WordPress Güvenliği

Haktan EMİK

Şubat 2020

İçindekiler

Giriş

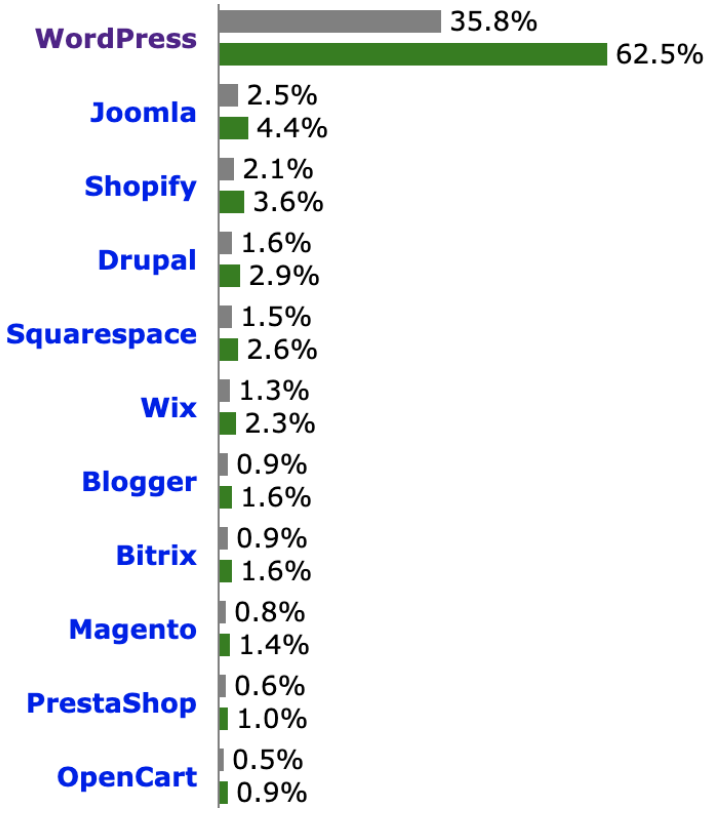
Bu doküman, WordPress tabanlı web uygulamalarında dikkat edilmesi gereken noktalar ve güvenliğini arttırmaya yönelik sıkılaştırma teknikleri ile ilgili bilgi vermek amacıyla hazırlanmıştır.

WordPress Hakkında

Wordpress GPL lisanslı bir içerik yönetim (CMS) sistemidir. PHP dili kullanılarak geliştirilmiştir. Yapılan araştırmalar sonucunda en çok kullanılan içerik yönetim sistemi olduğu bilinmektedir. Kurulum ve kullanım kolaylığı, geniş kitlelere hitap etmesi WordPress'in bu denli popüler olmasının başlıca nedenlerindendir. WordPress altyapısı, kişisel blog, haber ve e-ticaret uygulaması gibi aklınıza gelebilecek her alanda kullanılmaktadır.

WordPress, ilk çıktığı tarihlerden günümüze kadar sürekli gelişmekte ve kullanım oranı da gittikçe artmaktadır. W3techs adresinden alınan WordPress kullanımı ile ilgili istatistiksel bilgiler aşağıdaki gibidir;

CMS'ler ve diğer web sayfaları ile kıyaslandığında, en çok kullanılan içerik yönetim sistemi olduğu görülmektedir.



"WordPress is used by 35.8% of all the websites, that is a content management system market share of 62.5%"

WordPress'in kullanım oranları incelendiğinde, giderek yaygınlaştığı görülmektedir.

2011 1 Jan	2012 1 Jan	2013 1 Jan	2014 1 Jan	2015 1 Jan	2016 1 Jan	2017 1 Jan	2018 1 Jan	2019 1 Jan	2020 1 Jan	2020 16 Feb
%13.1	%15.8	%17.4	%21.0	%23.3	%25.6	%27.3	%29.3	%32.7	%35.4	%35.8

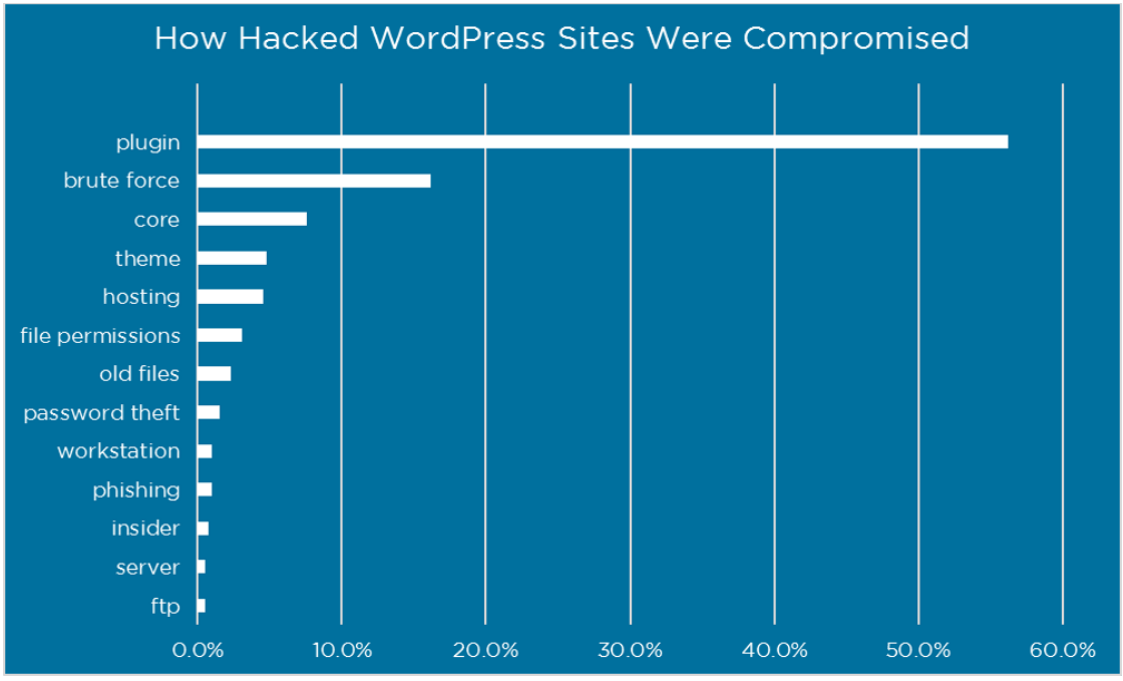
"Historical yearly trends in the usage statistics of content management systems"

Sıkça Yapılan Hatalar

WordPress altyapısına sahip sistemlerde, güvenlik problemlerine neden olan hatalar ve eksiklikler incelendiğinde en sık karşılaşılan maddeler aşağıdaki gibidir;

- Basit parola kullanımı
- Güncel olmayan sürüm kullanımı
- Zafiyetli tema/eklenti kullanımı
- Yapılandırma hataları (dizin listeleme, bilgi ifşaları ve erişilebilir yönetim paneli vs.)

Wordfence tarafından 2016 yılında yapılan “Saldırganlar WordPress sayfalarını nasıl ele geçiriyor” çalışması incelendiğinde eklenti ve kaba kuvvet (brute force) etkenlerinin üst sıralarda olduğu görülmektedir.



“How Attackers Gain Access to WordPress Sites”

WordPress Güvenliği

Bu başlık altında, WordPress kurulumu öncesi ve sonrasında genel olarak dikkat edilmesi gereken noktalar ve uygulamanın güvenliğini arttırmaya yönelik ek koruma yöntemlerine değinilecektir.

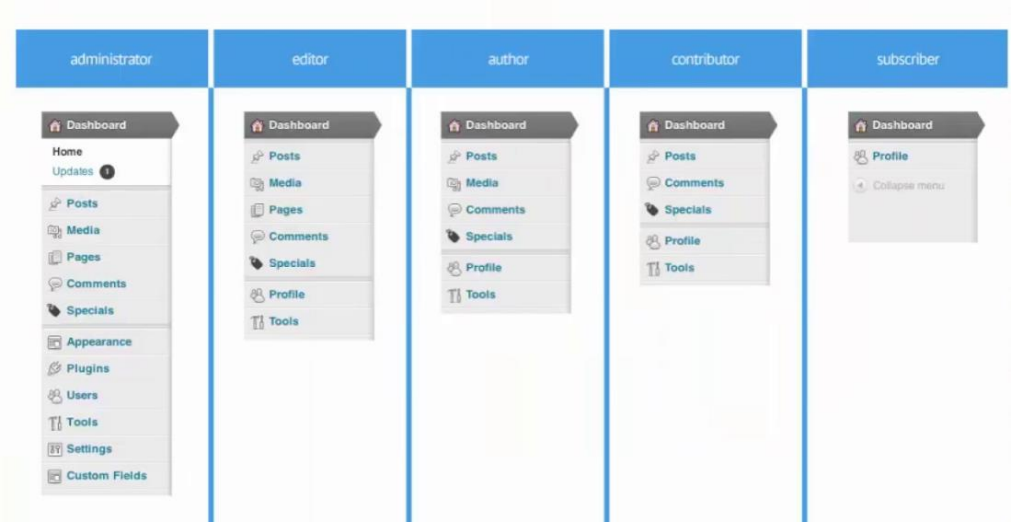
Sıkılaştırma

Sistemimiz üzerinde, bazı ek kontroller uygulayarak, uygulamamızı daha güvenli bir hale getirebiliriz. Genel hatlarıyla, dikkat edilmesi gereken noktalar ve yapılması önerilen ayarlamalara aşağıda yer verilmiştir.

- WordPress kurulumu yaparken, sıkça yapılan hatalarda birisi, yönetici hesap adı olarak admin, administrator ve root gibi bilinen kullanıcı adlarının kullanılmasıdır. Bu durum kaba kuvvet saldırılarında, kullanıcı adının kolay tahmin edilebilir olması nedeniyle sadece parola alanına denemeler yapılacağı için saldırganların işinin kolaylaştırmaktadır.

- Hesap parolaları belirlerken, güçlü ve tahmin edilemeyen bir parola kullanılmalıdır. Yönetici ekranından parola belirlenirken, sistem güçlü parola kullanımına zorlar. Ancak yaşanan olaylar, kullanıcıların zayıf parola kullanmayı onaylayarak, basit ve tahmin edilebilir parolalar kullandığını gözler önüne seriyor.

- WordPress'te birbirinden farklı kullanıcı rolleri mevcuttur. Her rolün yetkileri birbirinden farklıdır. Uygulamadaki roller, en az yetkiye sahip olacak şekilde tanımlanmalıdır. Örneğin, içerik girişi yapacak bir kullanıcının, "administrator" rolünde olmasına gerek yoktur. "Yazar" rolünde tanımlanması ihtiyacı karşılayacaktır. Ayrıca "administrator" rolüne sahip bir kullanıcı ile içerik eklenmesi, yönetici kullanıcı adının ifşa olmasına sebep olacağı için önerilen bir yöntem değildir. "Yazar" veya "editör" rolüne sahip kullanıcılar tarafından içerik sağlanması önerilmektedir.



WordPress'teki kullanıcı rolleri

- Giriş işlemi, iki aşamalı doğrulama yöntemi ile yapılmalıdır. (Örneğin: Google Authenticator)

- WordPress kurulumu sonrasındaki yönetici paneline giriş yolu "wp-admin" adresidir. Yönetim paneline, sadece izin verilen IP adreslerinden erişim sağlanacağı şekilde kısıtlama uygulanmalıdır.

- Giriş ekranı, yorum ekranı ve iletişim formu gibi veri gönderme/sorgulama fonksiyonlarına captcha korumasının eklenmesi önerilmektedir. Böylelikle, kaba kuvvet saldırılarına ve botlara karşı koruma sağlanacaktır.

- WordPress, yeni özellikler ve güvenlik yamaları için her ay güncelleme yayınlamaktadır. Bu güncellemeler takip edilmeli ve her zaman en güncel sürüm kullanılmalıdır. WordPress, otomatik güncelleme desteği de sunmaktadır. Bu işlem, yönetim panelinden yapılacağı gibi wp-config.php dosyasına aşağıdaki kod satırının eklenmesiyle de yapabilmektedir.

```
define( 'WP_AUTO_UPDATE_CORE', true ); #WordPress core için
```

- WordPress için hazırlanan birçok tema ve eklenti mevcuttur. Bu temalar ve eklentiler kolaylıkla sisteme yüklenip, kullanılabilmektedir. Wordfence tarafından yapılan araştırmada da görüldüğü üzere tema ve eklenti kaynaklı güvenlik zafiyetleri sebebiyle WordPress sitelerin kontrolü saldırganların eline geçebilmektedir. Bu nedenle, her tema ve eklenti yüklenmemeli, sadece güvenilir kaynaklar tarafından yayınlanan temalar ve eklentiler tercih edilmelidir. Uygulamamızın atak yüzeyinin artmaması için en az seviyede eklenti kullanımı önerilmektedir. Tema ve eklenti güncellemeleri takip edilerek en güncel sürüm kullanılmalıdır. WordPress, tema ve eklentiler içinde otomatik güncelleme desteği sunmaktadır.

- WordPress için wp-config.php dosyası hayati önem taşımaktadır. Bu dosya içerisinde veri tabanı bağlantı bilgileri gibi kritik öneme sahip bilgiler yer almaktadır. Bu nedenle, wp-config.php dosyasının webroot dışına taşınması, ek koruma sağlayacaktır.

- WordPress Güvenlik Anahtarları, oturum değerlerinin (cookie'lerin) daha güçlü şifrelenmesinde kullanılmaktadır. https://api.wordpress.org/secret-key/1.1/salt/ adresinden alınan değerlerin, wp-config.php dosyasına eklenmesiyle ek koruma sağlanacaktır.

- WordPress, pingback özelliği için XML-RPC'yi kullanmaktadır. Uygulamanız, bu özelliği kullanmıyor ise xmlrpc.php dosyasına erişim kısıtlanmalıdır. Aksi takdirde xmlrpc.php üzerinden kaba kuvvet ve hizmet dışı bırakma saldırıları yapılabilmektedir.

- WordPress REST API üzerinden sistemde kayıtlı kullanıcı bilgileri ve uygulama endpointleri gibi bilgiler açığa çıkabilmektedir. Bu sebeple REST API erişimlerinin kısıtlanması önerilmektedir.
  (Örneğin: example.com/wp-json/wp/v2/users/ yolu üzerinden kullanıcı bilgileri açığa çıkmaktadır.)

- WordPress kurulumu yapılırken, wp-config.php dosyasında veri tabanı tabloları için ön ek (prefix) belirlenmektedir. Veri tabanı ön ekinin varsayılan olarak (wp_) bırakılmaması önerilmektedir.

- Cookie'lerin "secure" bayrağının işaretlenmesi önerilmektedir. Secure bayrağının işaretlenmesiyle, cookie'nin sadece HTTPS bağlantılar üzerinden taşınması sağlanacaktır. Bu işlem için php.ini dosyasına aşağıdaki kod satırı eklenmelidir.

```
session.cookie_secure=1
```

- phpMyAdmin ile web üzerinden veri tabanı yönetimi yapılabilmektedir. Ancak web üzerinden erişilebilir olması risk doğurabilir. Bu nedenle, sadece yetkili kişiler tarafından erişim sağlanacak şekilde kısıtlanması önerilmektedir.

- Önemli/Hassas dosyalara erişim kısıtlanmalıdır. Bu işlemler .htaccess dosyası üzerinden yapılabilir. Aşağıdaki örnekte .htaccess dosyası üzerinden .htaccess ve wp-config.php dosyalarına erişim kısıtlanmıştır.

```
<files .htaccess>
order allow,deny
deny from all
</files>
```

```
<files wp-config.php>
order allow,deny
deny from all
</files>
```

- Dizin listeme kapatılmalıdır. Bu işlem için .htaccess dosyasına aşağıdaki kod satırı eklenmelidir.

```
Options All –Indexes
```

- Sunucu imzası, sistem hakkında bilgi ifşasına sebep olacağı için kapatılması önerilmektedir. Bu işlem için .htaccess dosyasına aşağıdaki kod satırı eklenmelidir. Aynı şekilde dönen cevaplardan, sunucu versiyonu ve kullanılan teknoloji bilgisinin kaldırılması önerilir.

```
ServerSignature Off
```

- Sistem üzerinde kullanılan WordPress sürümünün tespit edilmesi, saldırganların işini kolaylaştıracaktır. Bu sebeple, uygulama üzerinden versiyon bilgisi kaldırılmalıdır. (Versiyon bilgisi; generator, readme.html ve JS dosyaları üzerinden öğrenilebilir.) Bu işlem için function.php dosyası üzerine aşağıdaki kod parçası eklenmelidir;

```
remove_action(‘wp_head’, ’wp_generator’);
```

- WordPress, yönetim paneli üzerinden dosya düzenleme özelliğine sahiptir. Bu özellik kullanım açısından kolaylıklar sağlasa da, yönetim paneline erişim sağlayan bir saldırganın, odaklanacağı ilk nokta olacaktır. Bu sebeple, dosya düzenleme özelliğinin kapatılması önerilmektedir. Bu işlem için wp-config.php dosyasına aşağıdaki kod parçasının eklenmesi gerekmektedir.

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

Ek olarak tema ve eklenti yüklemelerinde de aynı durum söz konusu olmasından dolayı yönetici paneli üzerinden tema ve eklenti yüklenmesinin engellenmesi önerilmektedir. Aksi takdirde, yönetici paneline erişim sağlayan saldırgan, zararlı kod içeren tema veya eklenti dosyalarını sunucuya yükleyerek, sistemi ele geçirebilir.

- Sistemdeki, yönetici veya yazar gibi kullanıcı adlarının açığa çıkarılması, saldırganların kaba kuvvet saldırılarında işini kolaylaştırmaktadır. Bu durumun önüne geçebilmek için bilgi ifşasına yol açan adresler için kısıtlama getirilmelidir. “author” parametresi üzerinden kullanıcı adların açığa çıkarılmasını engellemek için .htaccess dosyasına aşağıdaki kod satırı eklenmelidir.
(Örneğin, example.com/?author=1 ve example.com/wp-json/wp/v2/users)

```
RewriteCond %{QUERY_STRING} author=d
RewriteRule ^ /? [L,R=301]
```

- Hata mesajları, eğer kontrol edilmez ise bilgi ifşalarına yol açabilmektedir. Bu durum, saldırganların sistem hakkında bilgi toplayarak daha spesifik saldırılar gerçekleştirmesine olanak tanıyabilir.

Bu nedenle PHP hata mesajlarının kapatılması önerilmektedir. Bu işlem için wp-config.php dosyasına aşağıdaki kod bloğu eklenmelidir;

```
@ini_set('display_errors', 0);
define('WP_DEBUG', false);
define('WP_DEBUG_DISPLAY', false);
```

- Kurulum sonrası gelen “readme.html” gibi bilgi ifşasına yol açabilecek dosyaların sunucu üzerinden kaldırılması önerilmektedir.

- PHP’nin, sistemi tehlikeye sokabilecek fonksiyonlarını (system, exec vs.) php.ini dosyasında “disabled_functions” tanımlamasıyla engellenmelidir.

- Dosya yükleme işlemi yapılırken, boyut sınırlaması belirlenmeli ve bu limit üzerindeki dosyaların kabul edilmemesi sağlanmalıdır. Aksi taktirde bu durum sistem üzerinde gereksiz kaynak tüketimine sebep olarak hizmet kesintisine neden olabilir.

- Arama motorlarında, sayfanın hangi adreslerinin gösterilip, gösterilmeyeceği bilgisini “robots.txt” dosyası üzerinden belirtebiliriz. Ancak bu dosyaya sadece arama motorlarının değil de, herkesin erişim sağlayabileceği sebebiyle, hassas/önemli bilgi içermemesi gerekmektedir.

- Dosya/klasör izinleri “777” olarak verilmemelidir. Sunucu üzerinde yeni bir kullanıcı oluşturarak, uygulama dosyaları bu kullanıcıya sahiplendirilmelidir.

- Sitenizi, paylaşımlı sunucular üzerinde barındırmamalısınız. Bu sunucular üzerinde birçok site yer almaktadır. Hal böyle olunca da sitenizin güvenliğini makul seviyede sağlasanız bile, sunucuda barınan diğer siteler üzerinden siteniz tehlike altında olacaktır.

- İletişim, SSL üzerinden sağlanmalıdır.

- WAF (Web Application Firewall) kullanılması önerilmektedir. Böylelikle, HTTP istekleri kontrol edilerek, zararlı bir trafik algılanması durumunda engellenecektir.

- FTP yerine SFTP üzerinden dosya transferi gerçekleştirilmelidir. FTP üzerinden iletişim düz metin olarak sağlandığı için olası bir ortadaki adam saldırısına karşı savunmasız durumdadır.

- Veri tabanı "root" kullanıcısı ile çalışmamalıdır. Uygulamamızın kullandığı veri tabanı kullanıcısı, veri tabanının yönetim kullanıcısı olmamalıdır. Veri tabanı kullanıcımız en az yetkiler ile çalışmalıdır. Veri tabanı kullanıcımızın parolası, güvenlik politikasına uygun olarak belirlenmelidir.

- Veri tabanındaki "anonymous" kullanıcılar silinmelidir.

- Veri tabanı portuna dışarıdan erişilmesine izin verilmemelidir. Sadece "localhost" üzerinden erişim sağlanacağı şekilde yapılandırılmalıdır. Bu işlem için my.cnf dosyasına aşağıdaki kod parçası eklenmelidir;

```
bind-address=127.0.0.1
```

- "Local_infile" gibi dosya erişim fonksiyonları kapatılmalıdır. Bu işlem için my.cnf dosyasına aşağıdaki kod parçası eklenmelidir;

```
local-infile=0
```

- Veri tabanı log kayıtlarının tutulması önerilmektedir. Kayıtların nereye yazılacağı bilgisi my.cnf dosyasından ayarlanabilir;

```
log=/var/log/mysql.log
```

- SQL history ("~/.mysql_history") kayıtlarının temizlenmesi önerilmektedir.

- Veri tabanı trafiği, SSL/TLS üzerinden sağlanmalıdır.

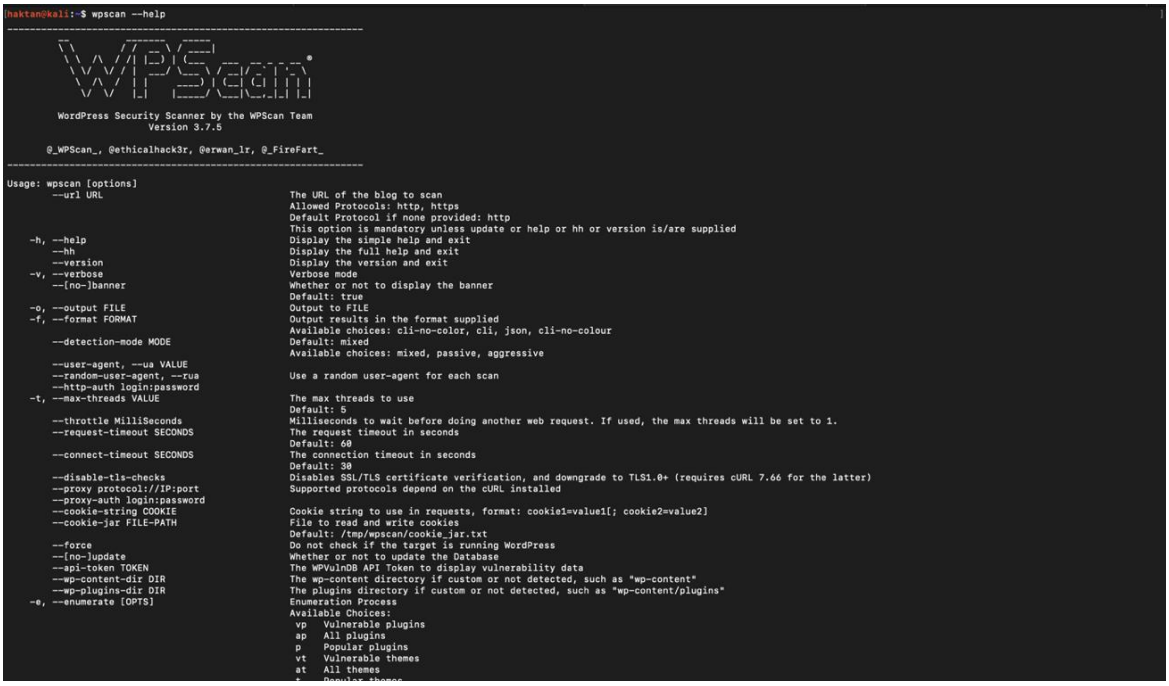
- Uygulamada, herhangi bir duruma karşı düzenli aralıklar ile yedekleme işlemi yapılmalıdır. Alınan yedeklerin, uygulama üzerinde saklanmaması önerilmektedir.

WPScan

WPScan, WordPress tabanlı uygulamalar için geliştirilmiş bir zafiyet tespit aracıdır. WPScan ile;

- WordPress sürüm tespiti
- Kullanılan tema ve eklentileri tespiti
- Tema ve eklenti sürüm tespiti
- WordPress çekirdeği, tema ve eklenti kaynaklı güvenlik zafiyetlerinin tespiti
- Kullanıcı tespiti
- Kaba kuvvet saldırısı gerçekleştirebilme
- …

ve daha birçok işlem yapılabilmektedir. Aşağıda, aracın kullanımı ile ilgili bazı örnekler verilmiştir.



Örnek kullanımlar:

```
wpscan --url https://example.com

wpscan --url https://example.com --enumerate p # kullanılan eklentiler ile ilgili bilgi toplanması

wpscan --url https://example.com --enumerate vt # kullanılan temanın zafiyet konrolünün yapılması

wpscan --url https://example.com --enumerate u # sistemdeki kullanıcılar ile ilgili bilgi toplanması

wpscan --url https://example.com --wordlist wordlist.txt # verilen dosya ile brute force atağı gerçekleştirme
```

Kapanış

Genel hatlarıyla, dikkat edilmesi noktalar ve ek koruma yöntemlerine değinilmiştir. Yapılacak olan sıkılaştırma yöntemleriyle birlikte uygulamamızı daha güvenli bir hale getirebiliriz. Ancak, güvenliğin bir bütün olarak ele alınması gerektiği unutulmamalıdır. Sadece web uygulaması üzerinde değil, sunucu ve veri tabanı üzerinde de güvenlik kontrolleri sağlanmalıdır.

Kaynaklar

- https://tr.wikipedia.org/wiki/WordPress
- https://w3techs.com/technologies/overview/content_management
- https://www.wordfence.com/blog/2016/03/attackers-gain-access-wordpress-sites/
- https://wordpress.org/support/article/hardening-wordpress/
- https://www.owasp.org/index.php/OWASP_Wordpress_Security_Implementation_Guideline
- https://www.mehmetince.net/wordpress-guvenligi-sunucu-guvenligi/
- https://www.tecmint.com/mysql-mariadb-security-best-practices-for-linux/
- https://wpscan.org/
- https://wpvulndb.com/