WHITEAB
سرویس ابری ارزیابی امنیتی

گزارش فنی

سیستم بررسی امنیتی خودکار وایت‌لب

تست نفوذ برنامه تحت وب

ارائه شده توسط آزمایشگاه امنیت وایت‌لب

WhiteLab.ir | @WhiteLabir | info@WhiteLab.ir

بررسی امنیتی وب سایت 192.168.1.181

اطلاعات بررسی وب سایت

مشخصات پایه			
مدت زمان بررسی	26 دقیقه و 7 ثانیه	زمان شروع	12:15:48 ,08/03/2019
آدرس وب سایت	/http://192.168.1.181/testt	هدف	192.168.1.181

مشخصات دیگر	
اطلاعات میزبان	Apache/2.4.18 (Ubuntu)

سطح آسیب پذیری

وضعیت وب‌سایت از نظر امنیتی خطرناک است

یک یا چند آسیب پذیری با اهمیت بالا توسط کاوشگر کشف شده اند

امکان سواستفاده توسط فرد مهاجم وجود دارد

هشدار های یافت شده 8	
خطرناک	2
ریسک پذیر	2
عادی	1
دیگر اطلاعات	1

خطرناک

WordPress 4.2.2 Multiple Vulnerabilities (0.7 - 4.2.2)

موارد تحت اثر

/wordpress/

شرح

نسخه‌های کمتر و شامل 4.2.2 وردپرس به چندین آسیب‌پذیری شامل cross-site scripting و آسیب‌پذیری‌های دورزدن مکانیزم امنیتی آسیب‌پذیر هستند.
سوء استفاده از این موارد به حمله‌کننده این اجازه را می‌دهد که کدهای دلخواه را در مرورگر کاربر در فضای وب‌سایت فعلی اجرا کند، موارد احرازهویت مبتنی بر کوکی را بدزدد و حملات دیگری اعمال کند و اعمال محدود شده انجام دهد.

تاثیر

-

پیشنهاد

به آخرین نسخه وردپرس به روزرسانی انجام دهید

منابع

http://seclists.org/oss-sec/2015/q3/187

https://wordpress.org/news/2015/07/wordpress-4-2-3/

CVE-2015-5622

CVE-2015-5623

جزئیات

نسخه فعلی: 3.9.27
وردپرس نسخه کمتر از 4.2.2 و بیشتر از 0.7 آسیب‌پذیر هستند.

هدرهای درخواست

```
Cookie: wordpress_test_cookie=WP+Cookie+check
Accept: */*
Accept-Encoding: gzip,deflate
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML,
like Gecko) Chrome/41.0.2228.0 Safari/537.21
Connection: Keep-alive
```

WordPress Possible Security Bypass Vulnerability (0.70 - 4.7.4)

خطرناک

موارد تحت اثر

/wordpress/

شرح

وردپرس در نسخه‌های بالاتر از 0.70 و کمتر از 4.7.4 (شامل این ورژن) مستعد آسیب‌پذیری رد کردن مکانیزم امنیتی است. سوء استفاده از این مورد می‌تواند به انجام اعمال محدود شده و در نتیجه تنظیم مجدد رمز عبور کاربر و دسترسی احراز هویت نشده به حساب کاربری آنها شود.

تاثیر

-

پیشنهاد

به عنوان یک راه‌حل موقتی کاربران می‌توانند با فعال‌سازی امکان UseCanonicalName تنظیم SERVER_NAME به صورت ثابت را اجبار کنند. همچنین می‌توانند وصله ارائه شده در منابع را مد نظر قرار دهند.

منابع

https://exploitbox.io/vuln/WordPress-Eexample-4-7-Unauth-Password-Reset-0day-CVE-2017-8295.html

https://www.exploit-db.com/exploits/41963/

https://gist.github.com/Neo23x0/9555f052c4e222043e6d8a44e34f5455

CVE-2017-8295

جزئیات

نسخه فعلی: 3.9.27
نسخه‌های وردپرس بین 0.70 تا 4.7.4 آسیب‌پذیر هستند.

هدرهای درخواست

```
GET /wordpress/ HTTP/1.1
Cookie: wordpress_test_cookie=WP+Cookie+check
Accept: */*
Accept-Encoding: gzip,deflate
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML,
like Gecko) Chrome/41.0.2228.0 Safari/537.21
Connection: Keep-alive
```

Directory listing	ریسک پذیر
/wordpress/wp-includes/ID3/	موارد تحت اثر

شرح

وب‌سرور به این صورت تنظیم شده است که محتویات داخل مسیر فعلی را نمایش دهد. این عمل پیشنهاد نمی‌شود از این جهت که ممکن است مسیر حاوی فایل‌هایی

باشد که به صورت عادی از طریق لینک‌های وب‌سایت به بیرون افشا نمی‌شوند.

تاثیر

یک کاربر ممکن است لیست فایل‌های موجود در این مسیر را ببیند که در نتیجه اطلاعات حساسی ممکن است افشا شود.

پیشنهاد

شما باید اطمینان حاصل کنید که در مسیرهای فوق اطلاعات حساسی موجود نمی‌باشد یا مد نظر داشته باشید که از طریق تنظیمات وب‌سرور دسترسی به لیست محتویات مسیرها را محدود کنید.

منابع

CWE-548: Information Exposure Through Directory Listing

جزئیات

-

هدرهای درخواست

```
GET /wordpress/wp-includes/ID3/ HTTP/1.1
Accept: */*
Accept-Encoding: gzip,deflate
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0
Safari/537.21
```

```
Connection: Keep-alive
```

ریسک پذیر	WordPress XML-RPC authentication brute force
موارد تحت اثر	/wordpress/xmlrpc.php

شرح

وردپرس یک رابط XML-RPC را از طریق اسکریپت xmlrpc.php ارائه می‌دهد. XML-RPC یک روند فراخوانی رویه از راه دور از طریق HTTP و XML است. حمله‌کننده می‌تواند از این رابط و به طور خاص فراخوانی wp.getUsersBlogs استفاده کند تا حمله گسترده به دست آوردن اطلاعات احراز هویتی اجرا کند.

تاثیر

حمله‌کننده می‌تواند اطلاعات احراز هویتی برای بلاگ وردپرس شما را بروت فورس کند.

پیشنهاد

در صورت امکان اسکریپت XML-RPC را غیر فعال کنید در غیر این صورت خطای احراز هویت XML-RPC را زیر نظر بگیرید و از طریق فایروال‌های مختص وب مانند ModeSecurity جلوی این امکان را بگیرید.

منابع

[WordPress XML-RPC Brute Force Scanning]

[Prevent XMLRPC]

[WordPress brute force attack via wp.getUsersBlogs]

جزئیات

پاسخ یافت‌شده:

```
<value><string>Incorrect username or password.</string></value>
```

هدرهای درخواست

```
POST /wordpress//xmlrpc.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive
Cookie: wordpress_test_cookie=WP+Cookie+check
Accept: */*
Accept-Encoding: gzip,deflate
Content-Length: 264
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0
Safari/537.21
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value><string>admin</string></value></param>
<param><value><string>894758954378954375349 87</string></value>
</param>
</params>
</methodCall>
```

Clickjacking: X-Frame-Options header missing

عادی

موارد تحت اثر

Web Server

شرح

مشخصه X-Frame-Options در هدر پاسخ ارسالی از سمت سرور مشخص کننده این است که آیا اجازه استفاده از این وب‌سایت در <iframe> وجود دارد یا وجود ندارد. به طور دقیق‌تر با حملات ClickJacking با اعمال این هدر مقابله خواهد شد. این حملات

ممکن است به اعمال عمل‌هایی بدون رضایت کاربر توسط حمله‌کننده منجر شود.

سرور هدر X-Frame-Options را در پاسخ ارسالی تنظیم نکرده است که به معنی وجود ریسک اعمال حملات ClickJacking است. اعمال این هدر در پاسخ ارسالی می‌تواند به مرورگر کاربر این اطلاع را بدهد که آیا مجاز به نمایش یک صفحه در یک frame یا iframe است یا خیر. سایت‌ها میتوانند با تنظیم این هدر از حملات ClickJacking جلوگیری کنند.

پیشنهاد

وب‌سرور خود را به این صورت تنظیم کنید که در هدر پاسخ X-Frame-Options را ارائه دهد. همچنین می‌توانید برای اطلاعات بیشتر و تکمیلی به منابع ارائه شده برای محتوی دقیق‌تر این هدر رجوع کنید.

منابع

The X-Frame-Options response header

Clickjacking

OWASP Clickjacking

Defending with Content Security Policy frame-ancestors directive

Frame Buster Buster

جزئیات

-

هدرهای درخواست

```
GET / HTTP/1.1
Connection: keep-alive
Accept: */*
```

```
Accept-Encoding: gzip,deflate
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0
Safari/537.21
```

Content Security Policy (CSP) not implemented	سایر اطلاعات
-	موارد تحت اثر

شرح

مشخصه Content-Security-Policy در هدر پاسخ ارسالی از سمت سرور مشخص کننده آدرس‌ها و منابع مجازی است که در صفحه قابل بارگیری است. این مکانیزم مقابله در برابر حملات XSS را تقویت می‌کند.
برای پیاده‌سازی این مکانیزم امنیتی شما می‌بایست آدرس منابع مجاز برای بارگیری در صفحه را در هدر Content-Security-Policy مشخص کنید. برای مثال اگر صفحه شما لازم است که اسکریپت‌ها یا تصویرهایی که به صورت محلی هستند و همچنین کتابخانه jQuery را از CDN بارگیری کند هدر مورد نظر می‌تواند به صورت زیر باشد:

```
Content-Security-Policy:
default-src 'self';
script-src 'self' https://code.jquery.com;
```

مشخص شد که این هدر از طریق برنامه تحت وب شما پیاده‌سازی نشده و در هدر پاسخ ارسالی وجود ندارد.

تاثیر

CSP می‌تواند از حملاتی که شامل تزریق کد و محتوی مانند حملات XSS، حملاتی که

نیاز به تعبیه‌کردن یک منبع مخرب و حملاتی که شامل استفاده غیر مجاز از iframes هستند مانند حملات clickjacking و غیره جلوگیری کند.

پیشنهاد

پیشنهاد می‌شود که CSP را در برنامه تحت وب خود پیاده‌سازی کنید.
تنظیم این مشخصه شامل اضافه کردن هدر **Content-Security-Policy** در پاسخ ارسالی یک صفحه از سرور و کنترل منابعی که مرورگر کاربر مجاز به بارگیری برای آن صفحه است

منابع

Content Security Policy (CSP)

Implementing Content Security Policy

جزئیات

-

هدرهای درخواست

```
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip,deflate
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0
Safari/537.21
Connection: Keep-alive
```