Login LockDown, zaman damgası ve IP si ile birlikte başarısız olan her oturum açma girişimini kaydeder. Belirli bir IP aralığından kısa bir sürede belirlenen bir sayının üstünde deneme yapılırsa, o IP aralığından gelen tüm istekler devre dışı bırakılabilir. WPS Hide eklentisi ile de uyumludur.

Sunucu ve İçerik Bazında Güvenliğin İncelenmesi

Dosya ve dizin izinleri

WordPress, çeşitli dosyaların web sunucusu tarafından yazılabilir olmasına izin verir. Bu durum bazı işlevsel özellikler kazanmasını sağlar. Ancak, dosyalara yazma erişimi vermek özellikle paylaşımlı ortamlarda risk taşımaktadır. Yapılması gereken yazma erişimi verilmesi gereken yerlerde kısıtlamaları gevşetip diğer durumlarda olabildiğince sınırlandırmaktır. Dosya yüklemek benzeri eylemlerde ise daha az kısıtlamalı klasörler oluşturulmalıdır. Doğal olarak yönetici hesabının tüm dosyalara yazma izni olmalıdır. Aynı zamanda WordPress'ten yazma erişimi gereken dosyalarda, web sunucusu tarafından yazılabilir durumda olmalıdır. Söz konusu dosyalar eğer barındırma kurulumu gerektiriyorsa, web sunucusu tarafından kullanılan kullanıcı hesabına ait olmalıdır (Wright, 2019). Örnek bir olası izin şeması aşağıda belirtilmiştir:

/WordPress Kök Dizini: Eğer .htaccess dışındaki tüm dosyalar sadece kullanıcı hesabınız tarafından yazılabilir durumdaysa WordPress, sizin için yazma kurallarını otomatik bir şekilde yeniden oluşturabilir.

/wp-admin/ - WordPress Yönetim Alanı: Bu alandaki dosyalar sadece kullanıcı hesabı tarafından yazılabilir durumda olmalıdır.

/wp-includes/ - WordPress Uygulama Mantığının İlgili Kısmı: Bu alandaki dosyalar sadece kullanıcı hesabı tarafından yazılabilir durumda olmalıdır.

/wp-content/ - Kullanıcı Tarafından Sağlanan İçerik: Kullanıcı hesabına ek olarak ihtiyaç duyulan durumlarda web sunucusu işlemi tarafından da yazılabilir olmalıdır.

/Wp-content/ içinde şu dizinler bulunmaktadır:

/wp-content/themes/ - Tema dosyaları: Yerleşik tema düzenleyicisi kullanılmak istenirse, tüm dosyaların web sunucusu işlemi tarafından yazılabilir olmasını gerekir. Eğer istenmezse sadece kullanıcı hesabının yazma yetkili olması yeterlidir.

/wp-content/plugins/ - Eklenti Dosyaları*:* Bu alandaki dosyalar sadece kullanıcı hesabı tarafından yazılabilir durumda olmalıdır.

/wp-content/ ile mevcut olabilecek diğer dizinler, hangi tema ya da eklenti gerektiriyorsa ona göre belgelenmelidir ve izinler ona göre değiştirilebilir.

Sunucuya shell erişimi varsa aşağıdaki komutla dosya izinlerini özyinelemeli (recursive) olarak değiştirebilir:

Dizinler için: find /wordpress/kurulum/ dizinin/yolu/ -type d -exec chmod 755 {} \;

Dosyalar için: find /wordpress/kurulum/ dizinin/yolu/ -type f -exec chmod 644 {} \;

WordPress'e otomatik güncelleme yapması söylendiğinde tüm dosya işlemleri web sunucusunun kullanıcısı olarak değil dosyaların sahibi olan kullanıcı olarak gerçekleştirilir. Tüm dosyalar 0644 olarak ayarlanmıştır ve tüm dizinler 0755 olarak ayarlanmıştır ve yalnızca kullanıcı tarafından yazılabilir ve web sunucusu dâhil herkes tarafından okunabilir.

Barındırma güvenliği seviyesi

WordPress güvenliğinde site yönetim panelini erişimlere kısıtlamaktan çok daha fazlası yapılabilir. WordPress'i kendi VPS'inde (Virtual Private Server) barındıranların gerekli önlemleri almak için yeterli teknik bilgiye sahip olması gerekir. Yeterli teknik bilgiye sahip olmadan tasarruf etme amaçlı sistem yöneticisi olmaya çalışmak kötü bir fikirdir. Sunucu sıkılaştırma (server hardening), bir WordPress ortamının güvenliğini sürdürmenin anahtarıdır. WordPress sitelerini barındıran BT altyapısının hem fiziksel hem de sanal karmaşık tehditlere karşı savunma yapabilmesini sağlamak için birden fazla donanım ve yazılım düzeyinde güvenlik önlemi katmanı gerekir. Bu nedenle WordPress barındıran sunucular en güncel işletim sistemi ve güvenlik yazılımlarıyla güncellenmeli, ayrıca güvenlik açıkları ve kötü amaçlı yazılımlara karşı kapsamlı bir şekilde test edilmeli ve taranmalıdır. Sunucu düzeyinde güvenlik duvarları ve izinsiz giriş tespit sistemleri (IDS) WordPress'i sunucuya yüklemeden önce, WordPress kurulumu ve web sitesi oluşturma aşamalarında bile iyi korunmasını sağlamak için hazır ve aktif olmalıdır. Ek olarak WordPress içeriğini korumak amacıyla makineye yüklenen her yazılım, optimum performansı sürdürmek için güncel veri tabanı yönetim sistemleriyle de uyumlu olmalıdır. Hassas içeriği kötü niyetli davetsiz misafirlerden gizlemek için sunucular güvenli ağ iletişimi ve dosya aktarım protokolleri (FTP yerine SFTP gibi) kullanacak şekilde yapılandırılmalıdır (WPBeginner, 2021a).

Yedekleme çerçevesinde inceleme

Güvenilir yedeklemeler için veri bütünlüğü çok önemlidir. Yedeklemeyi şifrelemek, her yedekleme dosyası için bağımsız bir MD5 hash kaydı tutmak ve/veya salt okunur medyaya yedekleme yerleştirmek, verilerin değiştirilmediğine dair güveni artırır. Sağlam bir yedekleme stratejisi için WordPress kurulumunun tamamına (çekirdek dosya ve veri tabanı dâhil) ait anlık görüntüler, düzenli olarak güvenilir bir konumda tutulmalıdır (WordPress, 2021b). Haftalık anlık görüntüler oluşturan bir sitenin güvenliği, örneğin 1 Şubat'ta ihlal edilirse,

ancak güvenlik ihlali 12 Şubat'a kadar tespit edilmezse; site sahibinin, siteyi yeniden oluşturmasına yardımcı olabilecek yedeklemeler tehlikeye atılmış olur.

All in One WP Migration

WordPress admin paneli eklentiler kısmından aranarak yüklenebilir. All In One WP Migration manuel olarak site yedeğinin kısa bir sürede alınmasını ve yine hızlı bir şekilde geri yüklenebilmesini sağlayan kullanışlı, iyi derecelendirilmiş ve yüksek kullanıma sahip bir eklentidir.

BackUpWordpress

BackUpWordPress, web sitesini otomatik olarak yedeklemek için ücretsiz ve güçlü bir eklentidir. BackUpWordPress, veri tabanı ve tüm dosyalar dâhil olmak üzere sitenin tamamını, belirlenen zamanlamaya göre yedekleyecektir. Geri yükleme işlemini çalıştırmadan önce veri tabanının ve dosyaların local yedeği oluşturulmak istenirse, geri yükleme çalıştırıldığında önceki sürümün geri alınamayacağı göz önünde bulundurulmalıdır.

Veri tabanı güvenliği çerçevesinde inceleme

WordPress veri tabanında güvenliği iyileştirmek için yapılması gereken birkaç farklı işlem vardır. Birincisi, akıllı bir veri tabanı adı kullanmaktır. Örneğin site wordpresssite.com olarak adlandırılmışsa, varsayılan olarak WordPress veri tabanı büyük olasılıkla wp_wordpresssite olarak adlandırılır. Veri tabanı adını daha belirsiz bir adla değiştirmek, bilgisayar korsanlarının veri tabanı ayrıntılarını tanımlamasını ve bunlara erişmesini zorlaştıracaktır. İkincisi, farklı bir veri tabanı tablosu öneki kullanmaktır. Varsayılan olarak WordPress wp_ önekini kullanır (CodeInWP, 2021). Bunu örneğin 34xw_ gibi bir önekle değiştirmek çok daha güvenli olabilir. WordPress kurulumda bir tablo öneki sorar. Mevcut kurulumlarda WordPress tablo önekini değiştirmenin yolları da vardır. Blog gönderileri gönderme, medya dosyalarını yükleme, yorum gönderme, yeni WordPress kullanıcıları oluşturma ve WordPress eklentileri yükleme gibi normal WordPress işlemleri için MySQL veri tabanı kullanıcısının yalnızca MySQL veri tabanına veri okuma ve yazma (SELECT, INSERT, UPDATE, DELETE...) ayrıcalıklarına ihtiyacı vardır. Bu nedenle, DROP, ALTER ve GRANT gibi diğer veri tabanı yapısı ve yönetim ayrıcalıkları iptal edilebilir. Bu tür ayrıcalıkları iptal etmek, sıkılaştırma politikalarını da iyileştirmiş olur.

Servis kesintisi saldırılarına karşı güvenlik

Belki de diğer zafiyetlerin hepsinden en tehlikelisi olan DoS güvenlik açığı, web site sunucusunun işletim sistemlerinin belleğini aşmak için koddaki açıklıkları ve hataları kullanır. Bilgisayar korsanları, DoS saldırıları ile WordPress yazılımının eski ve hatalı sürümlerini kullanarak milyonlarca web sitesini tehlikeye atmakta ve bu sayede milyonlarca dolar para elde etmektedirler. Mali olarak motive olmuş siber suçluların küçük şirketleri hedef alma olasılıkları daha düşük olsa da, büyük işletmelere saldırmak için botnet ağları oluştururken eski savunmasız web sitelerini hedef alma eğilimindedirler (Hughes, 2021).

WordPress yazılımının en son sürümleri bile yüksek profilli DoS saldırılarına karşı kapsamlı bir şekilde savunma yapamaz ancak güncel bir WordPress sürümünü kullanmak, en azından büyük çaplı kurumlar ve gelişmiş siber suçlular arasında gerçekleşen saldırılarda kullanılmamak için yardımcı olacaktır. 21 Ekim 2016 tarihinde gerçekleşen ve DNS DDoS saldırısı nedeniyle İnternet'in dünya çapında kesildiği gün göstermiştir ki site güvenliğini artırmak için üst seviye bir DNS sağlayıcısı kullanmak da çok önemlidir.

Wordpress'in Yönetim Bazında İncelenmesi

WordPress sürümü çerçevesinde inceleme

WordPress sürümünü gizlemek, WordPress güvenliği noktasında alınması gereken bir diğer önemli önlemdir. Güncel olmayan bir WordPress kurulumunun çalıştırılması ve bunun dışarıdan görülmesi risk oluşturabilir. Varsayılan olarak WordPress sürümü sitenin kaynak kodunun başlığında görünür. Aşağıdaki kodun WordPress temasının functions.php dosyasına eklenmesi sürümü bilgisinin kaldırılması için yeterlidir (Abela, 2020).

```
function remove_wp_version_rss() {
 return'';}      add_filter('the_generator','remove_wp_version_rss');
```

WordPress sürümünü gizlemeye olanak tanıyan Perfmatters gibi eklentiler de kullanılabilir. Sürüm bilgisi her WordPress sürümünde bulunan varsayılan readme.html dosyasında da gözükür. Dosya kurulumun wordpressdomainim.com/ readme.html kök dizininde bulunur. Bu dosya FTP yoluyla silinebilir. WordPress 5.0 veya üstünde, sürüm numarası artık bu dosyada yer almamaktadır.

Güncellemeler ve otomatikleştirme

WordPress güvenliğini sağlamlaştırmanın bir başka önemli yolu da her zaman güncel tutmaktır. Buna WordPress çekirdeği dışında eklentiler ve temalar da dâhildir. Bunlar için çoğu zaman güvenlik geliştirmeleri ve hata düzeltmeleri nedeniyle güncelleme yayınlanır. WordPress yazılımının ve eklentilerinin eski sürümlerini çalıştıran milyonlarca işletme hala mevcuttur. Güncelleme yapmama sebebi ise genelde "Güncelleme sonrası site bozulabilir. Temel değişiklikler ortadan kalkabilir. X eklentisi çalışmayabilir. Güncelleme ile gelen yeni işleve ihtiyaç duyulmaz" şeklindedir. Aslında, web siteleri çoğunlukla eski WordPress sürümlerindeki hatalar nedeniyle bozulur. Temel değişiklikler, WordPress ekibi ve ilgili riskleri anlayan uzman geliştiriciler tarafından asla tavsiye edilmez. WordPress güncellemeleri çoğunlukla, en son eklentileri çalıştırmak için gereken ek işlevsellikle birlikte sahip olunması gereken güvenlik yamalarını içerir. Eklenti güvenlik açıkları bilgisayar korsanları için bilinen giriş noktalarının % 55,9'unu temsil etmektedir. WordFence şirketi, saldırıların kurbanı olan 1000'den fazla WordPress site sahibiyle yaptıkları bir çalışmada bu bulguya ulaşmıştır. Dolayısıyla bu kurbanlardan biri olmamak için eklentileri de güncellemek önemlidir. Ayrıca, yalnızca güvenilir eklentileri yüklemek gerekmektedir. İhmal edilmiş WordPress eklentileri ve temaları kesinlikle kullanılmamalıdır. Çünkü geliştirilmesi ihmal edilmiş kodun hangi açıklıkları içerebileceği asla bilinemez (Holcombe, 2021). Bu da sitenin saldırıya uğrama riskini arttırır. Herhangi bir kötü amaçlı yazılım türü tespit etmek adına bir eklenti veya temanın dosyalarını taramak için VirusTotal gibi çevrimiçi bir tarama aracı kullanılabilir. Çekirdek, WordPress panosundaki "Güncellemeler" kısmından güncellenebilir veya en son sürüm indirilerek SFTP aracılığıyla yüklenebilir. WordPress eklentilerini güncellemek de sürüm güncellemeye çok benzer bir işlemdir ve panelden yapılabilir. Aynı şekilde bir eklenti manuel olarak da güncellenebilir. Eklenti geliştiricisinden veya WordPress'ten en son sürüm eklenti indirilip /wp-content/plugins dizinine SFTP yoluyla yüklenebilir. Eklentilerin "Son Güncelleme" tarihine ve kaç derecelendirmeye sahip olduğuna mutlaka bakılmalıdır. WordPress ayrıca bir süredir güncellenmemiş çoğu eklentinin üstünde bir uyarı içerir. Eklenti indirilirken buna da dikkat etmek gerekir.

Eklentilerde güvenlik incelemesi

WordPress sitesinin daha iyi korunmasına yardımcı olacak birçok işlevsel eklenti mevcuttur (Jackson, 2021). Bunlardan birkaçı aşağıda verilmiştir:

Sucuri Security
iThemes Security
WordFence Security
WP fail2ban
SecuPress

Yukarıdaki bu eklentilerin bazı tipik özellikleri ve kullanımları şunlardır:

1. Kullanıcı profilleri oluşturulurken güçlü parolalar oluşturmaya zorlama
2. Parolaların süresinin dolmasını ve düzenli olarak sıfırlanmasına zorlama
3. Kullanıcı işlemlerinin günlük log kayıtlarının tutulması
4. WordPress güvenlik anahtarlarını kolayca güncellenmeye imkan sunma
5. Kötü Amaçlı Yazılım Taraması
6. İki faktörlü kimlik doğrulama
7. reCAPTCHA kullanımı
8. WordPress güvenlik duvarları
9. IP beyaz listesi ve kara listesi
10. Dosya değişiklik log kayıtları
11. DNS değişikliklerini izleme
12. Kötü amaçlı ağları engelleme
13. Ziyaretçilerle ilgili WHOIS bilgilerini görüntüleme

Birçok güvenlik eklentisinin içerdiği çok önemli bir diğer özellik de checksum aracıdır. Bunun anlamı WordPress kurulumunun incelenip çekirdek dosyalarda değişiklik yapılıp yapılmadığının API aracılığıyla doğrulanmasıdır. Bu dosyalardaki herhangi bir değişiklik bir saldırıya işaret edebilir. Checksum aracını çalıştırmak için WP-CLI da kullanılabilir. Bir başka işlevsel eklenti WP Security Audit Log eklentisidir. WordPress multisite veya multi-author siteler üzerinde çalışanlar için çok kullanışlıdır. Kullanıcı üretkenliğinin sağlanmasına yardımcı olur ve yöneticilerin girişler, şifre değişiklikleri, tema değişiklikleri, widget değişiklikleri, yeni yayın oluşturma, WordPress güncellemeleri vb. gibi değişen her şeyi görmesini sağlar. Ayrıca e-posta bildirimleri, kullanıcı oturumları yönetimi, arama ve raporlar gibi ek fonksiyonlara da sahiptir.

Gerekli olmayan eklentiler çerçevesinde inceleme

Fazla eklentiler işi olması gerekenden daha zor ve karmaşık hale getirebilir. Çok sayıda gerekli eklentiye sahip olmanın yanlış bir tarafı olmasa da fazla kaynak kullanan çok sayıda eklentiye sahip olmak bir sorun haline gelebilir. Fazladan HTTP istekleri eklemek web sitesinin yüklenme hızını yavaşlatabilir. Ek olarak çok fazla eklentiye sahip olmak dikkati dağıtır ve esas odaklanılması gereken konudan yani sistem iyileştirmesinden uzaklaştırır.

Protokol ve Anahtar Bazlı İncelemeler

HTTP header çerçevesinde inceleme

WordPress'te güvenliği arttırmak adına yapılabilecek bir diğer işlem, HTTP güvenlik başlıklarını (header) kullanmaktır. Tarayıcıya sitenizin içeriğini işlerken nasıl davranması gerektiğini bildirir ve ekseriyetle web sunucusu düzeyinde yapılandırılır. Birçok değişik HTTP güvenlik başlığı bulunur. En önemlileri aşağıda belirtilmiştir (Ray, 2018):

Content Security Policy
X-XSS Protection
Strict Transport Security
X-Frame Options
Public Key Pins
X-Content Type

Chrome devtools aracı yardımıyla, site ilk yanıtındaki başlığa bakarak, WordPress sitesinde hangi başlıkların çalıştığı kontrol edilebilir. Mevcutta kullanılmakta olan güvenlik header'ları ise Scott Helme'nin ücretsiz olan securityheaders.io aracıyla bulunabilir. HTTP güvenlik header'larını uygularken bunun WordPress subdomainlerini (alt domain) nasıl etkileyebileceğini unutmamak önemlidir. Örneğin Content Security Policy başlığı eklenir ve erişimi domainlere göre kısıtlanırsa, alt domainlerin de eklenmesi gerekir.

HTTPS kullanımı çerçevesinde inceleme

WordPress güvenliğini sağlamanın en önemli yollarından biri, bir SSL sertifikası yüklemek ve siteyi HTTPS üzerinden çalıştırmaktır. HTTPS protokolü, web uygulamasının ya da tarayıcının web sitesine güvenli bir şekilde bağlanması için oluşturulan bir mekanizmadır (WPBeginner, 2019b).

SSL sadece kredi kartları ile işlem yapıldığında gereklidir algısı kesinlikle doğru değildir. Sitenin hangi işlevleri yerine getirdiğine bakılmaksızın SSL sertifikası kesinlikle kullanılmalıdır. Birçok hosting servis sağlayıcısı da Let's Encrypt gibi araçlarla ücretsiz SSL sertifikaları sunarlar. HTTPS kullanmanın neden önemli olduğuna dair birkaç nedeni aşağıda verilmiştir:

1. Güvenlik

Şifreli bilgi akışı ve güvenli giriş garantisi sağladığından dolayı SSL sertifikası Web siteleri için önemlidir. SSL'in sağlamış olduğu HTTPS protokolü siteye erişim yapılırken girilen login bilgilerini iki taraf (tarayıcı-sunucu) arasında şifreli olarak gönderir. Bu şekilde bilgiler, hackerların man in the middle (MITM) türü gibi saldırılarına karşı korunmuş olur.

2. SEO (Search Engine Optimization)

Google, HTTPS protokolünün kullanımını sıralama faktörü olarak değerlendirmektedir. Dolayısıyla HTTPS kullanımı önemli olup sıralamada yukarılara çıkmak için kesinlikle katkı sağlar.

3. Güvenilirlik

HTTPS protokolü ile siteye girildiğinde tarayıcıların sol tarafında yeşil bir simge belirmesi, mevcut sitedeki bilgi alışverişinin SSL sertifikasıyla şifrelenerek güvenli halde gerçekleştirildiğini ifade eder. Günümüzde kullanıcının güven duyması adına küçük farkların bile büyük etki yaratabildiği göz önünde bulundurulduğunda, SSL sertifikasının kullanımı da gün geçtikçe önem kazanmaktadır.

4. Referans Veri Kaybı

Google tarafından HTTPS'den HTTP protokolüne, link referansları ve veri alışverişleri net bir şekilde aktarım yapılmamaktadır. Eğer referans üzerine kurulu bir sistem planlanıyorsa HTTPS protokolü mutlaka kullanılmalıdır.

5. Tarayıcılarda "Bu Site Güvenli Değildir" Uyarıları

Hemen hemen tüm tarayıcılarda HTTPS protokolü kullanılmayan siteler not secure (güvenli olmayan) şeklinde işaretlenmektedir. Bu durum kullanıcının siteye girmesini engellemektedir. Özellikle Google Chrome 68 üstü sürümlerde aktifleşen bu uyarı, HTTP protokolünü kullanarak siteye bağlantı kurmak isteyen kullanıcıların girişini sekteye uğratmaktadır.

6. Performans

HTTPS protokolünü kullanan siteler HTTP/2 teknolojisinin avantajlarından faydalanarak çok daha hızlı açılış hızına ulaşabilirler.

Wp-config.php dosyasındaki güvenlik anahtarları bazında inceleme

WordPress güvenlik anahtarları (security key), kullanıcının çerezlerinde depolanan bilgilerin şifrelenmesini iyileştiren bir dizi rastgele değişkendir. WordPress 2.7'den bu yana kullanılan 4 farklı anahtar vardır: *LOGGED_IN_KEY, SECURE_AUTH_KEY, NONCE_KEY ve AUTH_KEY* (Belani, 2019). WordPress kurulduğunda bunlar rastgele oluşturulur. Dolayısıyla rastgele oluşturulan bu anahtarları değiştirmek önemlidir. Bununla birlikte birden fazla sürüm geçişi yapıldıysa veya başka birinden site satın alındıysa, yeni anahtarlar oluşturmak önerilmektedir. WordPress, rastgele anahtar oluşturmak için kullanabilecek ücretsiz bir araca da sahiptir. Bu araçla oluşturulacak yeni anahtar ile Wp-config.php dosyası güncellenebilir.

Yapılandırma Erişimleri Bazında Güvenlik İncelemeleri

Bazı konfigürasyon dosyalarına erişim çerçevesinde incelenmesi

WordPress sitelerinin çoğunun kök (root) dizininde bir .htaccess dosyası bulunur. Güçlü bir yapılandırma dosyası olan bu dosya, dizin taramasını devre dışı bırakmak, yönetici alanını parolayla korumak, SEO dostu bir URL yapısı oluşturmak ve benzeri işlemler için kullanılır. Bu dosya varsayılanda WordPress web sitesinin root dizinindedir. Ancak istenirse içteki WordPress dizinlerinde de oluşturulabilir ve kullanılabilir.

Bir .htaccess dosyası oluşturulması, bu dosyanın /wp-include/ ve /wp-content/uploads/ dizinlerine yüklenmesi, web sitesini backdoor (arka kapı) erişim dosyalarından korur. Örneğin Notepad benzeri bir program yardımıyla oluşturulan .htaccess adında boş bir dosyanın içine konfigürasyon dizinlerine dışarıdan erişimle ilgili aşağıdaki kod girilir ve ilgili dosya WordPress sunucusunun /wp-include/ ve /wp-content/uploads/ klasörlerine yüklenirse, bu dizinlerde kontrolsüzce bir PHP dosyasının çalışması engellenecektir (WPBeginner, 2017c).

```
<Files *.php>
```

```
deny from all
</Files>
```

Tema düzenleyici kısmının erişim çerçevesinde incelenmesi

Birçok WordPress sitesinin birden fazla kullanıcısı ve yöneticisi vardır ve bu da WordPress güvenliğini daha karmaşık hale getirmektedir. Yazarlara veya katkıda bulunanlara yönetici erişimi sağlamak doğru olmayan bir uygulamadır. Ancak birçok sitede yapılan bir uygulamadır. Kullanıcılara hiçbir şeyi bozmamaları için doğru rolleri ve izinleri vermek önemlidir. Bundan dolayı, WordPress'teki Görünüm ve Tema düzenleyicisini devre dışı bırakmak faydalı olabilir.

Görünüm Düzenleyicisinde bir şeyleri düzenlemek yerine dosyayı yerel olarak düzenlemek ve SFTP yoluyla yüklemek çok daha güvenilirdir. En doğru uygulama şekli ise bu gibi işlemlerin önce bir geliştirme ortamında test edilmesidir. WordPress sitesi saldırıya uğrarsa saldırganların ilk yapacağı muhtemel işlem, Görünüm Düzenleyicisi aracılığıyla bir PHP dosyasını veya temasını düzenlemektir. Bu şekilde sitede hızlı bir şekilde kötü amaçlı kod çalıştırılabilirler. Saldırganların, kontrol panelinde buna erişimlerinin olmaması, saldırıları bir nebze önlemeye yardımcı olabilir. Tüm kullanıcıların *edit_themes*, *edit_plugins* ve *edit_files* yetkilerini kaldırmak için aşağıdaki kod wp-config.php dosyasına yerleştirebilir (Bogdanovic, 2020):

```
define('DISALLOW_FILE_EDIT', true);
```

Hotlink çerçevesinde incelenmesi

Hotlinking, internetteki herhangi bir yerden kullanılan görsel URL'inin doğrudan sitede kullanılmasıdır. Bu durumda görsel orijinal konumundan gösterilir. Hotlinked sitenin bant genişliğinin kullanılması çok büyük bir sorun olarak görülmese de çok fazla bant genişliği ciddi bir maliyet oluşturabilir (Aslam, 2019).

Apache Sunucuda Hotlink Önleme: Apache sunucuda .htaccess dosyasına alttaki kod eklenerek engellenebilir.

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond%{HTTP_REFERER} !^http(s)?://(www\.)?domaininiz.com [NC]
RewriteRule\.(jpg|jpeg|png|gif)$ http://dropbox.com/hotlink-placeholder.jpg [NC,R,L]
```

Koddaki ikinci satır, doğrudan resme bağlanmasına izin verilen siteyi tanımlar. Eğer birden çok siteye izin verilmek istenirse ilgili satır çoğaltılarak yönlendiren değiştirilebilir.

NGINX Sunucuda Hotlink Önleme: *Hotlinking işleminin* NGINX sunucuda önlenmesi aşağıdaki kodun yapılandırma dosyasına eklenmesi ile mümkündür.

```
location ~ .(gif|png|jpe?g)$ {
valid_referers none blocked ~.google. ~.bing. ~.yahoo domaininiz.com *.domaininiz.com;
if ($invalid_referer) {
return 403;
}}
```

Resimler bir CDN'den sunuluyorsa, CDN'ler için kurulumlar farklı olacaktır.

XML-RPC özelliği çerçevesinde incelenmesi

Son yıllarda XML-RPC, brute force saldırıları için giderek daha büyük bir hedef haline gelmiştir. Bu protokolün arka plandaki önemli özelliklerinden biri de tek bir istekte birden çok fonksiyonu çalıştırmak için system.multicall fonksiyonunun kullanabilmesidir. Bu durum uygulamanın bir HTTP isteği içinde birden fazla komut geçirebilmesini sağlayan kullanışlı bir işlevdir. Ama kötü niyetli kullanımı da mümkündür. Bu durumda doğru olan, ihtiyaç olmayan yerde devre dışı bırakılmasıdır. Örneğin XML-RPC kullanan Jetpack eklentisinde, kullanıcıların çoğu bu fonksiyona ihtiyaç duymaz. XML-RPC'in web sitesinde çalışıp çalışmadığı XML-RPC Validator aracılığıyla öğrenilebilir (Bartley. 2020). WordPress sitesi bunun üzerinden çalıştırıldığında XML-RPC etkin değilse bir hata mesajı alınacaktır. Özelliği tümden devre dışı bırakmak istenirse ücretsiz olan Disable XML-RPC veya Premium Perfmatters eklentisi kullanılabilir.

Tartışma ve Sonuç

Web'de oturum açmanın standardı olarak parolalar kullanılmaktadırlar. Giriş sayfasına iki faktörlü kimlik doğrulama (2FA) modülünün tanıtılması önemli bir güvenlik önlemidir. Bu durumda, kullanıcı iki farklı bileşen için oturum açma ayrıntılarını sağlamak zorunda kalır.

Brute Force saldırılar, bir web sitesinin güvenliğindeki en zayıf halkaya yani insana yapılan bir saldırıdır. Piyasadaki her web uygulamasına yapılabilir ancak popüler olan WordPress, daha sık saldırıya uğramaktadır. Saldırıların çoğu, WordPress'in ilk sürümlerinin varsayılan olarak ayarlanması nedeniyle kullanıcıların admin kullanıcı adını kullandığını varsayılarak yapılır. Dolayısıyla ilk yapılacak işlerden biri, bu kullanıcı adının yerine yeni bir hesap oluşturmak, tüm gönderileri o hesaba aktarmak ve admini tamamen silmektir.

Saldırı ihtimalini azaltmak için bilgisayar korsanlarının tahmin edebileceği arka kapıların bulunması zor hale getirilmelidir. Bu noktada WordPress giriş ve yönetici alanının gizlenmesi ve kilitlenmesi, güvenliği artıracaktır. Bunun için varsayılan wp-admin oturum açma URL'i değiştirilmeli ve oturum açma girişimleri sınırlandırılmalıdır.

WordPress, çeşitli dosyaların web sunucusu tarafından yazılabilir olmasına izin verir. Bu durum bazı işlevsel özellikler kazanmasını sağlar. Ancak, dosyalara yazma erişimi vermek özellikle paylaşımlı ortamlarda risk taşımaktadır. Yapılması gereken yazma erişimi verilmesi gereken yerlerde kısıtlamaları gevşetip diğer durumlarda olabildiğince sınırlandırmaktır. Bir WordPress ortamının güvenliğini sürdürmenin anahtarıdır. Bu nedenle WordPress barındıran sunucular en güncel işletim sistemi ve güvenlik yazılımlarıyla güncellenmeli, ayrıca güvenlik açıkları ve kötü amaçlı yazılımlara karşı kapsamlı bir şekilde test edilmeli ve taranmalıdır.

Güvenilir yedeklemeler için veri bütünlüğü çok önemlidir. Yedeklemeyi şifrelemek, her yedekleme dosyası için bağımsız bir MD5 hash kaydı tutmak ve/veya salt okunur medyaya yedekleme yerleştirmek, verilerin değiştirilmediğine dair güveni artırır. Varsayılan olarak WordPress veri tabanı büyük olasılıkla wp_wordpresssite olarak adlandırılır. Veri tabanı adını daha belirsiz bir adla değiştirmek, bilgisayar korsanlarının veri tabanı ayrıntılarını tanımlamasını ve bunlara erişmesini zorlaştıracaktır.

WordPress yazılımının en son sürümleri bile yüksek profilli DoS saldırılarına karşı kapsamlı bir şekilde savunma yapamaz. Ancak güncel bir WordPress sürümünü kullanmak, en azından büyük çaplı kurumlar ve gelişmiş siber suçlular arasında gerçekleşen saldırılarda kullanılmamak için yardımcı olacaktır. WordPress sürümünü gizlemek, WordPress güvenliği noktasında alınması gereken bir diğer önemli önlemdir. WordPress sürümünü gizlemeye

olanak tanıyan Perfmatters gibi eklentiler kullanılabilir. WordPress'ten en son sürüm eklenti indirilip /wp-content/plugins dizinine SFTP yoluyla yüklenebilir. Eklentilerin "Son Güncelleme" tarihine ve kaç derecelendirmeye sahip olduğuna mutlaka bakılmalıdır. WordPress ayrıca bir süredir güncellenmemiş çoğu eklentinin üstünde bir uyarı içerir. Eklenti indirilirken buna da dikkat etmek gerekir.

Birçok güvenlik eklentisinin içerdiği çok önemli bir diğer özellik de checksum aracıdır. Bunun anlamı WordPress kurulumunun incelenip çekirdek dosyalarda değişiklik yapılıp yapılmadığının API aracılığıyla doğrulanmasıdır. Çok sayıda gerekli eklentiye sahip olmanın yanlış bir tarafı olmasa da fazla kaynak kullanan çok sayıda eklentiye sahip olmak bir sorun haline gelebilir. Sitenin hangi işlevleri yerine getirdiğine bakılmaksızın SSL sertifikası kesinlikle kullanılmalıdır. Birçok hosting servis sağlayıcısı da Let's Encrypt gibi araçlarla ücretsiz SSL sertifikaları sunarlar.

WordPress'de bir .htaccess dosyası oluşturulması, bu dosyanın /wp-include/ ve /wp-content/uploads/ dizinlerine yüklenmesi, web sitesini backdoor (arka kapı) erişim dosyalarından korur. WordPress'teki görünüm ve tema düzenleyicisini devre dışı bırakmak faydalı olabilir. Saldırganlar görünüm düzenleyicisi aracılığıyla bir PHP dosyasını veya temasını düzenleyebilir. Bu şekilde sitede hızlı bir şekilde kötü amaçlı kod çalıştırılabilirler. Saldırganların, kontrol panelinde buna erişimlerinin olmaması, saldırıları bir nebze önlemeye yardımcı olabilir.

Bu çalışma kapsamında yapılan uygulamalar göstermektedir ki; sanılanın aksine WordPress İYS siteleri, talep üzerine geliştirilen (on-demand) bir site kadar güvenlidir. Dünya çapında iyi bilinen ve yaygın bir İYS kullanılıyorsa, düzenli olarak gelişen sağlam bir çözüm olarak kabul edilebilir. Fakat düzenli bakım yapılmadığında, risk analizlerine göre bir güvenlik çerçevesi çizilmediğinde, teknik anlamda yeterli olmayan bir şirket veya serbest çalışan (freelance) kişiler tarafından geliştirildiğinde; WordPress sitesi için aynı sağlamlık ve güvenilirlik geçerli olmayabilir.

Bununla birlikte İYS sistemleri daha fazla saldırıya uğrama riski taşırlar. Bu sistemler, bağlantılı güvenlik açıklarından yararlanmak isteyen bilgisayar korsanlarının toplu saldırı hedefleridirler. WordPress, Joomla, Drupal, Shopify gibi başlıca İYS platformları, tüm web sitelerinin %46,8'ini oluşturmaktadır. Bu nedenle saldırganların savunmasız sitelere rastlama olasılığı da yüksektir. Wordpress sisteminin ve eklentilerin güvenliği, bilinen İYS sistemleri için genellikle çok aktif olan topluluğa bağlıdır. Ancak, sistemin eklenti bakımı konusunda dikkatli olmak gerekir. Çözümün daha az popüler olma, daha az korunma ve hatta bir noktada terk edilme riski her zaman vardır. Eklentileri aşağıda verilen iki ana ilke göz önünde bulundurularak seçilmelidir:

1. Bakımı yapılmayan bir eklentide yeni güvenlik açıkları keşfedilirse, risklere maruz kalmamak adına, güncel tutulan eklentiler tercih edilmelidir.
2. Özel ihtiyaçlar için tasarlanmış "in-house" bir eklenti yerine, tanınan ve yaygın olarak kullanılan eklentiler tercih edilmelidir.

Kısaca, WordPress için kullanılan her eklentinin gelişimi yakından izlenmelidir. Herhangi bir tescilli/bilinen üründe olduğu gibi güvenlik, şirket tarafından kendisine verilen öneme ve geliştirme ekibinin bu konudaki bilgisine bağlıdır. WordPress gibi bilindik İYS sistemlerinin avantajı, geliştirilmesinden ve güvenliğinden doğrudan bir ekibin sorumlu olmasıdır. Bu nedenle, planlanan güncellemelerin, test edilecek özelliklerin, düzeltilecek öğelerin ve benzeri işlemlerin bir yol haritası vardır. Ancak ürünün bir noktadan sonra geliştirici tarafından desteklenmemesi ve artık herhangi bir bakımın yapılmaması da mümkündür.

İster açık kaynaklı ister ticari lisanslı bir İYS olsun, seçimler ihtiyaçların durumuna göre yapılmalıdır. Sitenin güvenliği, açık kaynaklı bir İYS'ye dayanıp dayanmamasından ziyade, her şeyden önce bu sistemin nasıl yönetildiğine, nasıl yapılandırıldığına ve bunların nasıl sürdürüldüğüne bağlıdır.

Web sitelerinin karşı karşıya olduğu veri hırsızlığı (özellikle müşteri hesaplarının oluşturulmasına izin veren siteler için), hizmet kesintileri veya yasadışı içerik barındırma gibi riskler İYS'nin boyutuna ve işlevlerine göre değişir. İlk etap için en iyi çözüm, ilk başta yalnızca büyük risklere odaklanabilen bir web testi gerçekleştirerek, güvenlik açıklarını düzeltmek için tespit çalışması yapmak ve sonra adım adım detaylandırmaktır. Sonuç olarak, WordPress ile geliştirilen siteler de güvenlidir ancak güvenliklerini sağlamak için dikkate alınması gereken belirli şartlara sahiptir.

Yayın Etiği Bildirimi / Research Ethics

Yazarlar araştırmanın etik dışı bir sorunu olmadığını, araştırma ve yayın etiği konularını gözlemlediklerini beyan etmektedir. / The authors declare that the research does not have an unethical problem and that they observe research and publication ethics.

Araştırmacıların Katkı Oranı / Contribution Rate of Researchers

Birinci araştırmacı literatür taraması, yöntem, tartışma ve sonuç bölümlerinden ana sorumlu yazar olarak çalışmada yer alırken ikinci araştırmacı yöntem, veri analizi, tartışma ve sonuç bölümlerinde katkı getirmiştir. / While the first researcher took part in the study as the main responsible author for the literature review, method, data analysis, discussion and conclusion sections, the second researcher contributed to the method, data analysis, discussion and conclusion sections.

Çıkar Çatışması / Conflict of Interest

Yazarlar çalışmanın herhangi bir çıkar çatışması olmadığını belirtmektedir. / The authors state that the study has no conflict of interest.

Fon Bilgileri / Funding

Yazarlar bu çalışma için herhangi bir fonları bulunmadığını beyan etmektedir. / The authors declare that they do not have any funds for this study.

Etik Kurul Onayı / The Ethical Commitee Approval

Bu araştırmada, tüm araştırmacılara açık, uluslararası veri tabanında yer alan veriler kullanıldığından ve herhangi bir canlı türünün üzerinde deneysel işlem yapılmadığından ve veri toplama sürecine gerek duyulmadığından etik kurul kararı gerektirmemektedir. / This study does not require an ethics committee decision, since data in an international database open to all researchers is used, no experimental procedures have been performed on any living species, and there is no need for a data collection process.

Kaynakça/References

Abela, R. (2020, Mart). How to hide the wordpess version from the generator meta tag. https://www.wpwhitesecurity.com/hidewordpress-version-number adresinden elde edildi.

Amsler, S. & Churchville, F. (2021, Mart). Content management system (cms). https://search contentmanagement. techtarget.com/definition/content-management-system-cms? adresinden elde edildi.

Aslam, N. (2019, Mayıs). 8 Easy methods top prevent ımage hotlinking ın wordpress. http://www.enquerer.com/ 8-easy-methods-to-prevent-image-hotlinking-in-wordpress adresinden elde edildi.

Bartley, M. (2020, Ocak). How to disable xml-rpc for better wordpress security. https://blogvault.net/ wordpress-disable-xmlrpc adresinden elde edildi.

Belani, G. (2019, Eylül). Ultimate guide to wordpress salts and security keys. https://www.wpexplorer.com/ wordpress-salts-security-keys/ adresinden elde edildi.

Bogdanovic, M. (2020, Eylül). How to disable wordpress theme and plugin editors from admin panel. https://qodeinteractive.com/magazine/disable-wordpress-theme-and-plugin-editors/ adresinden elde edildi.

Boiko, B. (2005). *Content management bible.* John Wiley & Sons.

CodeInWP, (2021, Ekim). 25 Simple wordpress security tricks to keep your website safe in 2020. https://www.codeinwp.com/blog/secure-your-wordpress-website adresinden elde edildi.

Collins, H. (2003). *Enterprise knowledge portals: next-generation portal solutions for dynamic information access, better decision making and maximum results*. Amacom Books.

Corbin, J. & Strauss, A. (2008). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Thousand Oaks: Sage.

Duò, M. (2021, Mart). How to find your wordpress login url (change it, lock it down). https://kinsta.com/ blog/wordpress-login-url adresinden elde edildi.

Hallikainen, P., Kivijarvi, H., & Nurmimaki, K. (2002, January). Evaluating strategic iy investments: an assessment of investment alternatives for a web content management system. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 2977-2986). IEEE.

Holcombe, J. (2021, Mart). How to check for security updates in wordPress. https://www.greengeeks.com /tutorials/article/check-for-security-updates-in-wordpress adresinden elde edildi.

Hughes, J. (2019, Aralık). Wordpress ddos protection:5 methods to secure your website. https://themeisle.com/blog/wordpress-ddos-protection adresinden elde edildi.

Jackson, B. (2021, Ekim). 17 Best wordpress security plugins to lock out the bad guys. https://kinsta.com/blog/ wordpress-security-plugins/ adresinden elde edildi.

Kinsta, (2020, Eylül). What ıs a content management system (cms)?. https://kinsta.com/knowledgebase/content-management-system/ adresinden elde edildi.

Kohan, B. (2010, Kasım). What is a content management system (cms)?. https://www.comentum.com/what-is-cms-content-management-system.html adresinden elde edildi.

Paivarinta T. & Munkvold B. (2005, January). Enterprise content management: an integrated perspective on information management. In *Proceedings of the 38th annual hawaii international conference on system sciences* (pp. 96-96). IEEE.

Paulsen,, K. (2012). *Moving media storage technologies: application & workflows for video and media server platforms*. Routledge.

Ray, J. (2018, Aralık). How to add http security headers in wordpress. https://www.tripwire.com/state-of-security/risk-based-security-for-executives/risk-management/how-add-http-security-headers-wordpress adresinden elde edildi.

Rockley A., Kostur P., & Manning S. (2003). *Managing enterprise content: A unified content strategy*. New Riders.

Smith, H. A. & McKeen, J. D. (2003). Developments in practice VIII: Enterprise content management. *The communications of the association for information systems, 11*(1), 41.

Vetch, P. (2006). *The content management handbook*. Martin White.

Wach, E. & Ward, R. (2013). Learning about qualitative document analysis. *IDS Practice Paper in Brief, 13*, 1-11.

WordPress. (2021a, Mart). Two step authentication. https://wordpress.org/support/article/two-step-authentication adresinden elde edildi.

WordPress. (2021b, Mart). Wordpress backups. https://wordpress.org/support/article/wordpressbackups adresinden elde edildi.

WPBeginner. (2017c, Mayıs). 12 Most useful htaccess tricks for wordpress. https://www.wpbeginner.com/wp-tutorials/9-most-useful-htaccess-tricks-for-wordpress adresinden elde edildi.

WPBeginner. (2019b, Mayıs). How to properly move wordpress from http to http (Beginner's Guide). https://www.wpbeginner.com/wp-tutorials/how-to-add-ssl-and-https-in-wordpress/ adresinden elde edildi.

WPBeginner. (2021a, Ocak). The ultimate wordpress security guide – step by step. https://www.wpbeginner.com/wordpress-security adresinden elde edildi.

Wright, K. (2019, Ağustos). WordPress file permissions: A guide to securing your website. https://ithemes.com/wordpress-file-permissions/ adresinden elde edildi.