

Forum:[Hack Forums](#)

Thread: [improve Anti-VM Anti-SandBox code](#)

xaHyeG - January 28, 2014, 12:28 pm

I just wrote this class, It contains methods I grabbed from c++ sources from google.

Who can help me improve the code?

Edit: changed some UPPERCASE strings

Code public class Detection

```
{
[DllImport("advapi32.dll", SetLastError = true)]
public static extern bool GetUserName(StringBuilder sb, ref Int32 length);

[DllImport("kernel32.dll")]
public static extern IntPtr GetModuleHandle(string lpModuleName);

[DllImport("user32.dll", SetLastError = true)]
static extern IntPtr FindWindow(string lpClassName, IntPtr ZeroOnly);

[DllImport("kernel32.dll")]
extern public static IntPtr GetProcAddress(IntPtr hModule, string procedureName);

[DllImport("kernel32.dll", CharSet = CharSet.Auto, SetLastError = true)]
public static extern uint GetFileAttributes(string lpFileName);

public static bool IsSandbox(string startupPath)
{
    StringBuilder username = new StringBuilder();
    Int32 nSize = 50;
    GetUserName(username, ref nSize);

    // Detected sandboxie
    if ((int)GetModuleHandle("SbieDll.dll") != 0) { return true; }

    // unknown sandbox
    if (username.ToString().ToUpper() == "USER") { return true; }
    // unknown sandbox
    if (username.ToString().ToUpper() == "SANDBOX") { return true; }
    // unknown sandbox
    if (username.ToString().ToUpper() == "VIRUS") { return true; }
    // unknown sandbox
    if (username.ToString().ToUpper() == "MALWARE") { return true; }

    // Detected CW Sandbox.
    if (username.ToString().ToUpper() == "SCHMIDTI") { return true; }

    // Detected Norman Sandbox.
```

```

if (username.ToString().ToUpper() == "CURRENTUSER") { return true; }

// unknown sandbox
if (startupPath.ToUpper().Contains("\\VIRUS")) { return true; }

// unknown sandbox
if (startupPath.ToUpper().Contains("SANDBOX")) { return true; }

// Detected Anubis sandbox. any body can call there user andy :s
if (startupPath.ToUpper().Contains("SAMPLE") ){ return true; }
// this one will false detect pc with name andy // if(username.ToString().ToUpper() ==
"andy".ToUpper()) { return true; }

// Detected Sunbelt sandbox.
if(startupPath == "C:\\file.exe"){ return true; }

// Detected WinJail Sandbox.
if((int)FindWindow("Afx:400000:0", (IntPtr)0) != 0){ return true; }

return false;

}

public static bool IsVM()
{

// Detected vbox
if (regGet("HARDWARE\\DEVICEMAP\\Scsi\\Scsi Port 0\\Scsi Bus 0\\Target Id 0\\Logical Unit
Id 0", "Identifier").ToUpper().Contains("VBOX")) { return true; }
if (regGet("HARDWARE\\Description\\System",
"SystemBiosVersion").ToUpper().Contains("VBOX")) { return true; }
if (regGet("HARDWARE\\Description\\System",
"VideoBiosVersion").ToUpper().Contains("VIRTUALBOX")) { return true; }
if (regGet("SOFTWARE\\Oracle\\VirtualBox Guest Additions", "") == "noValueButYesKey") {
return true; }
if (GetFileAttributes("C:\\WINDOWS\\system32\\drivers\\VBoxMouse.sys") != (uint)4294967295)
{ return true; }

// Detected vmware
if (regGet("HARDWARE\\DEVICEMAP\\Scsi\\Scsi Port 0\\Scsi Bus 0\\Target Id 0\\Logical Unit
Id 0", "Identifier").ToUpper().Contains("VMWARE")) { return true; }
if (regGet("SOFTWARE\\VMware, Inc.\\VMware Tools", "") == "noValueButYesKey") { return
true; }
if (regGet("HARDWARE\\DEVICEMAP\\Scsi\\Scsi Port 1\\Scsi Bus 0\\Target Id 0\\Logical Unit
Id 0", "Identifier").ToUpper().Contains("VMWARE")) { return true; }
if (regGet("HARDWARE\\DEVICEMAP\\Scsi\\Scsi Port 2\\Scsi Bus 0\\Target Id 0\\Logical Unit
Id 0", "Identifier").ToUpper().Contains("VMWARE")) { return true; }
if (regGet("SYSTEM\\ControlSet001\\Services\\Disk\\Enum",
"0").ToUpper().Contains("vmware".ToUpper())) { return true; }
if
(regGet("SYSTEM\\ControlSet001\\Control\\Class\\{4D36E968-E325-11CE-BFC1-08002BE10318}\\
0000", "DriverDesc").ToUpper().Contains("VMWARE")) { return true; }
if

```

```

(regGet("SYSTEM\\ControlSet001\\Control\\Class\\{4D36E968-E325-11CE-BFC1-08002BE10318}\\
0000\\Settings", "Device Description").ToUpper().Contains("VMWARE")) { return true; }
    if (regGet("SOFTWARE\\VMware, Inc.\\VMware Tools",
"InstallPath").ToUpper().Contains("C:\\PROGRAM FILES\\VMWARE\\VMWARE TOOLS\\")) { return
true; }
    if (GetFileAttributes("C:\\WINDOWS\\system32\\drivers\\vmmouse.sys") != (uint)4294967295) {
return true; }
    if (GetFileAttributes("C:\\WINDOWS\\system32\\drivers\\vmhgfs.sys") != (uint)4294967295) {
return true; }

    // Detected whine
    if (GetProcAddress((IntPtr)GetModuleHandle("kernel32.dll"), "wine_get_unix_file_name") !=
(IntPtr)0) { return true; }

    // Detected QEMU
    if (regGet("HARDWARE\\DEVICEMAP\\Scsi\\Scsi Port 0\\Scsi Bus 0\\Target Id 0\\Logical Unit
Id 0", "Identifier").ToUpper().Contains("QEMU")) { return true; }
    if (regGet("HARDWARE\\Description\\System",
"SystemBiosVersion").ToUpper().Contains("QEMU")) { return true; }

    // some extra
    ManagementScope scope = new ManagementScope("\\\\.\\ROOT\\cimv2");
    ObjectQuery query = new ObjectQuery("SELECT * FROM Win32_VideoController");
    ManagementObjectSearcher searcher = new ManagementObjectSearcher(scope, query);
    ManagementObjectCollection queryCollection = searcher.Get();
    foreach (ManagementObject m in queryCollection)
    {
        //Detected MS VPC with Additions
        if (m["Description"].ToString() == "VM Additions S3 Trio32/64") { return true; }
        //Detected MS VPC without Additions
        if (m["Description"].ToString() == "S3 Trio32/64") { return true; }
        //Detected VirtualBox with Additions
        if (m["Description"].ToString() == "VirtualBox Graphics Adapter") { return true; }
        //Detected VMWare with Additions
        if (m["Description"].ToString() == "VMware SVGA II") {return true; }
        //Detected VMWare
        if (m["Description"].ToString().ToUpper().Contains("VMWARE")) {return true; }
        //Detected a VM
        if (m["Description"].ToString() == "") { return true; }
    }

    return false;
}

public static string regGet(string key, string value)
{
    RegistryKey registryKey;
    registryKey = Registry.LocalMachine.OpenSubKey(key, false);
    if (registryKey != null)
    {
        object rkey = registryKey.GetValue(value, (object)(string)"noValueButYesKey");
        if (rkey.GetType() == typeof(string))
        {

```

```

    return rkey.ToString();
}
if (registryKey.GetValueKind(value) == RegistryValueKind.String ||
registryKey.GetValueKind(value) == RegistryValueKind.ExpandString)
{
    return rkey.ToString();
}
if (registryKey.GetValueKind(value) == RegistryValueKind.DWord)
{
    return Convert.ToString((Int32)rkey);
}
if (registryKey.GetValueKind(value) == RegistryValueKind.QWord)
{
    return Convert.ToString((Int64)rkey);
}
if (registryKey.GetValueKind(value) == RegistryValueKind.Binary)
{
    return Convert.ToString((byte[])rkey);
}
if (registryKey.GetValueKind(value) == RegistryValueKind.MultiString)
{
    return string.Join("", (string[])rkey);
}
return "noValueButYesKey";
}

return "noKey";
}

}

```

Debroski - January 28, 2014, 12:51 pm

You stole 1 entire source and ask us to make it better? No.

xaHyeG - January 28, 2014, 12:56 pm

I did not steal the code, I grabbed c++ samples for anti vm/sandbox (from differet public source code) and made one big class in c# for it. How you dare call it stealing :(

kingroot - January 28, 2014, 12:58 pm

Some pretty clever ways of detecting sandboxie right there.

the anti-sandboxie detects if the application is running under a sandboxed environment, if it is, it doesn't execute the malicious backdoor code. ^ thats what OP is trying to do.

You can also see what sandboxie does via cheat engine and create checks for the api functions it modifies from usermode.

Debroski - January 28, 2014, 1:00 pm

Quote:if (username.ToString().ToUpper() == "Schmidt").ToUpper()) { return true; }
do you really need to throw the string in instead of typing SCHMIDTI?

xaHyeG - January 28, 2014, 1:16 pm

(Jan 28, 2014 02:00 PM)___Angel Wrote: [CITE](#) do you really need to throw the string in instead of

typing SCHMIDTI?

You are right, I will change that, Thanks.

Smelly Shoes - January 28, 2014, 2:41 pm

Just for example. This is how easy you can shorten your code and neaten it.

```
Code public static bool IsSandbox(string startupPath)
{

    StringBuilder username = new StringBuilder();
    Int32 nSize = 50;
    GetUserName(username, ref nSize);

    if ((int)GetModuleHandle("SbieDLL.dll") != 0)
        return true;

    switch(username.ToString().ToUpper())
    {
        case "USER": return true;
        case "SANDBOX": return true;
        case "VIRUS": return true;
        case "MALWARE": return true;
        case "SCHMIDTI": return true;
        case "CURRENTUSER": return true;
    }

    string sPath = startupPath.ToUpper();

    if (sPath == "C:\\FILE.EXE")
        return true;

    if (sPath.Contains("\\VIRUS"))
        return true;

    if (sPath.Contains("SANDBOX"))
        return true;

    if (sPath.Contains("SAMPLE"))
        return true;

    if ((int)FindWindow("Afx:400000:0", (IntPtr)0) != 0)
        return true;

    return false;
}
```

xaHyeG - January 29, 2014, 1:37 pm

(Jan 28, 2014 03:41 PM)Smelly Shoes Wrote: [\[CITE\]](#) Just for example. This is how easy you can shorten your code and neaten it.

```
Code public static bool IsSandbox(string startupPath)
{
```

```
StringBuilder username = new StringBuilder();  
Int32 nSize = 50;  
GetUserName(username, ref nSize);
```

```
if ((int)GetModuleHandle("SbieDLL.dll") != 0)  
return true;
```

```
switch(username.ToString().ToUpper())  
{  
case "USER": return true;  
case "SANDBOX": return true;  
case "VIRUS": return true;  
case "MALWARE": return true;  
case "SCHMIDTI": return true;  
case "CURRENTUSER": return true;  
}
```

```
string sPath = startupPath.ToUpper();
```

```
if (sPath == "C:\\FILE.EXE")  
return true;
```

```
if (sPath.Contains("\\VIRUS"))  
return true;
```

```
if (sPath.Contains("SANDBOX"))  
return true;
```

```
if (sPath.Contains("SAMPLE"))  
return true;
```

```
if ((int)FindWindow("Afx:400000:0", (IntPtr)0) != 0)  
return true;
```

```
return false;  
}
```

Thankyou! this will be nice and clean