



网络安全预警通报

2026 年第 9 期（总第 67 期）

西安交通大学网络信息中心 2026 年 10 月 22 日

【预警类型】严重预警

【预警内容】

PHP 内核远程代码执行及内存泄露安全风险预警通报

一、漏洞概述

2026 年 10 月 21 日，PHP 官方发布安全公告，针对 PHP 解释器内核中的两项高危安全漏洞发布安全更新，漏洞编号分别为 CVE-2026-109618 和 CVE-2026-109619。相关漏洞被统称为“PHP2Shell”，涉及内存耗尽处理异常、陈旧全局状态及释放后使用（Use-After-Free, UAF）等问题，可能导致攻击者在未经过身份认证的情况下远程执行代码，或读取 PHP 进程内存中的敏感信息。

根据公开披露的技术分析，研究人员已在 WordPress、Drupal、Joomla 等主流内容管理系统中演示了未认证远程代码执行攻击，并验证了针对 PHP 应用的远程内存泄露风险。由于漏洞位于 PHP 解释器内核，影响范围不仅限于特定网站程序或应用框架，运行于受影响 PHP 版本上的其他 Web 应用同样可能面临安全风险。

该漏洞具备以下特点：

(1) 漏洞影响范围广：漏洞存在于 PHP 运行时内核，涉及多个 PHP 版本分支及 PHP-FPM、Apache mod_php、php-cgi 等 Web 服务运行方式。

(2) 攻击门槛较低：在受影响的部署环境中，攻击者可能通过构造特定 HTTP 请求触发漏洞，无需有效账号即可实施攻击。

(3) 安全影响严重：漏洞利用成功后，可能导致远程代码执行、服务器权限被非法获取、敏感信息泄露及业务系统遭到破坏。

(4) 漏洞利用具有隐蔽性；部分攻击路径可直接读取 PHP 进程内存中的敏感数据，不一定需要在服务器上写入恶意文件或执行恶意程序，传统基于 Webshell 和异常进程的检测手段可能无法及时发现。

(5) 存在现实攻击风险：公开信息显示，漏洞披露期间已有安全研究机构监测到相关利用尝试，需重点关注互联网暴露的 PHP 应用及未及时更新的服务器。

二、漏洞详情

1.技术原理

本次 PHP2Shell 系列漏洞主要涉及两类安全问题。

攻击流程如下：

第一类为内存耗尽处理异常导致的内存破坏漏洞（CVE-2026-109618）。PHP 在处理特定 multipart/form-data 请求时，可能因异常内存分配失败而进入不完整的错误处理流程，导致部分内部状态或指针未得到正确初始化及清理。在后续请求处理或资源释放阶段，相关异常状态可

能被继续访问，从而形成内存破坏，并在特定环境下被利用实现远程代码执行。

第二类为陈旧全局状态引发的释放后使用漏洞 (CVE-2026-109619)。PHP 在请求处理过程中维护部分全局状态，某些异常退出路径可能导致请求级数据在释放后仍被全局状态引用。攻击者通过构造特定请求，在满足相关运行条件时，可能使后续请求访问已释放的内存区域，进而造成内存破坏、敏感信息泄露或远程代码执行。

2.涉及漏洞详情

漏洞编号：CVE-2026-109618

漏洞名称：PHP 内存耗尽处理异常导致的内存破坏漏洞

漏洞类型：内存破坏、远程代码执行

漏洞等级：严重

CVSS v4.0 评分：9.2

影响组件：PHP 解释器内核

漏洞编号：CVE-2026-109619

漏洞名称：PHP 陈旧全局状态释放后使用漏洞

漏洞类型：释放后使用、内存泄露、远程代码执行

漏洞等级：严重

CVSS v4.0 评分：9.2

影响组件：PHP 解释器内核

三、漏洞影响范围

1.影响范围

目前确认风险主要涉及：

- 使用 PHP 8.2、8.3、8.4、8.5 受影响版本的 Web 服务器；
- 使用 PHP-FPM、Apache mod_php、php-cgi、FrankenPHP、LiteSpeed 等运行方式提供 Web 服务的系统；
- 部署 WordPress、Drupal、Joomla、MediaWiki、Magento 等 PHP 应用的平台；
- 运行自研 PHP 网站、业务管理系统及其他 PHP 后端服务的服务器；
- 直接暴露互联网、未及时安装安全更新或使用停止维护 PHP 版本的服务器；
- 通过容器镜像、虚拟主机管理面板或第三方软件捆绑 PHP 运行时的业务系统。

需要注意的是，漏洞位于 PHP 解释器内核，并不要求目标系统安装 WordPress、Drupal 或 Joomla 等特定应用。只要实际 Web 请求由受影响的 PHP 运行时处理，就应开展风险排查。

2.威胁严重性

攻击成功后，可能造成以下安全影响：

（1）服务器被远程控制

攻击者利用远程代码执行漏洞后，可能以 PHP Web 服务进程的权限执行恶意指令，进一步上传 Webshell、植入后门程序或执行其他恶意操作，威胁服务器安全。

（2）敏感信息泄露

攻击者可能读取 PHP 进程内存中的会话数据、身份认证令牌、数据库凭据、API 密钥及其他敏感信息，并利用窃取的信息访问业务系统或相关服务。

(3) 业务系统中断

攻击者可能通过异常请求触发内存耗尽、PHP 工作进程崩溃或服务异常，导致网站及相关业务系统响应缓慢、请求失败或服务不可用。

(4) 业务数据及数据库安全受到威胁

在获得代码执行权限或窃取有效凭据后，攻击者可能进一步访问数据库、读取业务数据、篡改应用文件，甚至破坏重要业务信息。

(5) 攻击范围进一步扩大

若受影响服务器与其他内部系统之间存在信任关系、共享账号或复用凭据，攻击者可能利用已获取的权限和敏感信息进一步攻击内部其他主机，扩大安全事件影响范围。

四、处置建议

1.官方修复方案

及时安装安全更新。对于使用 PHP 8.5、8.4、8.3、8.2 的系统，应分别升级至 PHP 8.5.12、8.4.27、8.3.36、8.2.35 或更高的适用安全版本。

使用操作系统发行版维护包的系统，应优先依据供应商安全公告确认补丁状态，避免仅根据 PHP 版本字符串判断是否已修复。对于 PHP 8.1 及更早的旧版本，应核实供应商是否提供安全回补补丁；对于已停止维护的版本，应优先制定迁移至受支持 PHP 版本的计划。

升级完成后，应重启 Apache、PHP-FPM 及其他常驻 PHP 工作进程，确保实际处理业务请求的运行时已加载修复版本。同时，应核查容器镜像、主机管理面板及第三方软件捆绑的 PHP 运行时，防止遗漏未修复组件。