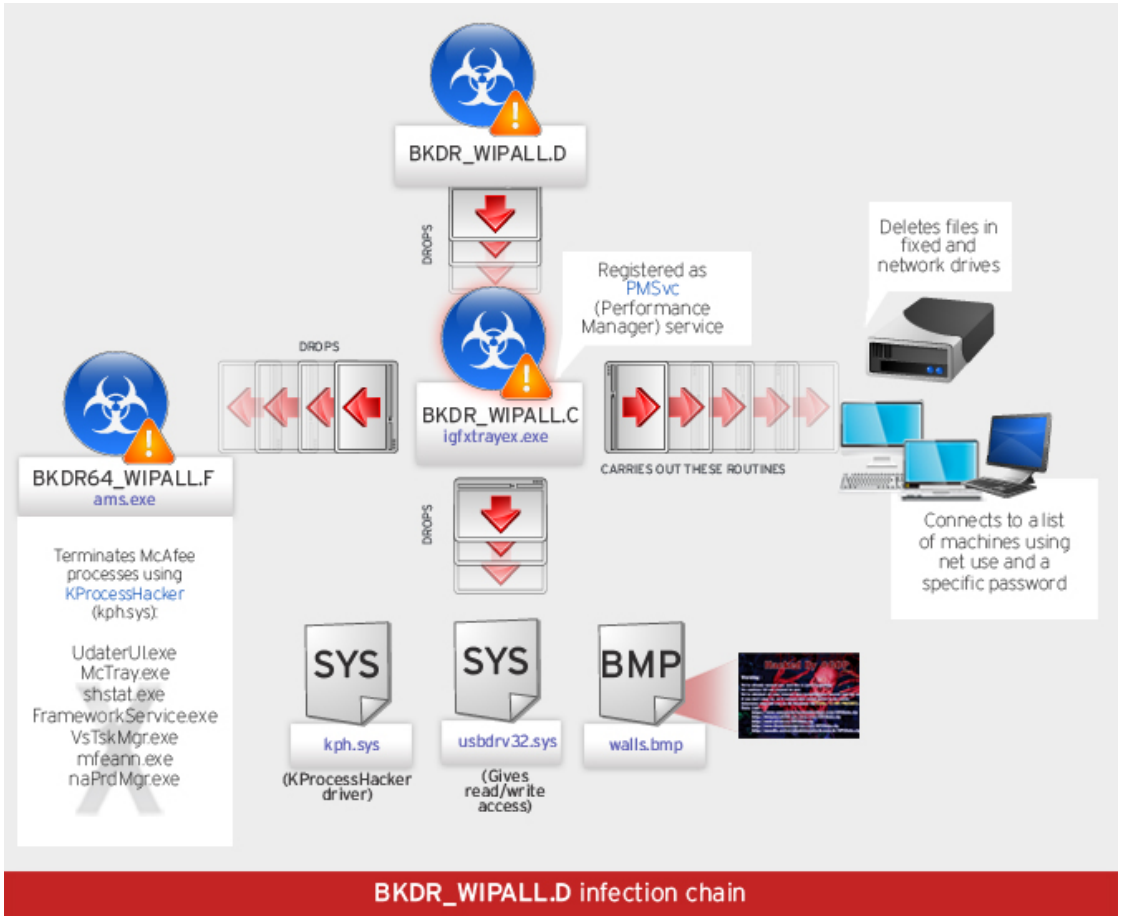
WIPALL Malware Leads to #GOP Warning in Sony

By Trend Micro

Our previous blog entry discussed the “destructive” FBI security advisory and an analysis about the WIPALL malware family and its direct connection to the massive Sony Pictures hack. In this blog post, we will further discuss other WIPALL malware variants and their main routines that link to the #GOP warning seen in infected computers of Sony Pictures employees. Below is an overview of the infection chain to be discussed in this entry:



BKDR_WIPALL.D infection chain

BKDR64_WIPALL.F Disables McAfee’s Services

The WIPALL variant BKDR_WIPALL.C shares the same coding as the previously discussed variant, BKDR_WIPALL.B. In the case of BKDR_WIPALL.C, the dropped copies are named as *igfxtrays{2 random characters}.exe* and executes several copies of itself with specific parameters (*-a, -m, -d, -s*), which contain its main routines.

Pseudocode-I

```
if ( v0 == 107 )
{
  Sleep(0x2932E0u);
  Dest = 0;
  memset(&v10, 0, 0x204u);
  v11 = 0;
  wcscpy(&Dest, L"-a");
  sub_4033A0(&Dest);
  Sleep(0x1388u);
  wcscpy(&Dest, L"-m");
  sub_4033A0(&Dest);
  wcscpy(&Dest, L"-d");
  sub_4033A0(&Dest);
  wcscpy(&Dest, L"-s");
  sub_4033A0(&Dest);
  v1 = CreateThread(0, 0, StartAddress, 0, 0, 0);
  WaitForSingleObject(v1, 0xFFFFFFFFu);
  CloseHandle(v1);
  Sleep(0xBB8u);
  WSAStartup(0x202u, &WSAData);
  memset(&unk_415D60, 0, 0x28u);
  sub_402DD0(&unk_415D60);
  dword_415D84 = 4;
  result = sub_402D10();
}
else
{
  result = v0 - 97;
  switch ( result )
  {
    case 0:
      result = sub_401500();
      break;
    case 12:
      result = sub_401A90();
      break;
    case 3:
      v8 = 0;
      v7 = 0;
      v6 = (void *)1;
      v5 = (int (__stdcall *)(int))sub_4028A0;
      goto LABEL_9;
    case 13:
      v3 = CreateThread(0, 0, sub_4028A0, (LPVOID)2, 0, 0);
      Sleep(0x493E0u);
      sub_402E20();
      WaitForSingleObject(v3, 0xFFFFFFFFu);
      result = CloseHandle(v3);
      dword_415D84 = 0;
      break;
    case 18:
      v8 = 0;
      v7 = 0;
      v6 = 0;
      v5 = sub_401730;
```

Figure 1. Main malware routines of BKDR_WIPALL.C

It is a notable observation that BKDR_WIPALL.C checks if the infected system is 64-bit. If found to be running on a 64-bit system, the malware drops *kph.sys* (KProcessHacker driver) and its component *ams.exe* (detected as BKDR64_WIPALL.F).

We noticed that BKDR64_WIPALL.F replaces McAfee's real-time scanner, *mcshield.exe* with another file located in its current directory, while the original *mcshield.exe* is placed in the system32 directory. In turn, when McAfee's service executes, the replacement file will be executed instead of the legitimate real-time scanner component, effectively disabling the antivirus' operation.

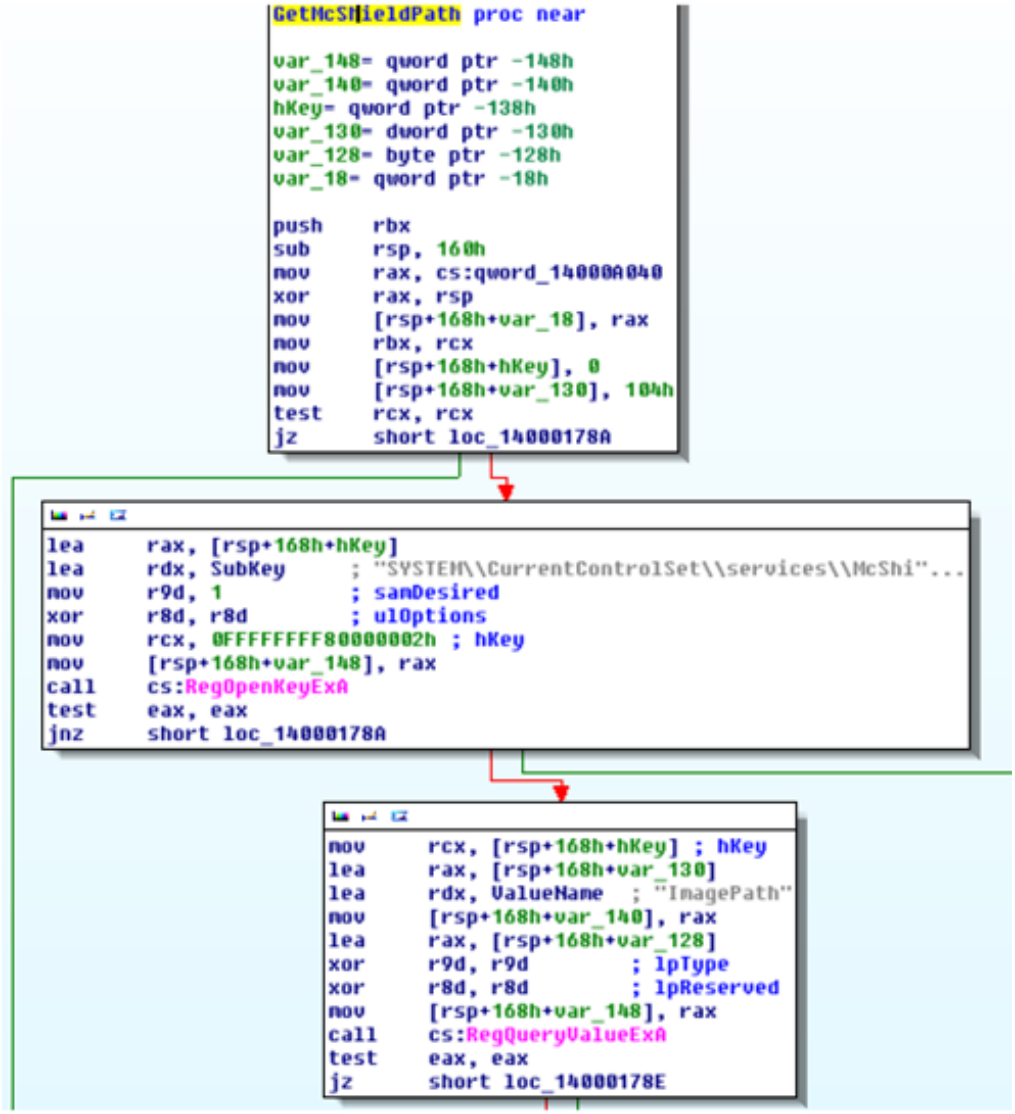


Figure 2. BKDR64_WIPALL.F obtains the Image Path of McShield.exe *from the registry's list of services:* HKLM\CurrentControlSet\services\McShield

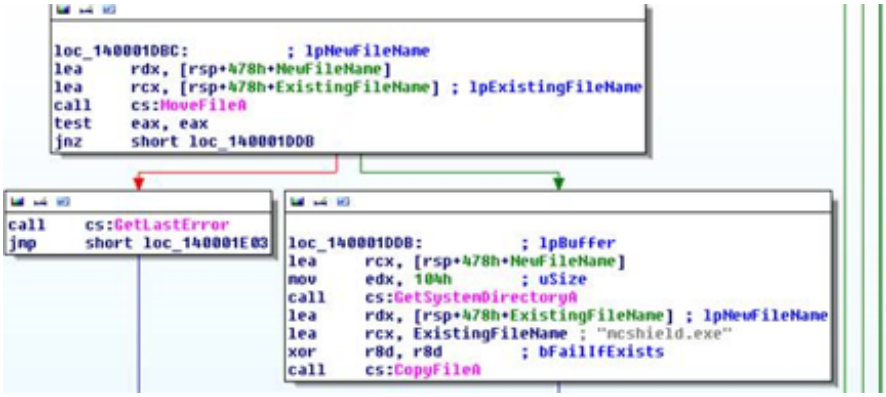


Figure 3. BKDR64_WIPALL.F *moves the legitimate* mcshield.exe *to the System32 folder and replaces it with another* mcshield.exe *located in the malware's current directory*

BKDR64_WIPALL.F installs *KprocessHacker* as a driver service and uses it to terminate the following running processes related to McAfee's antivirus application (also listed in the infection chain above). This is an added measure in order to ensure the malware's smooth execution.

- *mcshield.exe*

- *UdaterUI.exe*
- *McTray.exe*
- *shstat.exe*
- *FrameworkService.exe*
- *VsTskMgr.exe*
- *mfeann.exe*
- *naPrdMgr.exe*

Based on our analysis, the malware BKDR64_WIPALL.F may have used a driver service because it has a higher privilege than a typical user-mode application. This is to ensure that the processes will be terminated.

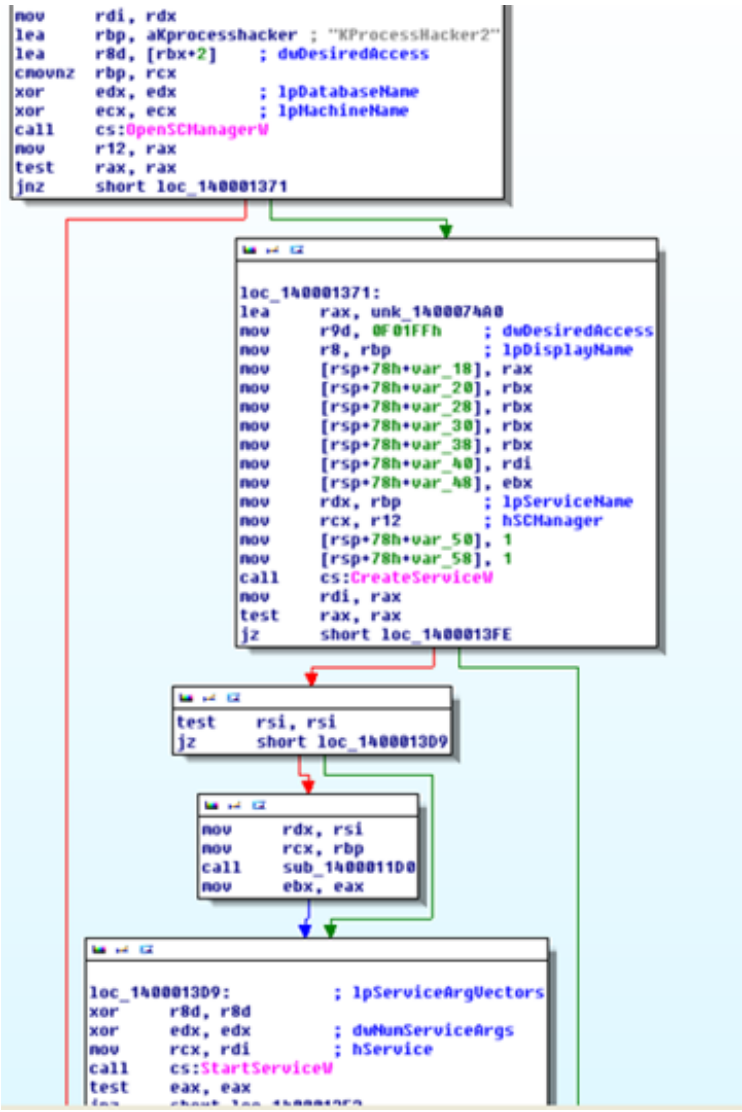


*Figure 4. BKDR64_WIPALL.F installs the KProcessHacker component (*kph.sys*) as a service driver*

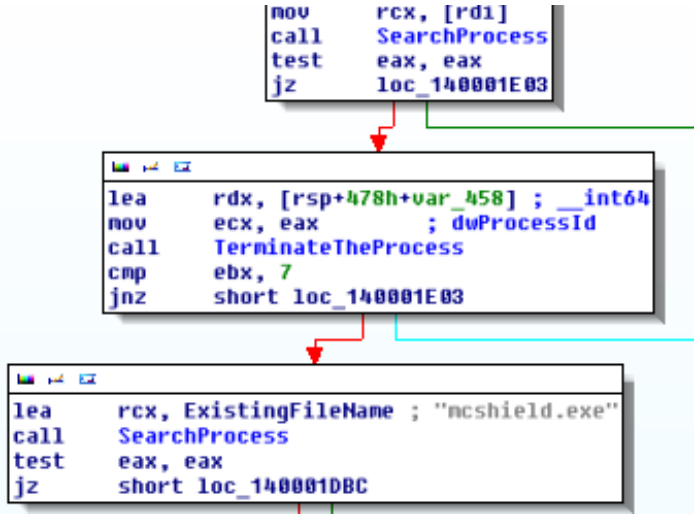


Figure 5. BKDR64_WIPALL.F checks all running processes with the hardcoded list of processes related to McAfee antivirus applications

```
mov     rcx, cs:qword_14000AF60
lea     rax, [rsp+68h+var_18]
xor     r8d, r8d
xor     edx, edx
mov     [rsp+68h+var_48], rax
call    cs:qword_14000AF58 ; ZwDeviceIoControlFile
add     rsp, 68h
retn
```

Figure 6. It uses the KprocessHacker service driver as a device object to terminate the processes

Tracing Back to #GOP

This attack, along with the one we discussed in our previous blog entry, were both found to trace back to the hacker group named #GOP or “Guardians of Peace.”

The BKDR_WIPALL.A infection chain (via its component BKDR_WIPALL.E) leads to an HTML file displaying the message with the files *back.jpg* and *index.wav.* All of these are encrypted and embedded in the component *iissvr.exe* (detected as BKDR_WIPALL.E).

Similarly, the infection chain for BKDR_WIPALL.D (via its component BKDR_WIPALL.C) displays the #GOP message in an image file dropped as *walls.bmp*.

Figure 7: ***Top****:* walls.bmp *dropped by BKDR_WIPALL.C;*
Bottom*: Scrolling message in an HTML file loaded by BKDR_WIPALL.E*

There have been reports linking these attacks to North Korea as the culprit, and some claim that the Sony hack may have been an inside job. While nothing is confirmed at the moment, we advise users to exercise vigilance in their online to ensure private data stays that way.

Read our timeline of events related to the Sony hack in our page: The Hack of Sony Pictures: What We Know and What You Need to Know.

Analysis by Rhena Inocencio and Joie Salvio

Related hashes:

- *D1C27EE7CE18675974EDF42D4EEA25C6 as BKDR_WIPALL.A*
- *760C35A80D758F032D02CF4DB12D3E55 as BKDR_WIPALL.B*
- *E1864A55D5CCB76AF4BF7A0AE16279BA as BKDR_WIPALL.E*
- *B80AA583591EAF758FD95AB4EA7AFE39 as BKDR_WIPALL.C*
- *2618dd3e5c59ca851f03df12c0cab3b8 as BKDR_WIPALL.D*
- *7E5FEE143FB44FDB0D24A1D32B2BD4BB as BKDR64_WIPALL.F*

This entry was posted on Friday, December 5th, 2014 at 4:47 pm and is filed under Bad Sites, Malware . Both comments and pings are currently closed.