PHP webshell 实时动态检测

杜海章　方　勇

（四川大学电子信息学院　四川　610065）

【摘　要】PHP Webshell 是一种使用 PHP 编写的可以在 web 服务器中执行的网站后门文件。现有网站采用的基于特征匹配检测和网络流量分析的 webshell 检测方法难以检测经过变形或加密的 webshell。本文提出了一种基于 PHP 扩展的 webshell 实时动态检测方法，该方法基于 PHP 扩展对 PHP 代码的编译执行进行监控并结合外部输入变量的标记追踪、黑白名单机制来进行 webshell 的实时动态检测。

【关键词】webshell 检测；PHP 扩展；实时动态

中图分类号：TP212.2　　文献标识码：A　　文章编号：1009-6833（2014）12-120-03

PHP webshell real-time dynamic detection

Du Haizhang，Fang Yong

Abstract：PHP webshell is a kind of backdoor written in php。Because the traditional webshell detection methods are difficult to detect confused and encrypted webshell，this paper puts forward a webshell detection method based on PHP extension. This method can real-time detect the webshell's running.

Keywords：webshell Detection；PHP extension；real-time

0　引言

在 web 服务器上上传 Webshell 是网站攻击者常见的用来控制 web 服务器以进行进一步进行渗透的方法，对 webshell 进行检测和防御是进行网站安全防御和降低网站损失的重要方法[1]。现阶段针对 PHP webshell 的检测技术主要分为两类，即静态特征检测技术和动态特征检测技术[2]。静态特征检测是指不需要代码运行，根据文件中是否存在常见的恶意字符串特征结合文件的信息熵等特征进行检测；动态特征检测是指在通过在 WAF 中对网络流量进行分析来检测和防御 webshell。这两类 webshell 检测方法都难以有效检测经过混淆变形或经过加密的 webshell。

鉴于现阶段检测 PHP webshell 的方法的不足，本文分析了 PHP 代码运行的流程和各种 webshell 变形在 PHP 内核中的特征，提出了一种基于 PHP 扩展的 webshell 检测和防御方法。该方法在 PHP 扩展中通过对 PHP 代码的编译和运行进行监控，结合对外部输入变量的标记追踪和黑白名单机制，可以实时有效的检测 webshell 和阻止 webshell 的运行。该方法不仅可以检测 webshell 还可以阻止 webshell 的运行。

1　PHP 原理

1.1　PHP 生命周期

PHP 在 web 容器上的运行方式主要有三种：以模块加载的方式运行；以 CGI 方式运行；以 FastCGI 的方式运行。不管采用哪种方式运行，PHP 程序的生命周期都需要经过模块初始化阶段、请求初始化阶段、代码执行阶段、请求结束阶段、模块结束阶段，如图 1。不同运行方式的区别是生命周期中各个阶段执行的频率和次数。

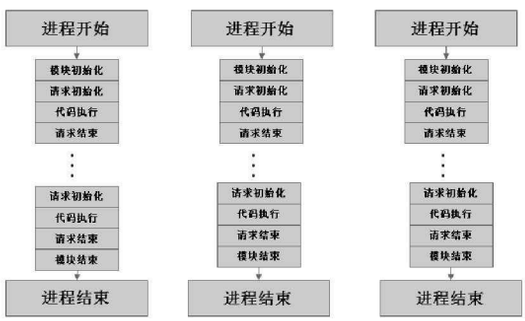

图 1　采用模块加载方式的 PHP 生命周期

1.2　PHP 代码执行流程

PHP 是解释型语言，代码需要被翻译成中间字节码后由 ZEND 引擎进行解析执行。PHP 代码的执行流程主要包括：词法分析、语法解析、代码编译、opcodes（中间字节码）执行四个步骤。词法分析是指将 PHP 代码转换为语言片段，语法分析是将语言片段转换成简单而有意义的表达式，代码编译是指将表达式编译成 opcodes，opcodes 执行是指 zend 虚拟机执行 opcodes 并将结果输出。其详细的执行流程如图 2。

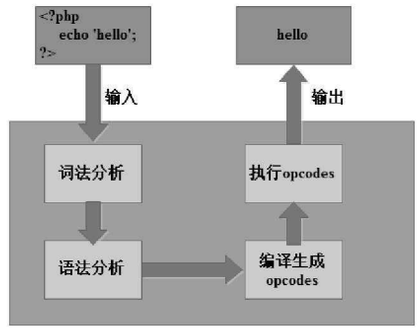

图 2　PHP 代码执行流程

1.3　PHP 内核 hook 机制

对 PHP 函数进行 hook 是指在 PHP 内核中通过函数重写或修改编译函数来达到改变对 PHP 函数运行进行监控以获取函数运行时的参数。PHP 的函数分为两种，一种是 Zend 提供的函数，如 eval；一种是 PHP_FUNCTION 宏编写的函数，如 shell_exec。其中 Zend 提供的函数可以通过修改编译函数 zend_compile_string 的方式来进行 HOOK；PHP_FUNCTION 宏编写的函数可以通过操纵函数表进行重写来进行 hook。除了这两种方法，PHP 内核还提供了通用的 HOOK 方法，即使用 zend_set_user_opcode_handler 修改中间字节码对应的处理函数。

2　Webshell 常见变形

Webshell 本质上是可以执行恶意功能的 PHP 代码文件。Webshell 为了执行恶意功能，其代码结构主要由两部分组成：数据传递部分和数据执行部分，如图 3。数据传递部分是指 webshell 中用于接收外部输入数据的部分，webshell 可以根据外部输入数据来动态交互式执行恶意功能。数据执行部分是指

webshell 中用于执行恶意功能的函数，如代码执行的 eval 函数、命令执行的 system 函数。

图 3　webshell 结构

在基本的 webshell 中，数据传递主要通过$_GET、$_POST、$_COOKIES 等变量传递或者直接写入代码中，数据执行主要通过使用 eval 或 assert 进行代码执行或直接调用功能函数进行执行。为了绕过检测机制，各种 webshell 都在基本 webshell 上采取相应的变形，变形的方法根据其变形的部分主要分为两种：数据传递部分的变形和数据执行部分的变形。

数据传递部分的变形主要有：

将数据放入服务器外部文件中，webshell 读取文件获取执行数据。

（1）将数据放在远程服务器上，通过 curl/file_get_contents 等函数获取远程 URL 中的执行数据。

（2）将数据放入数据库，通过读取数据库获取执行数据。

数据执行部分的变形方法主要有：

（1）使用 preg_replace 函数的/e 修饰符进行代码执行。

（2）使用支持回调机制的函数进行代码回调执行。如：array_map，array_filter，array_reduce 等。

（3）使用变量函数进行函数执行。

（4）使用匿名函数进行函数执行。

（5）利用反射函数 ReflectionFunction 进行函数执行。

3　PHP webshell 实时动态检测

PHP webshell 实时动态检测是一种基于 PHP 扩展通过对 PHP 代码的编译和执行进行监控并结合外部输入变量标记追踪、黑白名单机制来进行 webshell 检测的方法，主要包含五个模块：变量标记追踪、禁用函数 hook 检测、危险函数 hook 检测、编译函数重载检测、数据库黑白名单[4]。

3.1　变量标记追踪

在 PHP 扩展中可以通过 PG（http_globals）变量获取脚本运行时通过 GET、POST、COOKIE 方式传递的参数内容。变量标记是指对 PG（http_globals）里保存的字符串变量进行特征标记。变量追踪是指对简单字符串处理函数如 strval、explode 进行 hook，当函数参数是标记的变量时也对函数结果进行变量标记。

PHP 中的字符串变量的值存储在 zvalue_value 结构体中，保存有字符串指针和字符串的长度，PHP 内核是根据保字符串的长度来读取字符串内容。可以通过将字符串变量的所占内存扩大后，在字符串的值后添加标记特征的方式来将字符串变量进行标记。由于字符串长度没有修改，通过这种方式进行变量标记不会修改字符串的值。在进行变量标记检测时，只要检测字符串指针在长度之后的内容是否是标记特征就可以。

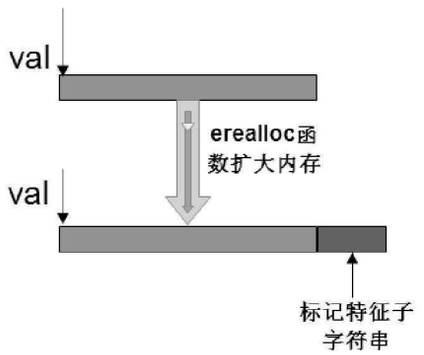

图 4　变量标记

3.2　禁用函数 hook 检测

在 PHP 的配置文件中，disable_functions 参数主要用来设置禁用 PHP 危险函数。禁用函数 hook 检测是指在模块初始化阶段，读取配置文件中 disable_functions 参数后在函数表中添加针对这些函数的自定义实现。在 PHP 代码执行阶段，如果自定义函数被调用，说明 PHP 页面中运行有危险函数。如果该 PHP 脚本不在数据库白名单中，那么可以判定该 PHP 脚本就是 webshell，检测过程如图 5.

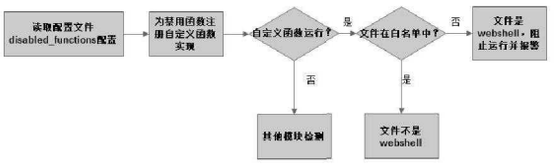

图 5　禁用函数 hook 检测

3.3　危险函数 hook 检测

Webshell 要执行恶意功能，如命令执行函数、文件目录操作，必须要调用相应的功能函数。危险函数 hook 检测就是指对能执行命令、目录操作等危险功能的函数进行 hook 后检测函数执行时的参数，如果危险函数执行的参数是恶意的或者经过变量标记的，且没有出现在白名单中则认为该文件是 webshell 文件，检测过程如图 6。需要进行 hook 的危险函数主要有：

（1）命令执行类：passthru、system、popen、exec、shell_exec 等。

（2）文件系统函数：fopen、opendir、basename、dirname、file、pathinfo、scandir 等

（3）数据库操作函数：mysql_query、mysqli_query、sqlite_query、sqlite_single_query 等。

（4）回调函数类：array_map，array_filter，array_reduce、usort、uksort、array_walk 等。

（5）反射函数：ReflectionFunction

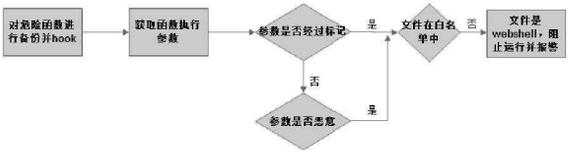

图 6　危险函数 HOOK 检测

3.4　编译函数重载检测

编译函数重载检测是指在模块初始化阶段重载编译函数以检测是否有 eval 或 assert 代码块的执行，并对 eval/assert 代码块的内容进行正则匹配来检测是否有恶意代码。其中 eval 代码块在编译结果的 filename 中有 eval（）'d code 标记，assert 代码块则有 assert code 标记。

检测过程如图 7.

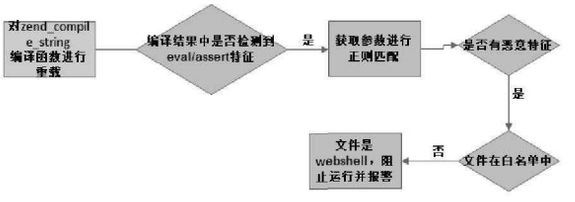

图 7　编译函数重载检测

3.5　数据库黑白名单

为了减少对 webshell 的漏报和误报，采用黑白名单机制。即用户可以添加网站的正常页面进白名单，在进行 webshell 检测时如果在白名单中就跳过检测；用户可以添加某些目录进黑名单，则进行 webshell 检测时如果检测到执行文件在黑名单中则认为是 webshell 并报警处理。

4　实验结果

为了验证该 PHP webshell 实时动态检测模型是否能够有效检测并阻止 webshell，本人收集了较为流行的各种 webshell，

（下转第 125 页）

目录，再分配以适当权限。例如，建立静态文件目录，包括 HTML 所有静态文件，给予允许读取、拒绝写的权限；建立脚本目录，包含 asp 等脚本文件，给予允许执行、拒绝写和读取的权限；建立可执行程序目录，包含所有二进制执行文件，给予允许执行、拒绝写和读取的权限。

对 Windows 系统和 IIS 服务器安全策略控制不当，产生了各种系统级和应用级的漏洞，从而对服务本身产生了诟病。只要管理员正确的部署了 Web 服务的安全性防护措施，一个健壮强大的 IIS 服务器是可以立足于互联网的应用第一线。

参考文献：

[1]杨方，潘大丰. IIS 日志在网站安全中的研究与应用[J]. 农业网络信息. 2010，（10）：96～98.

[2]覃正超. 打造安全的 IIS Web 服务器[J]. 科技资讯 2009,(7):23.

作者简介：

乔晓刚（1975—），男，山西运城，硕士，讲师，主要研究方向：计算机应用技术。

基金项目：山西省教育厅高等学校科技创新项目支持，项目名称：基于“山西消费网”构建校园电子商务创业实践平台，项目编号：20131115。

（上接第 121 页）

包括一句话 webshell 及变形、普通变形 webshell、加密 webshell，与网站安全狗、Avira Antivirus 一起对比查杀，检测结果如图 8.

	一句话 webshell 及变形		普通变形 webshell		加密 webshell	
	检测数	发现数	检测数	发现数	检测数	发现数
网站安全狗	10	10	10	8	10	3
Avira Antivirus	10	5	10	2	10	3
实时 webshell 检测框架	10	10	10	9	10	9

图 8　检测结果

从结果表中可以看到基于 PHP 扩展的 PHP webshell 实时动态检测框架可以无视 PHP 代码是否进行加密，能够优于网站安全狗、Avira Antivirus 来检测并防御大多数的 webshell。只要尽可能的覆盖更多检测函数，可以达到接近 100%的查杀率，缺点是需要消耗较多的系统资源。

5　结束语

本文在分析了 PHP 代码运行的原理和各种 webshell 变形后，提出并实现了一种基于 PHP 扩展的 webshell 实时动态检测方法，该方法能够高效的检测出各种 webshell。在网站中利用该方法进行 webshell 检测与防御可以有效的保证网站的安全运行。

参考文献：

[1]张红瑞. WebShell 原理分析与防范实践[J]. 现代企业教育，2013（20）：254-255.

[2]孟正，梅瑞. Linux 下基于 SVM 分类器的 WebShell 检测方法研究[J].技术研究，2014，5：5-9

[3]胡建康，徐震，马多贺，等. 基于决策树的 Webshell 检测方法研究[J]. 网络新媒体技术，2012，1（6）：15-19.

[4]浅谈从 PHP 内核层面防范 PHP WebShell[EB/OL]，http://sebug.net/，2011.7.24

（上接第 122 页）

ISAKMP、SKIP 这两种。SKIP 在密钥的传输过程中，采用了 Diffie-Hellman 演算法，而 ISAKMP 则采用了公开密钥机制，使通信的双方同时拥有公、私两种密钥。

3　VPN 技术网关

VPN 的网关一般位于组织内部与互联网的连接处，主要用于确保网络安全。网关直接面临着外部网络的威胁，所以必须集成 VPN 网关与防火墙来确保网关安全。网络防火墙可以与 VPN 网关协调，利用防火墙的性能来实现 VPN 网管安全性的提高，从而建立起专用的网关系统，重新优化、编译 linux，形成并加密 VPN 技术隧道，完成 IDS 的集成，搜集并分析关键信息，对于防御黑客非法攻击有实质性作用。另外，安全网关还可以用于管理服务，监控、优化服务质量，做到策略的及时调整，以实现满足用户不同需求的目的。安全漏洞检测也是安全网关的功能之一，对攻击行为起到有效阻隔作用。

4　结语

结合全文的分析，VPN 是一种新兴的互联网通信技术，它体现了当今互联网的发展趋势。VPN 借鉴了传统的网络网络安全技术，对数据共享结构进行了高效率的精简，不但能降低通信成本，还能有效地保证传输通道的安全。相关技术人员在今后的发展中，还应不断研究和探索 VPN 技术在网络安全中的应用，保障互联网通信安全。

参考文献：

[1]吴烈勇.基于 VPN 技术的多校区校园网络安全探讨[J].硅谷，2012（10）.

[2]赵钊.基于 VPN 技术的校园网络安全体系构建[J].自动化与仪器仪表，2014（1）.

[3]沙涛.手机 VPN 技术对网络安全监管挑战分析[J].信息网络安全，2013（6）.

[4]黎伟.VPN 技术在校园网络安全体系中的应用[J].当代教育论坛：学科教育研究，2013（9）.

作者简介：

郑瑞银（1972—），男，江西南丰，硕士研究生，副教授，研究方向：物联网、网络安全。

（上接第 123 页）

合的数据多为单数据；逻辑数据的获取存在相当大的难度，如攻击预定义模型的建立，以及攻击的前提和后续条件的形式化描述都存在很大的难度；逻辑关系不能解释系统中存在的不确定性。

3.2　基于规则推理的融合方法

首先，模糊量化多元多属性信息的不确定性，然后利用利用规则进行逻辑推理，以实现网络安全态势的评估，目前 D-S 证据组合方法和模糊逻辑是研究的热点。D-S 证据组合方法对单元数据的每一个可能决策的支持程度都做了度量，然后寻找一种证据合成规则，接着反复运用合成规则，最终得到整个数据的联合体，对某种决策总的支持程度，从而完成证据融合的过程。基于规则推理的融合方法，并不需要精确了解概率分布情况，但概率很难获得时这种方法最为有效，但是缺点是计算复杂度较高，而且当证据出现冲突时，方法的准确性会受到很大的影响。

3.3　基于概率统计的融合方法

它充分的利用了先验知识的统计特性，根据信息的不确定性，建立态势评估模型，然后通过此模型评估网络安全态势[3]。这种方法的优点是推理过程清晰，易于理解，但是统计的数据过于繁琐，实际工作中会增大工作量，而且运用计算的运算量过于大，影响评估态势的实时性；先验知识和模型提取都存在一定的难度。

4　结语

本文阐述了 D-S 证据理论，可以最大限度的防止关键的态势信息的遗失，同时可以使组合预测的结果对某个单项预测结果不理想的方法不太敏感；另一方面还提高了网络安全态势预测的精度，科学家利用 D-S 证据理论的合成法则，对多个单项预测模型的预测结果进行分析融合，从而弥补单项预测模型带来的预测的局限性。

参考文献：

[1]石波，谢小权.基于 D-S 证据理论的网络安全态势预测方法研究[J].计算机工程与设计，2013，03：821-825.

[2]刘鹏，孟炎，吴艳艳.大规模网络安全态势感知及预测[J].计算机安全，2013，03：28-35.

[3]王莹.网络安全态势预测常用方法探究[J].电子制作，2013，12：135.