php代码审计基础篇

第6课 代码执行

漏洞介绍

代码执行漏洞

代码执行漏洞是指应用程序本身过滤不严，用户可以通过请求将代码注入到应用中执行。当应用在调用一些能将字符串转化成代码的函数(知php中的eval)时，没有考虑到用户是否能控制这个字符串，将造成代码执行漏洞。

狭义的代码注入通常指将可执行代码注入到当前页面中，

php代码执行漏洞就是可以把代码注入应用中最终到webServer去执行，这样的漏洞如果没有特殊的过滤，相当于直接有一个web后门(webshell)的存在

漏洞防范

对于eval()函数一定要保证用户不能轻易接触eval的参数或者用正则严格判断输入的数据格式。对于字符串一定要使用单引号包裹可控代码，并且插入前进行addslashes().

对于preg_replace放弃使用e修饰符。如果必须要用e修饰符，请保证第二个参数中，对于正则匹配出的对象，用单引号包裹

尽量不要执行外部的应用程序或命令

使用自定义函数或函数库来替代外部应用程序或命令的功能使用escappeshellarg函数来处理命令的参数使用sare_mode_exec_dir来指定可执行的文件路径将执行的参数做白名单限制，在代码或配置文件中限制某些参数

php代码审计

漏洞挖掘思路

存在可执行代码的危险函数.用户可控输入

eval() assert()

正则匹配函数

preg_replace()

mixed preg_replace(mixde $pattern,mixed $replacement,mixed $subject[int $limit =-1[,in&$count]])

回调函数:

mixed call_user_func(callable $callback[,mixed $parameter[,mixed $....]])

$callback是要调用的自定义函数名称$parameter是自定义函数的参数

array_map() call_user_func() call_user_func_array() 动态函数$_GET($_POST['cmd']

实例代码分析

ucms存在代码执行漏洞

谢谢观赏