VERSIONS OF PSIXBOT

blog.comodo.com/comodo-news/versions-of-psixbot/

March 11, 2020

Comodo

03.11.2020

3 years ago

Comodo News

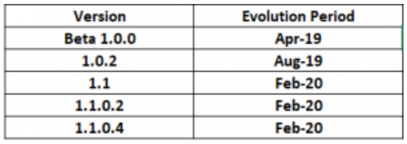
★★★★☆ (**7** votes, **4.71** / 5)
Loading...

Reading Time: 4 minutes

Introduction of PSIXBOT:

PsiXBot is data-stealing trojan capable of harvesting confidential data and passwords from a victim's computer. It can steal cookies, extract logins/passwords from applications like Firefox and Microsoft Outlook, record the victim's keystrokes, allow criminals to remotely view/interact with the victim's desktop, and can even add the victim's computer to a botnet. It is most often spread via infected email attachments, via online adverts which contain the bot, and via other social engineering methods.

The original PsixBot malware surfaced in November 2017 but underwent significant development before arriving in beta format in 2019. It has since been developed further and currently stands at version 1.1.0.4 in February 2020:



Version	Evolution Period
Beta 1.0.0	Apr-19
1.0.2	Aug-19
1.1	Feb-20
1.1.0.2	Feb-20
1.1.0.4	Feb-20

PsixBot was generated in .NET framework. This blog takes you through the various iterations of PsixBot to illustrate how online criminals constantly update their malware to improve its performance and features.

Behaviour of PsixBot

PsixBot changes the system certificate settings, which gives it virtually unlimited user access rights on the host machine:

Keys added:

KEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\TrustedPeople\Certificates\636D2838EB7A7F3A8E6B6F7CD035375E7E7042

Values added:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\TrustedPeople\Certificates\636D2838EB7A7F3A8E6B6F7CD035375E7E70
02 00 00 ……..

Files added:

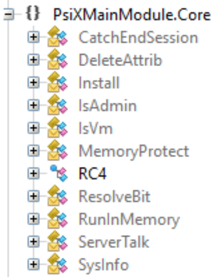
C:\Documents and Settings\Administrator\Application Data\

Microsoft\SystemCertificates\My\Certificates\636D2838EB7A7F3A8E6B6F7CD035375E7E704248

Beta 1.0.0

The first version of PsixBot covered in this blog is Beta 1.0.0 with the core class 11. Each class has its individual task. The following basic classes are used in all versions of PsixBot:

- **Servertalk** – used to initialize the global variable, create the connection with the mothership server, and send results back and forth.
- **RunInMemory** – used to actually execute the file.
- **SysInfo** – used to obtain information about the user's system, including antivirus name, CPU, Windows version, user type and user permissions.
- **CatchEndSession** – used to create hidden autoruns.
- **DeleteAttrib** – used to kill the system's antivirus software, Windows Explorer, and any system error alerts.
- **IsAdmin** – used to assume membership of the admin group.
- **IsVm** – detects the presence of any virtual machines.
- **ResolveBit** – used to resolve DNS requests from the user.
- **RC4** – the algorithm used to encrypt and decrypt data.
- **Install** – installs the bot file and sets up the file's security and update modules.



```
{} PsiXMainModule.Core
    CatchEndSession
    DeleteAttrib
    Install
    IsAdmin
    IsVm
    MemoryProtect
    RC4
    ResolveBit
    RunInMemory
    ServerTalk
    SysInfo
```

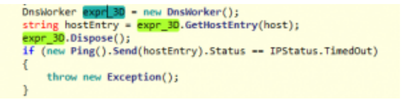
Version 1.0.2

Beta 1.0.2 retained the basic class functionality of the first version, but renamed some of the classes as follows:

- **ServerTalk –** renamed as **CpWorker**
- **RunInMemory –** renamed as **MemoryModulesWorker**
- **SysInfo –** renamed as **SysHelper**

… and added the following class:

DNSWorker – used to get the host entry and ping the host to check whether or not it is up.

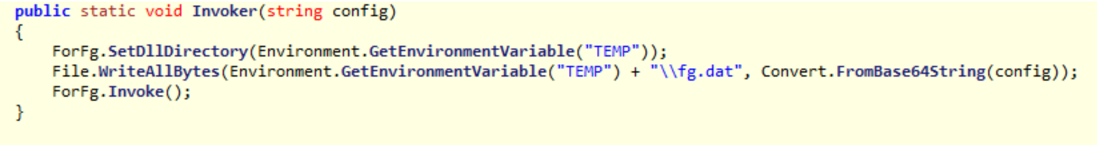


```
DnsWorker expr_3D = new DnsWorker();
string hostEntry = expr_3D.GetHostEntry(host);
expr_3D.Dispose();
if (new Ping().Send(hostEntry).Status == IPStatus.TimedOut)
{
    throw new Exception();
}
```

Version 1.1

Version 1.1 again retained the same class structure as its predecessor but added the following task to the features list:

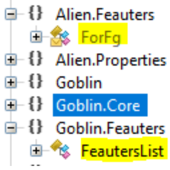
Forfg – used to obtain the path to the temp variable, set the DLL directory and write it to a .dat file:

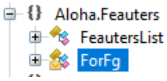


```
public static void Invoker(string config)
{
    ForFg.SetDllDirectory(Environment.GetEnvironmentVariable("TEMP"));
    File.WriteAllBytes(Environment.GetEnvironmentVariable("TEMP") + "\\fg.dat", Convert.FromBase64String(config));
    ForFg.Invoke();
}
```

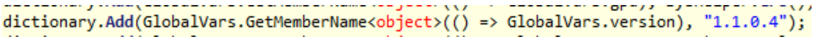
Version 1.1.0.2

Version 1.1.0.2 saw an update whereby the **FORFG** feature was combined with the other feature list. All other classes and activities remained the same.





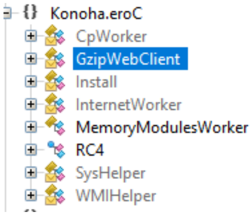
Version 1.1.0.4

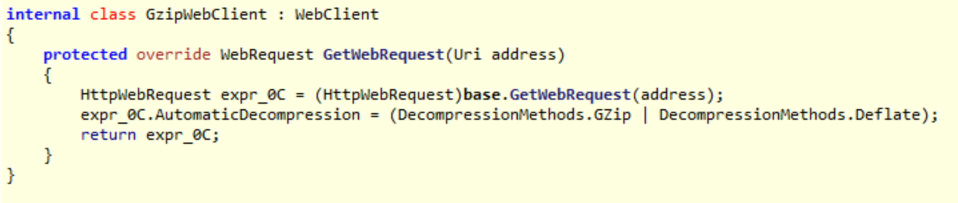


```
dictionary.Add(GlobalVars.GetMemberName<object>(() => GlobalVars.version), "1.1.0.4");
```

Again, the basic classes remained the same as the previous version but with the addition of the following, important, class

GzipWebClient – used to decompress any Gzip files downloaded by the bot:

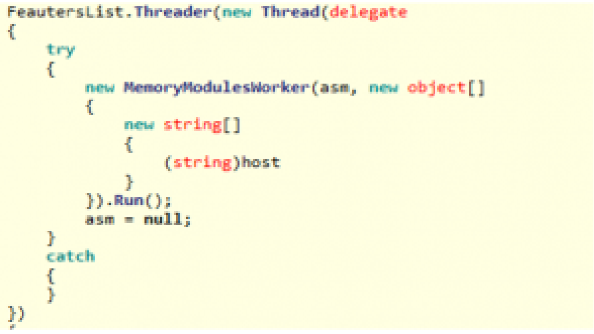




```
internal class GzipWebClient : WebClient
{
    protected override WebRequest GetWebRequest(Uri address)
    {
        HttpWebRequest expr_0C = (HttpWebRequest)base.GetWebRequest(address);
        expr_0C.AutomaticDecompression = (DecompressionMethods.GZip | DecompressionMethods.Deflate);
        return expr_0C;
    }
}
```

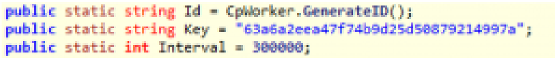
Feature List Updates

Threader – Invoke the thread function used to run the file and run it it memory (**RunInMemory**).



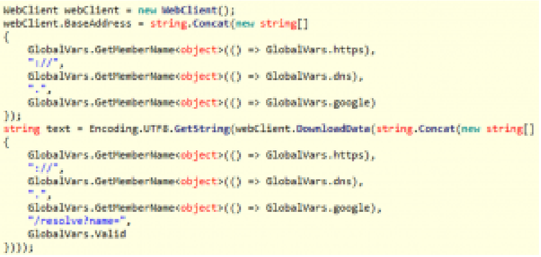
```
FeautersList.Threader(new Thread(delegate
{
    try
    {
        new MemoryModulesWorker(asm, new object[]
        {
            new string[]
            {
                (string)host
            }
        }).Run();
        asm = null;
    }
    catch
    {
    }
})
```

Bot Key – PsixBot has a common, hard-coded key in all versions:



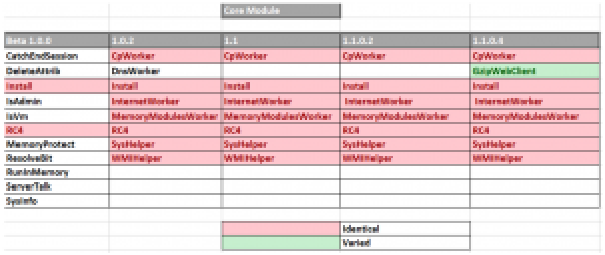
```
public static string Id = CpWorker.GenerateID();
public static string Key = "63a6a2eea47f74b9d25d50879214997a";
public static int Interval = 300000;
```

Network Activities– PsixBot initially uses Google DNS then later communicates with its own DNS:



```
WebClient webClient = new WebClient();
webClient.BaseAddress = string.Concat(new string[]
{
    GlobalVars.GetMemberName<object>(() => GlobalVars.https),
    "://",
    GlobalVars.GetMemberName<object>(() => GlobalVars.dns),
    ".",
    GlobalVars.GetMemberName<object>(() => GlobalVars.google)
});
string text = Encoding.UTF8.GetString(webClient.DownloadData(string.Concat(new string[]
{
    GlobalVars.GetMemberName<object>(() => GlobalVars.https),
    "://",
    GlobalVars.GetMemberName<object>(() => GlobalVars.dns),
    ".",
    GlobalVars.GetMemberName<object>(() => GlobalVars.google),
    "/resolve?name=",
    GlobalVars.Valid
})));
```

Core Modules per Version

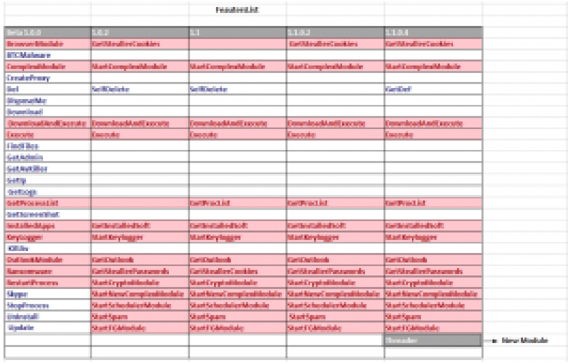


Core Module

Beta 1.0.0	1.0.2	1.1	1.1.0.2	1.1.0.4
CatchEndSession	CpWorker	CpWorker	CpWorker	CpWorker
DeleteAttrib	DnsWorker			GzipWebClient
Install	Install	Install	Install	Install
IsAdmin	InternetWorker	InternetWorker	InternetWorker	InternetWorker
IsVm	MemoryModulesWorker	MemoryModulesWorker	MemoryModulesWorker	MemoryModulesWorker
RC4	RC4	RC4	RC4	RC4
MemoryProtect	SysHelper	SysHelper	SysHelper	SysHelper
ResolveBit	WMIHelper	WMIHelper	WMIHelper	WMIHelper
RunInMemory				
ServerTalk				
SysInfo				

🟥	Identical
🟩	Varied

FeautersList per Version



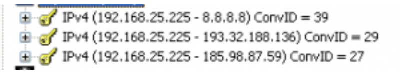
FeautersList

Beta 1.0.0	1.0.2	1.1	1.1.0.2	1.1.0.4
BrowserModule	GetMozillaCookies		GetMozillaCookies	GetMozillaCookies
BTCMalware				
CompiledModule	StartCompiledModule	StartCompiledModule	StartCompiledModule	StartCompiledModule
CreateProxy				
Del	SelfDelete	SelfDelete		GetDef
DisposeMe				
Download				
DownloadAndExecute	DownloadAndExecute	DownloadAndExecute	DownloadAndExecute	DownloadAndExecute
Execute	Execute	Execute	Execute	Execute
FindFiles				
GetAdmin				
GetAvKiller				
GetIp				
GetLogs				
GetProcessList		GetProcList	GetProcList	GetProcList
GetScreenShot				
InstalledApps	GetInstalledSoft	GetInstalledSoft	GetInstalledSoft	GetInstalledSoft
KeyLogger	StartKeylogger	StartKeylogger	StartKeylogger	StartKeylogger
KillAv				
OutlookModule	GetOutlook	GetOutlook	GetOutlook	GetOutlook
Ransomware	GetMozillaPasswords	GetMozillaCookies	GetMozillaPasswords	GetMozillaPasswords
RestartProcess	StartCryptoModule	StartCryptoModule	StartCryptoModule	StartCryptoModule
Skype	StartNewCompiledModule	StartNewCompiledModule	StartNewCompiledModule	StartNewCompiledModule
StopProcess	StartSchedulerModule	StartSchedulerModule	StartSchedulerModule	StartSchedulerModule
Uninstall	StartSpam	StartSpam	StartSpam	StartSpam
Update	StartTORModule	StartTORModule	StartTORModule	StartTORModule
				[illegible]

New Module

Network Traffic

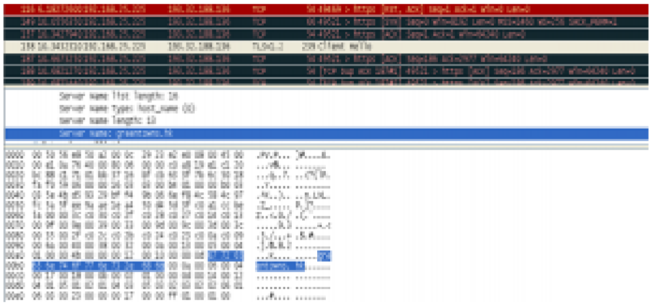
PsixBot initially connects to Google DNS then connects to its own DNS server at **greentowns.hk**:



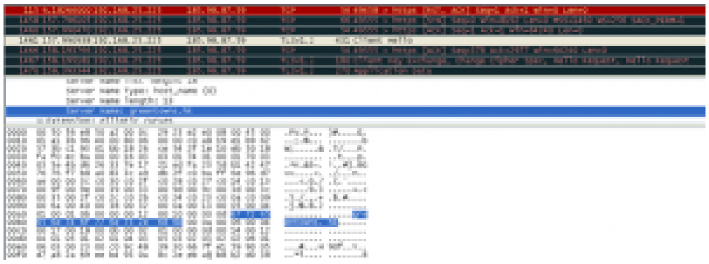
IPv4 (192.168.25.225 - 8.8.8.8) ConvID = 39
IPv4 (192.168.25.225 - 193.32.188.136) ConvID = 29
IPv4 (192.168.25.225 - 185.98.87.59) ConvID = 27

193.32.188.136 (greentowns.hk)

185.98.87.59 (greentowns.hk)



```
Server name list length: 15
Server name type: host_name (0)
Server name length: 13
Server name: greentowns.hk
```



IOC

a85e280e24099a2ffb5ea6efbe3fcb6fbc0c8cfa	**09-04-2019**	**Beta 1.0.0**
0956cec17f1a8801042b8e6628f54e3156d05918	**26-08-2019**	**1.0.2**
4d3b1bd14ca92609fa8d1a536d814fd0d54c5666	**03-02-2020**	**1.1**
a16c7263a36a235db8c71477be3f2442a8a5f894	**04-02-2020**	**1.1.0.2**
1e29be939667354a8fe9477179c6851622118e23	**12-02-2020**	**1.1.0.4**

193.32.188.136(greentowns.hk)

185.98.87.59(greentowns.hk)