

知 ONESstor执行加固脚本IOError: [Errno 13] Permission denied

存储系统 存储配置 刘路 2023-12-31 发表

问题描述

某局点进行端口等保加固，使用ftp的工具进行操作，执行命令后报错，提示IOError: [Errno 13] Permission denied。

```
[SDS_Admin@pts-1 ~]$ ./port_harden.py -a
Traceback (most recent call last):
  File "./port_harden.py", line 16, in <module>
    logging.basicConfig(filename="log_port_harden.log", level=logging.INFO, format=LOG_FORMAT)
  File "/usr/lib64/python2.7/logging/_init_.py", line 1529, in basicConfig
    hdlr = FileHandler(filename, mode)
  File "/usr/lib64/python2.7/logging/_init_.py", line 902, in __init__
    StreamHandler._init__(self, self.open())
  File "/usr/lib64/python2.7/logging/_init_.py", line 925, in open
    stream = open(self.baseFilename, self.mode)
IOError: [Errno 13] Permission denied: '/home/SDS_Admin/port_harden/log_port_harden.log'
```

过程分析

现场的版本为3338版本，该版本已经禁用了root。
ONESstor在禁用root版本，要执行该端口加固操作需要到SDS_Admin账户下执行，现场是在该账户下执行的，这点可以排除。
尝试切换到root账户下执行，看下报错是否会一样，在root下执行会重复让输入密码，即使正确的输入几次后还是会报错。

```
[root@192.200.80.1 ~]$ ./port_harden.py -a
(root@192.200.80.1) Password:
(root@192.200.80.1) Password:
(root@192.200.80.1) Password:
root@192.200.80.1's password:
root@192.200.80.1's password:
Traceback (most recent call last):
  File "./port_harden.py", line 771, in <module>
    cluster_ips = get_cluster_internal_ips(all_nodes_ip, ip_type, cur_user)
  File "./port_harden.py", line 146, in get_cluster_internal_ips
    addr = line.split('=')[1].strip()
IndexError: list index out of range
```

检查加固脚本的日志，提示当前的用户为root，即使切换为SDS_Admin账户也是这打印报错。

```
2023-08-31 09:43:16,061-[INFO]-[run_cmd]: exec cmd <<whoami >> get result: root
, err:
2023-08-31 09:43:16,061-[INFO]-[<module>]: current user is root
2023-08-31 09:43:16,022-[INFO]-[get_need_block_all]: get need block all nodes:所有节点
2023-08-31 09:43:16,023-[INFO]-[<module>]: need to block nodes:['192.200.80.1', '192.200.80.2', '192
2023-08-31 09:43:16,023-[INFO]-[get_ip_type]: get ip type:ipv4
2023-08-31 09:43:43,767-[INFO]-[run_cmd]: exec cmd <<ssh -o ConnectTimeout=10 192.200.80.1 "cat /etc
```

正常情况下，在禁用root版本，该脚本会判断当前账户是否为SDS_Admin账户，如果是该账户就会执行下去。

脚本是通过whoami这个命令来获取账户信息的，实际在现场环境切换账户查询，其都是正常的。

```
[SDS_Admin@pts-1 ~]$ who am i
SDS_Admin pts/1 2023-08-31 09:39 (10.2.91.253)
```

怀疑是该文件的权限有问题，ll /home/SDS_Admin/port_harden 查看文件权限，跟正常环境的对比，发现异常的文件用户为root，正常环境的应该为SDS_Admin。

```
[SDS_Admin@pts-1 ~]$ ll /home/SDS_Admin/port_harden/log_port_harden.log
total 40
-rw-r--r-- 1 root root 6045 Aug 31 10:14 config_port_harden.xlsx
-rwxr-xr-x 1 root root 30130 Aug 31 10:16 port_harden.py
```

异常

```
root@NODE24 ~]# ll /home/SDS_Admin/port_harden/
total 272
-rw-r--r-- 1 SDS_Admin root 9442 Nov 1 14:50 config_port_harden.xlsx
-rw-r--r-- 1 SDS_Admin root 232555 Nov 1 14:47 log_port_harden.log
-rwxr-xr-x 1 SDS_Admin root 30130 Nov 16 16:34 port_harden.py
```

正常

解决方法

如上分析，将该文件的用户名改回SDS_Admin后再次执行可以正常运行。