AWD线下赛防守脚本

author：说书人

说明

1.该脚本基于python，可直接在linux靶机上运行。
2.开局直接运行起来，会自动对web目录进行备份，并建立hash索引。当web目录下有文件被删除或者被篡改的时候，会自动从备份中恢复文件。如果存在其他文件上传，会自动删除。
3.无法避免的缺点：由于条件竞争，如果对方在我们删除shell之前就已经在内存中开始生成不死马了，还是有一定几率沦陷。

```
[ctf@localhost www]$ python ssr.py
Tue Jun  8 04:13:29 2021   安全
Tue Jun  8 04:13:31 2021   安全
Tue Jun  8 04:13:32 2021   安全
Tue Jun  8 04:13:33 2021   安全
Tue Jun  8 04:13:35 2021   安全
Tue Jun  8 04:13:36 2021   安全
Tue Jun  8 04:13:37 2021   安全
Tue Jun  8 04:13:39 2021   安全
Tue Jun  8 04:13:40 2021   安全
[delete]webshell:/var/www/1.php
Tue Jun  8 04:13:41 2021   安全
Tue Jun  8 04:13:43 2021   安全
Tue Jun  8 04:13:44 2021   安全
Tue Jun  8 04:13:45 2021   安全
Tue Jun  8 04:13:47 2021   安全
Tue Jun  8 04:13:48 2021   安全
Tue Jun  8 04:13:49 2021   安全
[delete]webshell:/var/www/.main.php.swp
Tue Jun  8 04:13:51 2021   安全
Tue Jun  8 04:13:52 2021   安全
[modify-recover]file:/var/www/main.php
Tue Jun  8 04:13:53 2021   安全
Tue Jun  8 04:13:55 2021   安全
Tue Jun  8 04:13:56 2021   安全
Tue Jun  8 04:13:57 2021   安全
```

代码

```python
# -*- coding: utf-8 -*-#
# awd文件监控脚本
# author: 说书人
import os
import json
import time
import hashlib


def ListDir(path):  # 获取网站所有文件

    for file in os.listdir(path):
        file_path = os.path.join(path, file)
        if os.path.isdir(file_path):
            if initialization['ok'] == 'false':
                dir_list.append(file_path)
            else:
                dir_list_tmp.append(file_path)
            ListDir(file_path)
        else:
            if initialization['ok'] == 'false':
                file_list.append(file_path)
            else:
                file_list_tmp.append(file_path)


def GetHash():  # 获取hash，建立索引
    for bak in file_list:
        with open(bak, 'rb') as f:
            md5obj = hashlib.md5()
            md5obj.update(f.read())
        hash = md5obj.hexdigest()
        bak_dict[bak] = hash
    if os.path.exists('/tmp/awd_web_hash.txt') == False:
        os.system('mkdir /tmp/awd_web_bak/')
        os.system('\\cp -a {0}* /tmp/awd_web_bak/'.format(web_dir))
        with open('/tmp/awd_web_hash.txt', 'w') as f:  # 记录web文件hash
```

```
            f.write(str(json.dumps(bak_dict)))
        for i in file_list:  # 记录web文件列表
            with open('/tmp/awd_web_list.txt', 'a') as f:
                f.write(i + '\n')
        for i in dir_list:  # 记录web目录列表
            with open('/tmp/awd_web_dir.txt', 'a') as f:
                f.write(i + '\n')


def FileMonitor():  # 文件监控
    # 提取当前web目录状态
    initialization['ok'] = 'true'
    for file in os.listdir(web_dir):
        file_path = os.path.join(web_dir, file)
        if os.path.isdir(file_path):
            dir_list_tmp.append(file_path)
            ListDir(file_path)
        else:
            file_list_tmp.append(file_path)
    for file in file_list_tmp:
        with open(file, 'rb') as f:
            md5obj = hashlib.md5()
            md5obj.update(f.read())
        hash = md5obj.hexdigest()
        bak_dict_tmp[file] = hash
    with open('/tmp/awd_web_hash.txt', 'r') as f:  # 读取备份的文件hash
        real_bak_dict = json.loads(f.read())
    with open('/tmp/awd_web_list.txt', 'r') as f:  # 读取备份的文件列表
        real_file_list = f.read().split('\n')[0:-1]
    with open('/tmp/awd_web_dir.txt', 'r') as f:  # 读取备份的目录列表
        real_dir_list = f.read().split('\n')[0:-1]

    for dir in real_dir_list:  # 恢复web目录
        try:
            os.makedirs(dir)
            print("[del-recover]dir:{}".format(dir))
        except:
            pass

    for file in file_list_tmp:
        try:
            if real_bak_dict[file] != bak_dict_tmp[file]:  # 检测被篡改的文件，自动恢复
                os.system('\\cp {0} {1}'.format(file.replace(web_dir, '/tmp/awd_web_bak/'), file))
                print("[modify-recover]file:{}".format(file))
        except:  # 检测新增的文件，自动删除
            os.system('rm -rf {0}'.format(file))
            print("[delete]webshell:{0}".format(file))

    for real_file in real_file_list:  # 检测被删除的文件，自动恢复
        if real_file not in file_list_tmp:
            os.system('\\cp {0} {1}'.format(real_file.replace(web_dir, '/tmp/awd_web_bak/'), real_file))
            print("[del-recover]file:{0}".format(real_file))
    file_list_tmp[:] = []
    dir_list_tmp[:] = []


os.system("rm -rf /tmp/awd_web_hash.txt /tmp/awd_web_list.txt /tmp/awd_web_dir.txt /tmp/awd_web_bak/")
web_dir = "/var/www/"  # web目录，注意最后要加斜杠
file_list = []
dir_list = []
bak_dict = {}
file_list_tmp = []
dir_list_tmp = []
bak_dict_tmp = {}
initialization = {'ok': 'false'}
ListDir(web_dir)
GetHash()
while True:
    print(time.ctime()+"   安全")
    FileMonitor()
    time.sleep(1)  # 监控间隔，按需修改

```