NETTITUDE

A member of the Lloyd's Register group

COM and the PowerThief

MAY 2018

NETTITUDE

A member of the Lloyd's Register group

1: Introduction

NETTITUDE
A member of the Lloyd's Register group

C:\net user "Rob Maslen"/domain

Principal Security Consultant at Nettitude

- Red Teamer/Tools Developer
- Spent most of career in financial services writing C++, C#, SQL, HTML/JS…..
- Moved into InfoSec in 2013
- Developer of PoshC2 Modules
  - SharpSocks
  - Port/Arp Scanner
- Member of SpicyWeasel Winner 2017 DerbyCon CTF
- Thick Client Destruction Workshop Steelcon 2017
- Huge fan of C# and LinqPad

NETTITUDE
A member of the Lloyd's Register group

C:\net user "Rob Maslen"/domain

Some blogs I have written/co-written

- https://labs.nettitude.com/blog/poshc2-v3-with-socks-proxy-sharpsocks/
- https://labs.nettitude.com/blog/derbycon-2017-ctf-write-up/
- https://labs.nettitude.com/blog/a-quick-analysis-of-the-latest-shadow-brokers-dump/

Goals of this talk

- Talk about some Red Teaming tradecraft
- Show some problems we face, how we overcome and improve the solution over time
- Release PowerThIEf our new IE post exploitation tool
- And try to make the case to a room full of hardened IE fan-peeps that maybe it's time to use another browser ☺

Common Attributes of Primary/Secondary Red Team Objectives

- Web Application or Thick Client.
- Hold vast amounts of business critical/personal data.
- Legacy (very rare to see newly implemented core business applications).
- Ability to trade or transact.
- Quite often if a Web Application that accessed by Internet Explorer (IE) - still unfortunately the business browser of choice - which we will get onto later.

How do we gain access….We need creds

- Keylogging
- Thick Clients often have creds to the database within a config file/registry
- Get lucky and find an Excel spreadsheet/Word doc on the desktop called passwords
- Or if none of these work and the application is open we can extract from…..

NETTITUDE

A member of the Lloyd's Register group

2: Memory

Cleaning up memory

Thankfully few applications do it

- Once login has been completed any memory allocated that was storing credentials should be zeroed.
- This can be hard in managed languages such Java or C#.
- Or connection details encrypted at rest, decrypt them at start-up and leave them in memory for convenience.

What do we need to do?

Create a crash dump

- This is an image of memory and current the state of the processor
- This can then either be analysed (read and regex'd) on the target machine or loaded into a debugger once exfiltrated

Dumping processes

Methods

- Procump from Sysinternals https://docs.microsoft.com/en-us/sysinternals/downloads/procdump
- Task manager (Win 10: Right Click on process -> Create Dump File)

Issues

- Potentially taking sensitive client data out of the environment if analysing offline
- They can be big often ~1gb, can be very time consuming to "beacon" back out
- Not very elegant when searching for data you don't know the format of

Dumping processes

Alternative Methods

- Use VirtualQueryEx and ReadProcessMemory to scan target process's memory

VirtualQueryEx - https://msdn.microsoft.com/en-us/library/windows/desktop/aa366907(v=vs.85).aspx

RedProcessMemory - https://msdn.microsoft.com/en-gb/library/windows/desktop/ms680553(v=vs.85).aspx

An example

Cisco AnyConnect

- A common VPN client
- Runs as User process which handles the GUI and a Service which does the actual VPNing
- Service has had some privilege escalation bugs in the past
- Some customers will use single factor Domain Creds or Domain Password+token

Cisco AnyConnect in memory credentials

In addition to winning Risky Business's coveted accolade of "Worlds leading supplier of hardcoded credentials"

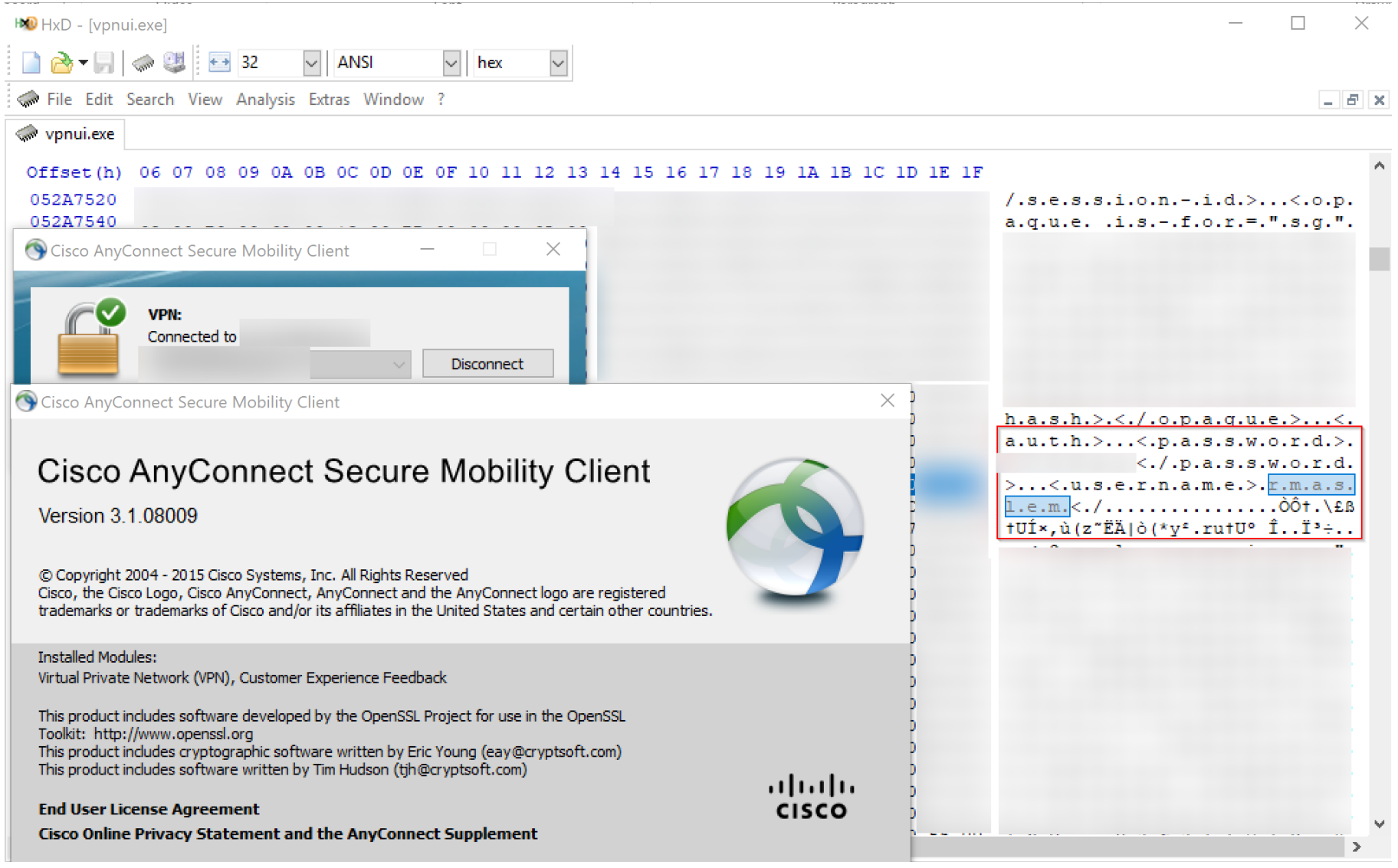
Cisco Bug: CSCtr80410 - Password may be available in clear text in RAM

https://quickview.cloudapps.cisco.com/quickview/bug/CSCtr80410
https://quickview.cloudapps.cisco.com/quickview/bug/CSCtf94284

Credit to those who reported these bugs

NETTITUDE
A member of the Lloyd's Register group

Who needs mimikatz ☺



How about dumping browsers for creds

Absolutely

- Browsers have multiple processes (yep welcome to the party Firefox), lots of files to search through
- Chrome in particular can have at least a gazillion running
- Definitely will take more out of the environment than you intend
- Can be very time consuming searching through a lot of dump files
- Lets face it, not very elegant

So IE being our "business browser of choice" is there a better way for it at least?

NETTITUDE

A member of the Lloyd's Register group

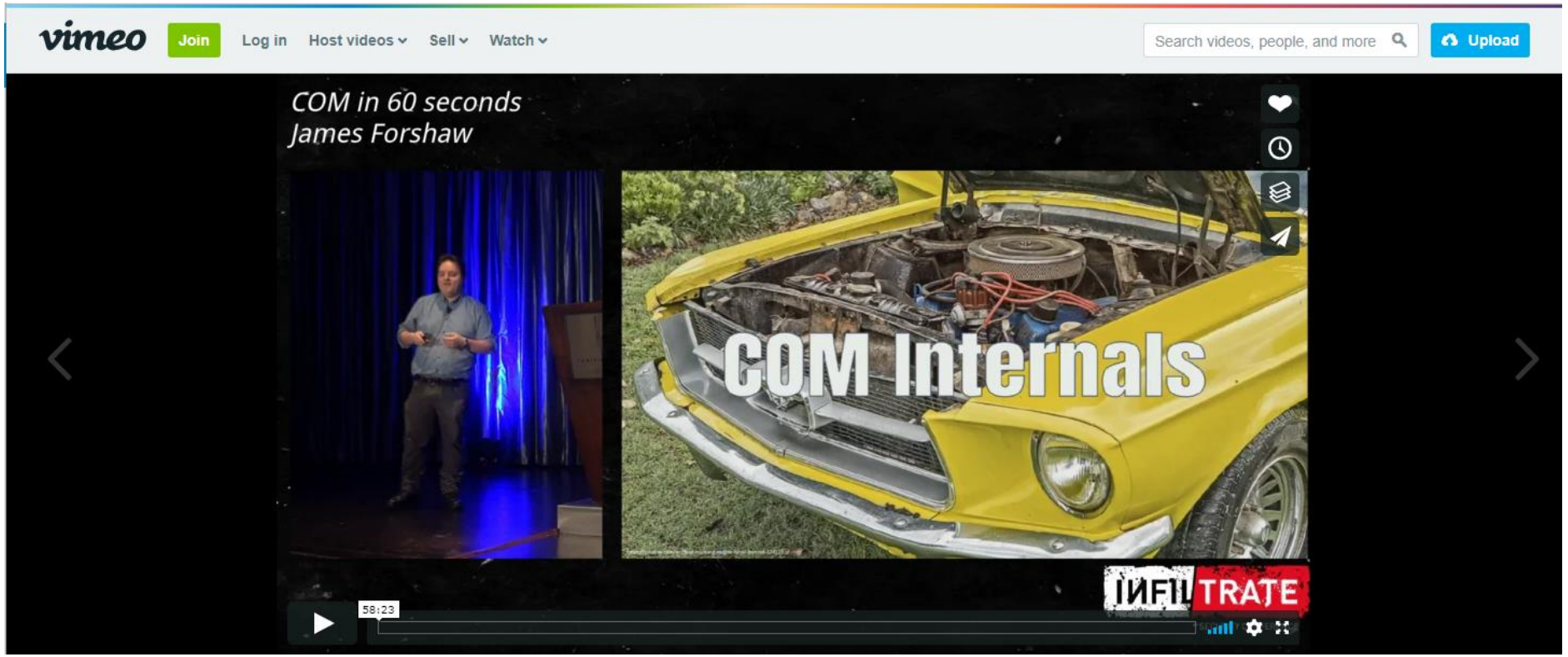
3: COM

COM

Introducing and explaining COM would take up most of the talk sooooo…….

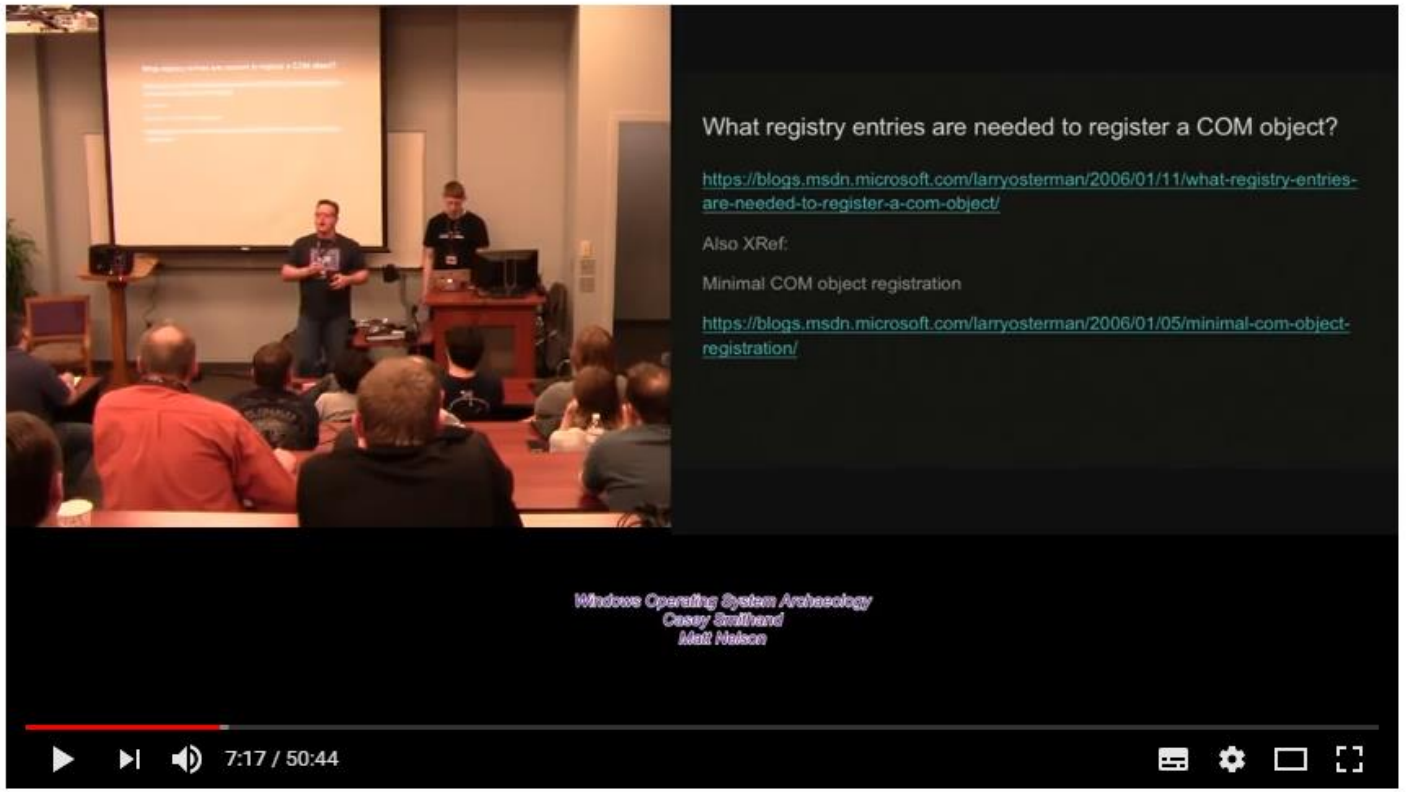
COM Introductions

https://vimeo.com/214856542



subTee and Matt Nelson's Windows Archaeology

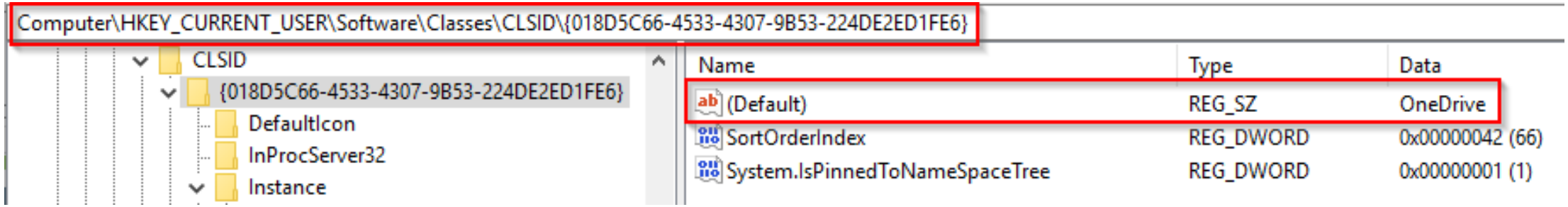
https://www.youtube.com/watch?v=3gz1QmiMhss

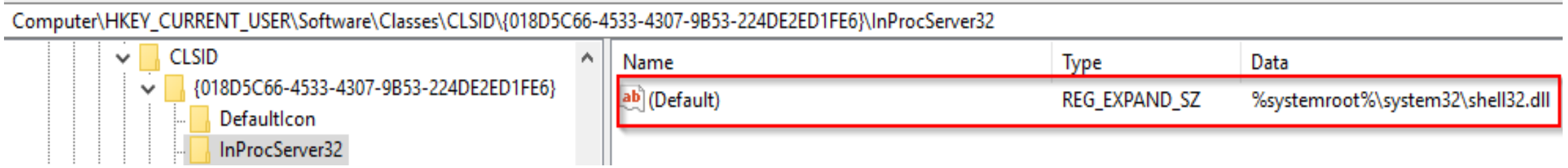


BSides Nashville 2017 Red01 Windows Operating System Archaeology Casey Smithand Matt Nelson

But if you haven't watched them yet well

- Application Binary Interface NOT Programming Language
- Enables different languages to talk to each other
- Consists of *classes* which are collections of *interfaces*
- Classes are identified by a CLSID (GUID) and details stored in the registry





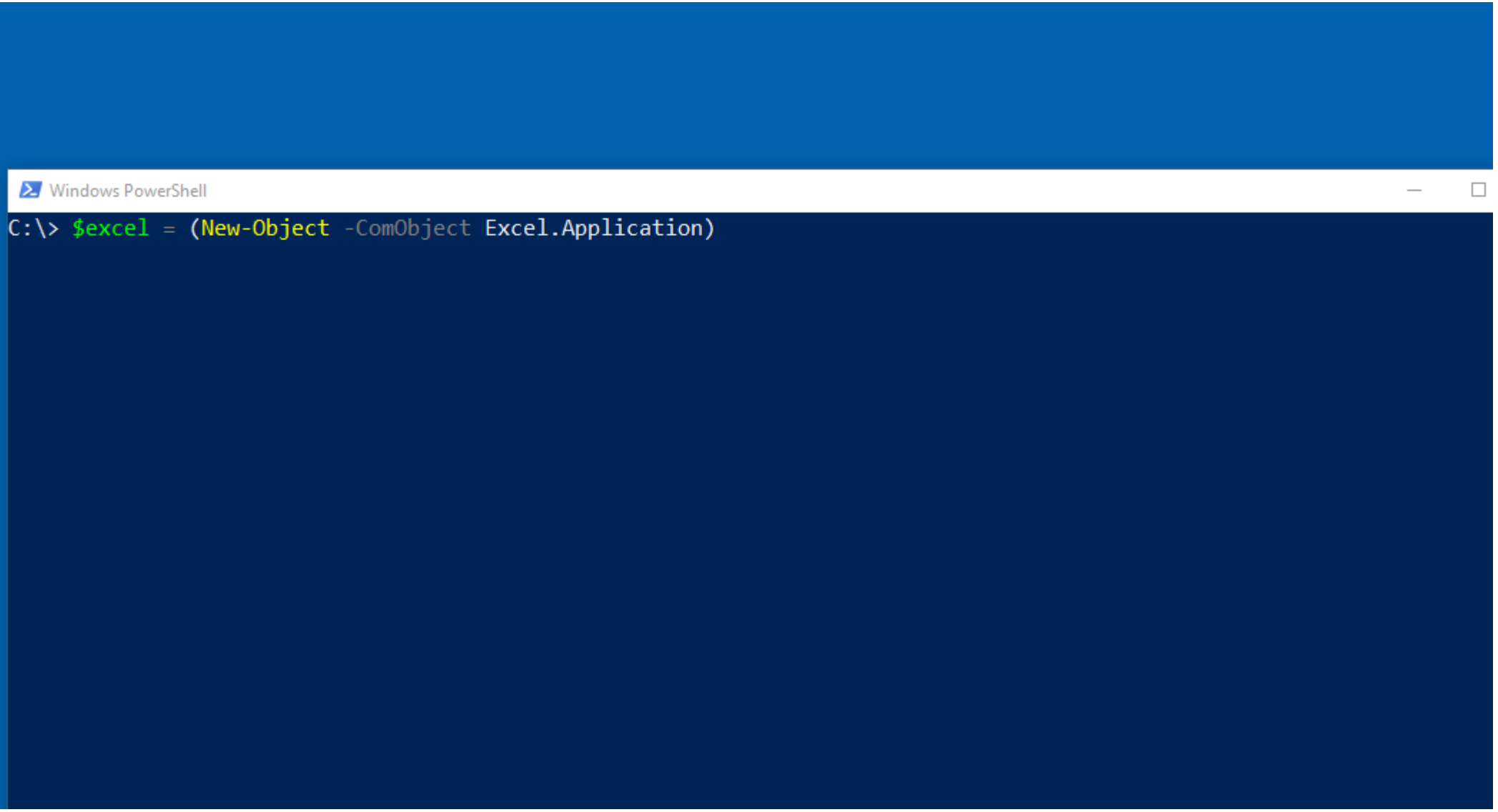
COM Concepts (Interfaces)

An Interface is a external contract between an object and the any callers. It defines what can be called but does NOT define what happens when it is. Naming convention is I<InterfaceName>
e.g. IUnknown

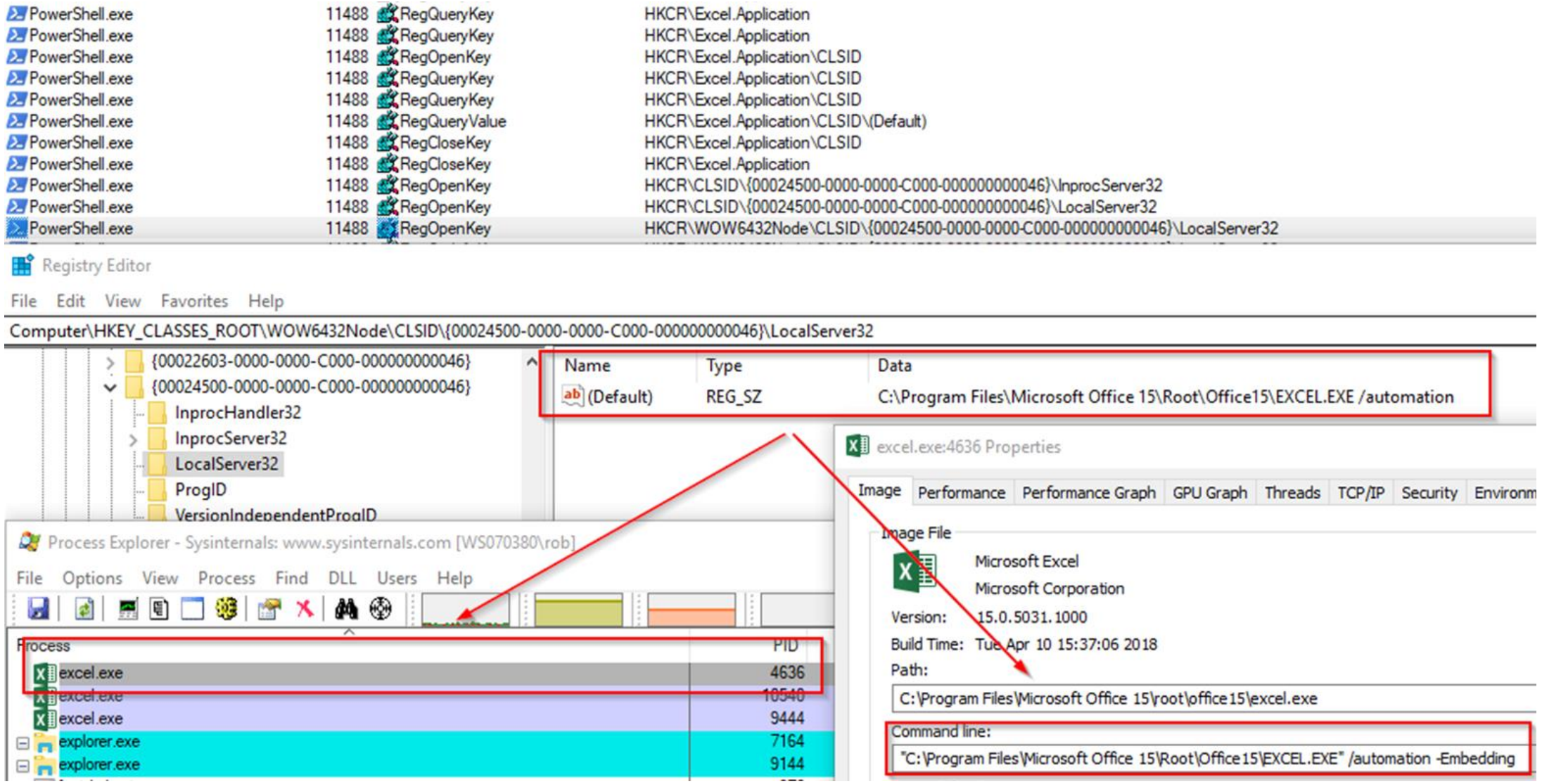
```
Inteface IDoor
{
  TurnHandle()
  bool Open()
}

Interface IDoorWithKey
{
  Unlock(IKey pKey)
}
```

COM Concepts (Activating)

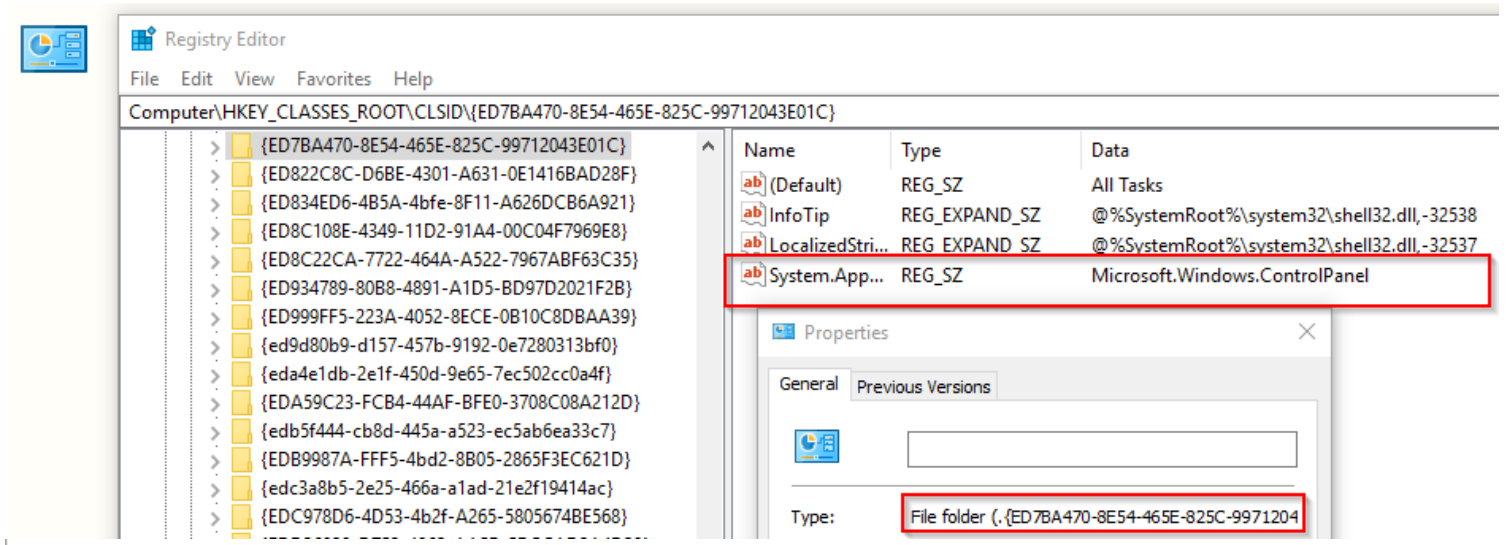


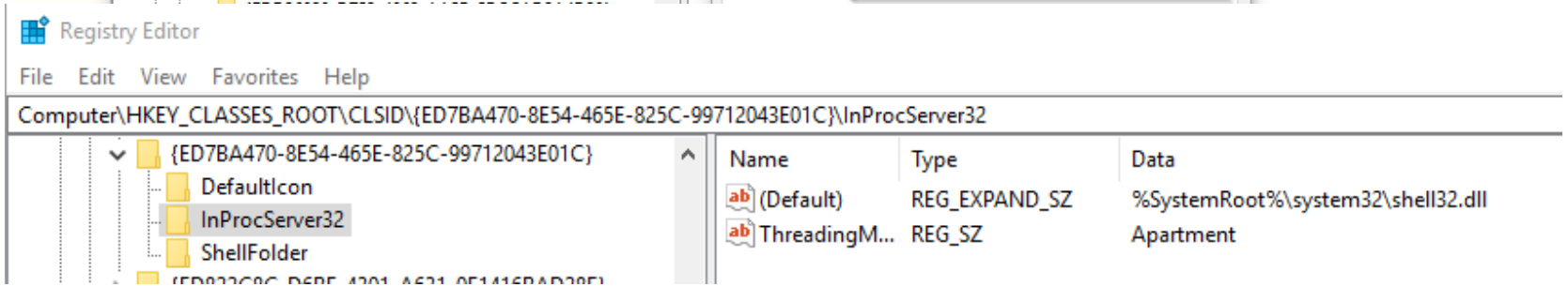
COM Concepts (Activating)



CLSID and Junction Folders

Technique was leaked last year in the Vault7 dumps





CLSID and Junction Folders

b33f (@fuzzysec) has some information and code on the technique in his DefCon 25 workshop (definitely worth a read anyway!!)
https://github.com/FuzzySecurity/DefCon25

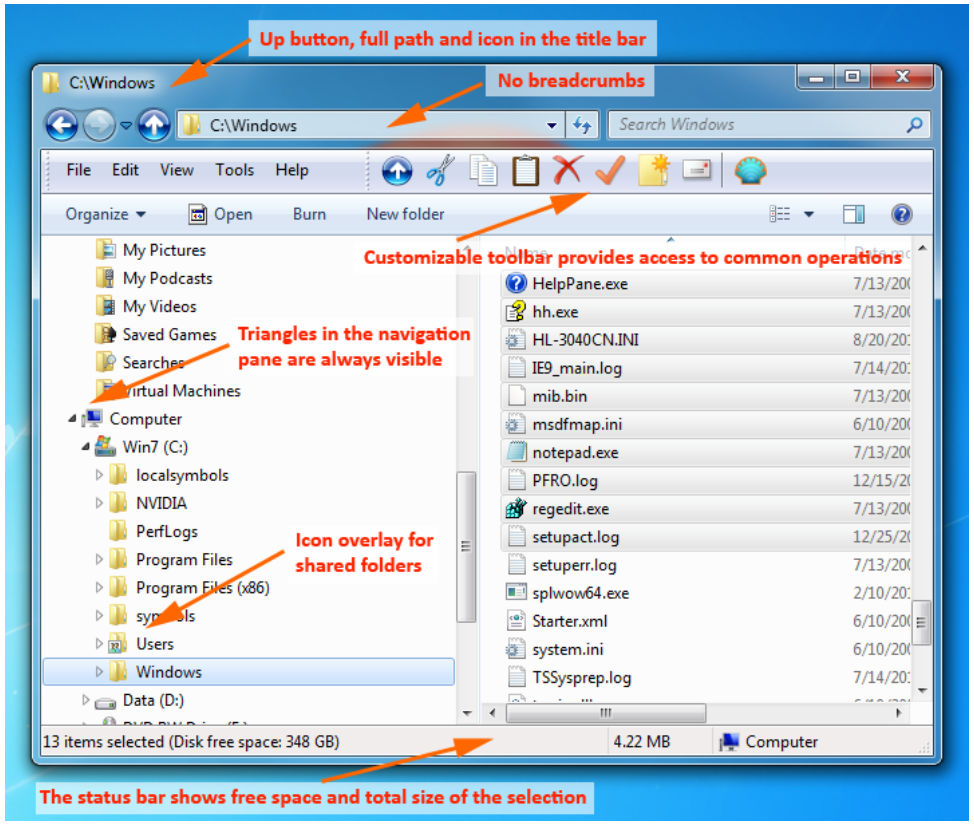
Creating your own registry entries & folder, then navigating means you will launch a DLL within *explorer.exe*.

This was seen as a persistence mechanism but I wanted to see if you could use it for migration (yeah I know it means dropping a dll)

The Windows Shell

The Shell represents UI Components in Windows, the following are examples

- Taskbar
- Desktop
- Explorer
- AutoRun
- Recycle Bin
- Internet Explorer
- Right Click menu



The Shell can be customised and replaced an example here is from https://www.classicshell.net

ShellWindows Object

https://msdn.microsoft.com/en-us/library/windows/desktop/bb773974(v=vs.85).aspx

ShellWindows object

Represents a collection of the open windows that belong to the Shell. Methods associated with this objects can control and execute commands within the Shell, and obtain other Shell-related objects.

Members

The **ShellWindows** object has these types of members:

- Methods
- Properties

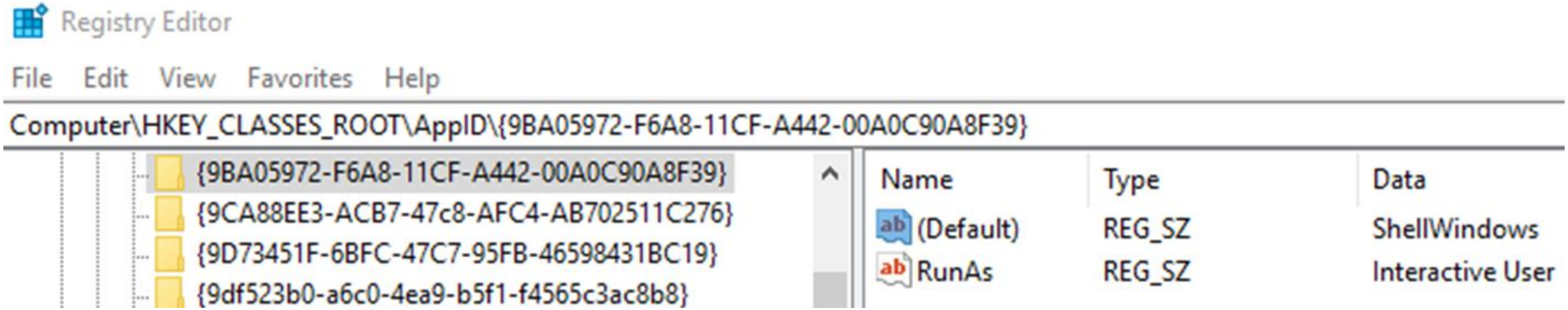
Methods

The **ShellWindows** object has these methods.

Method	Description
_NewEnum	Creates and returns a new **ShellWindows** object that is a copy of this **ShellWindows** object.
Item	Retrieves an **InternetExplorer** object that represents the Shell window.

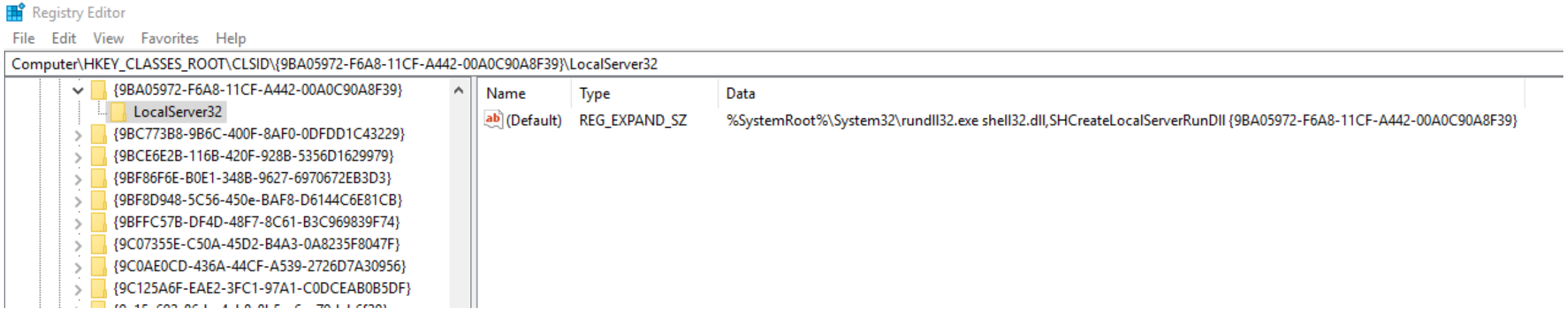
ShellWindows Object

The ShellWindow object can be found in the registry



ShellWindows Object

The ShellWindow object can be found in the registry



IWebBrowser2 interface

Filter

- IWebBridge
- IWebBrowser2
  - AddressBar
  - Application
  - Busy
  - Container
  - Document
  - ExecWB
  - FullName
  - FullScreen
  - GetProperty
  - GoBack
  - GoForward
  - GoHome
  - GoSearch
  - Height
  - Left
  - LocationName
  - LocationURL
  - MenuBar
  - Navigate

IWebBrowser2 interface

05/02/2017 • 5 minutes to read

Exposes methods that are implemented by the **WebBrowser** control (Microsoft ActiveX control) or implemented by an instance of the **InternetExplorer** application (OLE Automation). For the Microsoft .NET Framework version of this control, see WebBrowser Control (Windows Forms).

Members

The **IWebBrowser2** interface inherits from the **IDispatch** interface. **IWebBrowser2** also has these types of members:

- Methods
- Properties

Methods

The **IWebBrowser2** interface has these methods.

Method	Description
IWebBrowser2::ClientToWindow	Computes the full size of the browser window when given the specified width and height of the content area.
IWebBrowser2::ExecWB	Executes a command and returns the status of the command execution using the IOleCommandTarget interface.
IWebBrowser2::GetProperty	Gets the value associated with a user-defined property name.

IWebBrowser2::Navigate2

In the remarks section is this interesting note

https://msdn.microsoft.com/en-us/library/aa752134(v=vs.85).aspx

Remarks

See IWebBrowser2::Navigate for additional usage notes.

This method extends the **IWebBrowser2::Navigate** method to allow for Shell integration; however, this method does not make **IWebBrowser2::Navigate** obsolete. The original method can still be used for URL navigations.

IWebBrowser2::Navigate2

And confirmed in that the first param can take a reference to a shell folder

Navigates the browser to a location that might not be expressed as a URL, such as a pointer to an item identifier list (PIDL) for an entity in the Windows Shell namespace.

Syntax

```
HRESULT Navigate2(
    VARIANT *URL,
    VARIANT *Flags,
    VARIANT *TargetFrameName,
    VARIANT *PostData,
    VARIANT *Headers
);
```

Parameters

URL

[in] A pointer to a **VARIANT** that evaluates to the URL of the resource to display, the full path to the file location, or a PIDL that represents a folder in the Shell namespace.

Flags

[in] A pointer to a **VARIANT** of type **VT_I4** or **VT_I2** that specifies a combination of the values defined by the BrowserNavConstants enumeration.

Shell Folders

Turns out there is a way to navigate

- A well documented breakout method of getting round group policy file restrictions
- You can nav shell:<locations> to eg “shell:Administrative Tools”
- After a load of Googling it is also possible to navigate to
  - **shell:::{CLSID}**

So back to ShellWindows

By activating the ShellWindows object and enumerating the Item member we should have access to any open IExplore or Explorer window right?? Remember the CLSID for control panel….

```
$shellWinGuid = [System.Guid]::Parse("{9BA05972-F6A8-11CF-A442-
00A0C90A8F39}")
$typeShwin = [System.Type]::GetTypeFromCLSID($shellWinGuid)
$shwin = [System.Activator]::CreateInstance($typeShwin)
$shWin[0].Navigate2("shell:::{ED7BA470-8E54-465E-825C-99712043E01C}",
2048)

/*CLSID must be in the format "shell::{CLSID}"
Second param 2048 is BrowserNavConstants value for navOpenInNewTab
https://msdn.microsoft.com/en-us/library/dd565688(v=vs.85).aspx
Further ideas on what payloads you may be able to use
https://bohops.com/2018/06/28/abusing-com-registry-structure-clsid-
localserver32-inprocserver32/ */
```

Navigating to CLSIDs (Control Panel in this case)

Navigating to CLSIDs (Control Panel in this case)

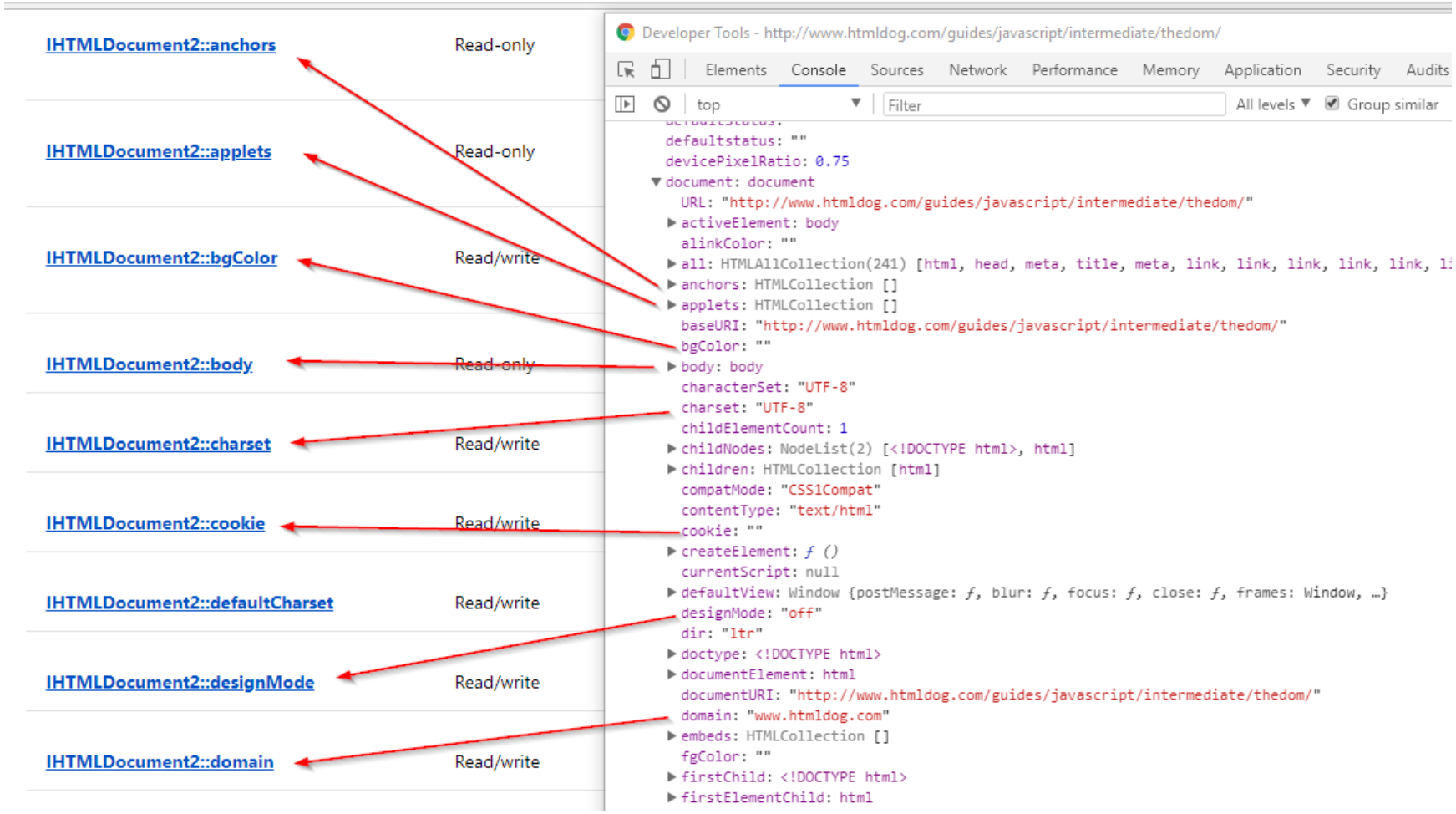
So does this mean we can craft our own CLSIDs and have IE launch them from the comfort of another process, oohh yeah!!

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

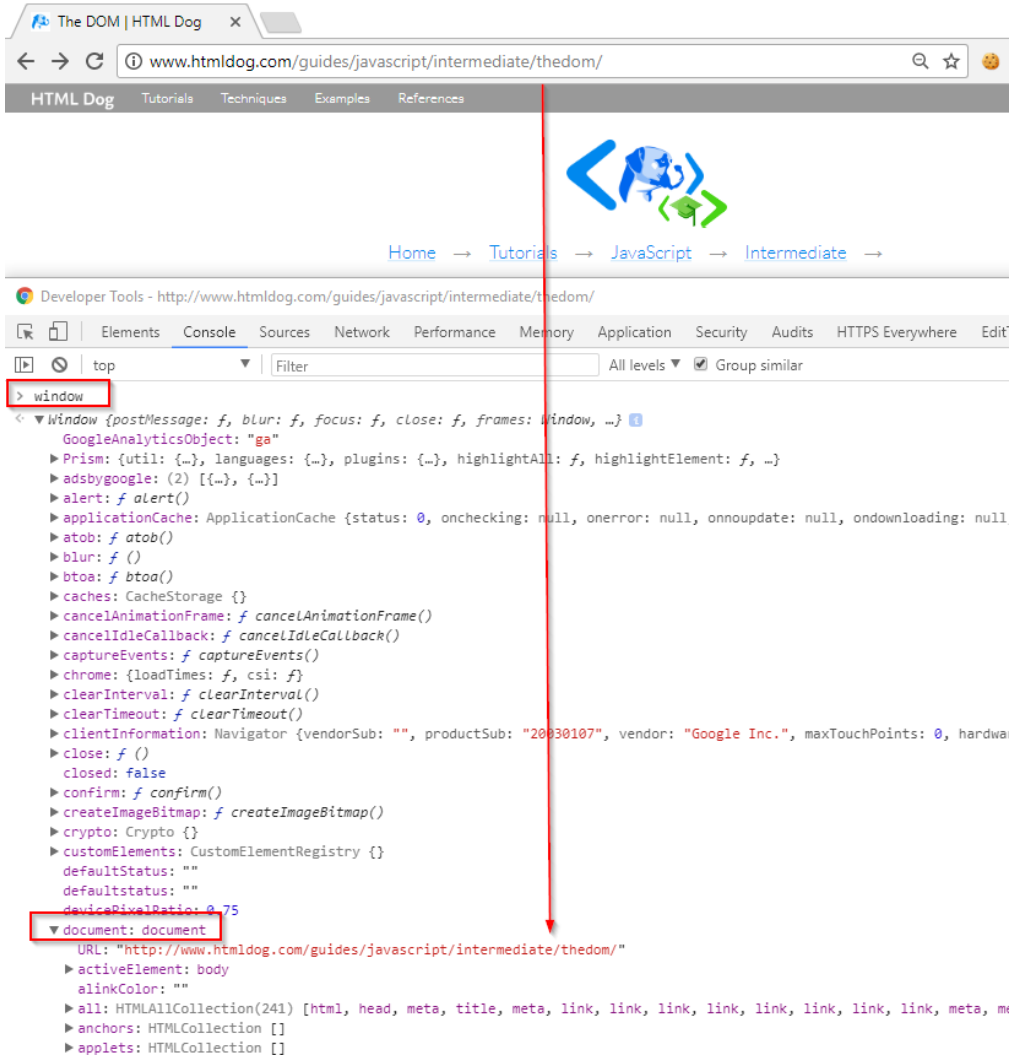
Process Name	PID	Operation	Path	Result	Detail
iexplore.exe	12232	RegOpenKey	HKCU\Software\Classes\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}	NAME NOT FOUND	Desired Access: Read
iexplore.exe	12232	RegOpenKey	HKCR\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}	SUCCESS	Desired Access: Read
iexplore.exe	12232	RegQueryKey	HKCR\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}	SUCCESS	Query: Name
iexplore.exe	12232	RegQueryKey	HKCR\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}	SUCCESS	Query: HandleTags, HandleTags
iexplore.exe	12232	RegOpenKey	HKCU\Software\Classes\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}	NAME NOT FOUND	Desired Access: Maximum Allowe
iexplore.exe	12232	RegQueryValue	HKCR\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}\SortOrderIndex	NAME NOT FOUND	Length: 144
iexplore.exe	12232	RegCloseKey	HKCR\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}	SUCCESS	
iexplore.exe	12232	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellCompatibility\Objects\{ED7BA470-8E54-465E-825C-99712043E01C}	NAME NOT FOUND	Desired Access: Query Value
iexplore.exe	12232	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Desktop\NameSpace\{ED7BA470-8E54-465E-825C-99712043E01C}	NAME NOT FOUND	Desired Access: Query Value
iexplore.exe	12232	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Desktop\NameSpace\{ED7BA470-8E54-465E-825C-99712043E01C}	NAME NOT FOUND	Desired Access: Query Value
iexplore.exe	12232	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellCompatibility\Objects\{ED7BA470-8E54-465E-825C-99712043E01C}	NAME NOT FOUND	Desired Access: Query Value
iexplore.exe	12232	RegOpenKey	HKCU\Software\Classes\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}\Instance	NAME NOT FOUND	Desired Access: Read
iexplore.exe	12232	RegOpenKey	HKCR\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}\Instance	NAME NOT FOUND	Desired Access: Read
iexplore.exe	12232	RegOpenKey	HKCU\Software\Classes\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}\Instance	NAME NOT FOUND	Desired Access: Read
iexplore.exe	12232	RegOpenKey	HKCR\CLSID\{ED7BA470-8E54-465E-825C-99712043E01C}\Instance	NAME NOT FOUND	Desired Access: Read

DOM to COM Mapping



From https://docs.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/platform-apis/aa752127%28v%3dvs.85%29

The DOM


The DOM | HTML Dog
www.htmldog.com/guides/javascript/intermediate/thedom/
HTML Dog
Tutorials
Techniques
Examples
References
Home → Tutorials → JavaScript → Intermediate →
Developer Tools - http://www.htmldog.com/guides/javascript/intermediate/thedom/
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
HTTPS Everywhere
top
Filter
All levels
Group similar
> window
Window {postMessage: ƒ, blur: ƒ, focus: ƒ, close: ƒ, frames: Window, …}
GoogleAnalyticsObject: "ga"
Prism: {util: {…}, languages: {…}, plugins: {…}, highlightAll: ƒ, highlightElement: ƒ, …}
adsbygoogle: (2) [{…}, {…}]
alert: ƒ alert()
applicationCache: ApplicationCache {status: 0, onchecking: null, onerror: null, onnoupdate: null, ondownloading: null
atob: ƒ atob()
blur: ƒ ()
btoa: ƒ btoa()
caches: CacheStorage {}
cancelAnimationFrame: ƒ cancelAnimationFrame()
cancelIdleCallback: ƒ cancelIdleCallback()
captureEvents: ƒ captureEvents()
chrome: {loadTimes: ƒ, csi: ƒ}
clearInterval: ƒ clearInterval()
clearTimeout: ƒ clearTimeout()
clientInformation: Navigator {vendorSub: "", productSub: "20030107", vendor: "Google Inc.", maxTouchPoints: 0, hardwa
close: ƒ ()
closed: false
confirm: ƒ confirm()
createImageBitmap: ƒ createImageBitmap()
crypto: Crypto {}
customElements: CustomElementRegistry {}
defaultStatus: ""
defaultstatus: ""
devicePixelRatio: 0.75
document: document
URL: "http://www.htmldog.com/guides/javascript/intermediate/thedom/"
activeElement: body
alinkColor: ""
all: HTMLAllCollection(241) [html, head, meta, title, meta, link, link, link, link, link, link, link, link, meta, me
anchors: HTMLCollection []
applets: HTMLCollection []

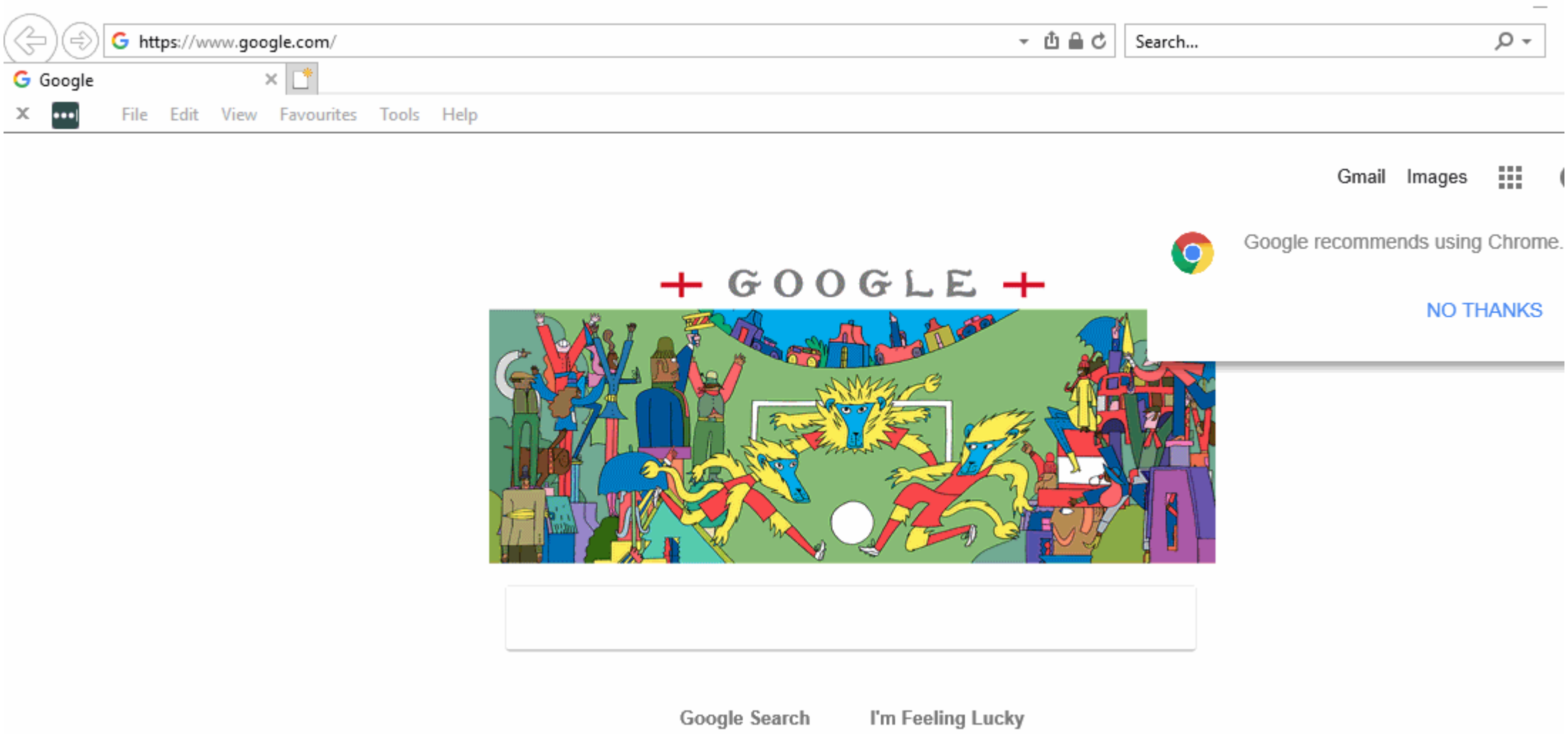
Accessing the DOM

Activating the ShellWindows object and this time looking at the DOM (via .Document).

Just remember that Explorer instances can be returned and if you want the DOM you need to filter for IExplore instances. An easy way is

```
$shellWinGuid = [System.Guid]::Parse("{9BA05972-F6A8-11CF-A442-
00A0C90A8F39}")
$typeShwin = [System.Type]::GetTypeFromCLSID($shellWinGuid)
$shwin = [System.Activator]::CreateInstance($typeShwin)
Write-Host $shWin[0].Document.GetType().ToString()
$loc = $shWin[0].Document.location.href
Write-Host "Location: $loc"
$shWin[0].Document |gm
```

So back to ShellWindows



Very important to remember

The ShellWindows.Item collection will contain both IE and Explorer instances which support different interfaces

An easy way is to use the .FullName property which contains the image name

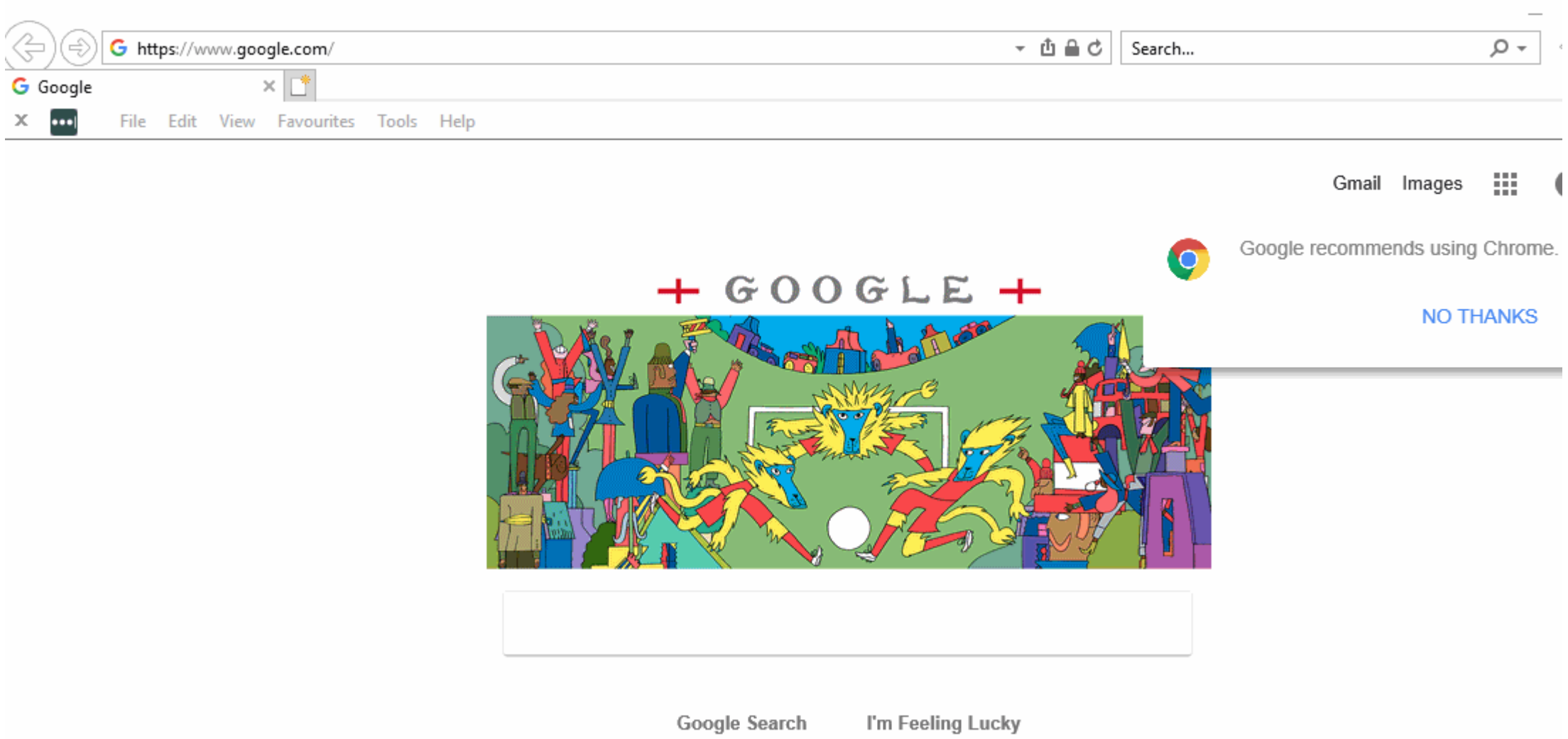
```
$fileName = [System.IO.Path]::GetFileNameWithoutExtension($shWin[0].FullName);
if ($fileName.ToLower().Equals("iexplore"))
{
        // Do your IE stuff here
}
```

Everyone loves a XSS style popup err don't they??

Taking that a bit further, what do you think will happen?

```
$shellWinGuid = [System.Guid]::Parse("{9BA05972-F6A8-11CF-A442-
00A0C90A8F39}")
$typeShwin = [System.Type]::GetTypeFromCLSID($shellWinGuid)
$shwin = [System.Activator]::CreateInstance($typeShwin)
$parentwin = $shWin[0].Document.parentWindow
$parentwin.GetType().InvokeMember("eval",
[System.Reflection.BindingFlags]::InvokeMethod, $null, $parentwin,
@("alert('Hi SteelCon 2018!! \r\n\r\n from '+ document.location.href)"))
```

So back to ShellWindows



Self XSS is no fun!!

Matt Nelson demonstrated that it's possible to laterally move via ShellWindows by passing an IP to GetTypeFromCLSID
https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2/

```
$shellWinGuid = [System.Guid]::Parse("{9BA05972-F6A8-11CF-A442-00A0C90A8F39}")
$typeShwin = [System.Type]::GetTypeFromCLSID($shellWinGuid, "xxx.xxx.xxx.xxx")
$shwin = [System.Activator]::CreateInstance($typeShwin)
$parentwin = $shWin[0].Document.parentWindow
$parentwin.GetType().InvokeMember("eval",
[System.Reflection.BindingFlags]::InvokeMethod, $null, $parentwin, @("alert('Hi
SteelCon 2018!! \r\n\r\n from another Machine '+ document.location.href)"))
```

Kinda looks useful

Putting this all together we have a methods of

- Enumerating Internet Explorer windows
- Injecting JavaScript
- Navigating/editing the DOM
- Getting IE to navigate to a CLSID and loading DLLs of our choosing

HMMM considering the targets we face this might make a useful library….

NETITUDE

A member of the Lloyd's Register group

4: The PowerThIEf

Design goals

Build a library with the previous functionality and more

- Write it in PowerShell where possible (written a lot of software but first time with PowerShell)
- Compatible with tools such as Posh C2
- Try not to touch disk (not always possible im afraid)

Design goals

Be useful in scenarios Red Teamers face like

- The target is using a password manager e.g. LastPass where keylogging is ineffective
- User is logged into application and want to be able to log them out without having clear all browser history and cookies.
- The target application is in a background tab and can‘t wait for user to switch tabs. Need to view or get HTML from the page
- Want to view a site from the targets IP without the target being aware.

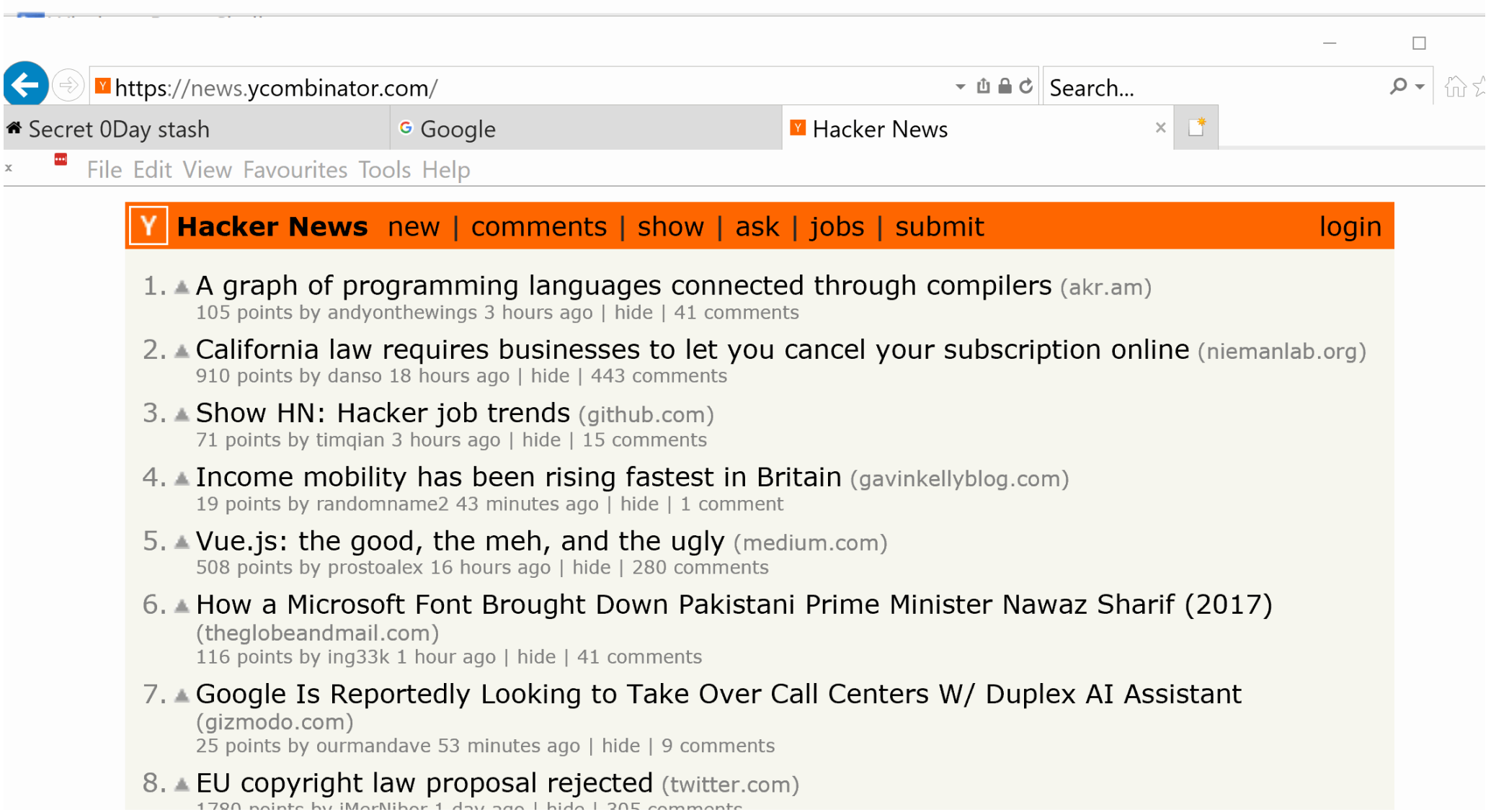
Introducing PowerThIEf

It's will be on GitHub today @ https://github.com/nettitude/Invoke-PowerThief

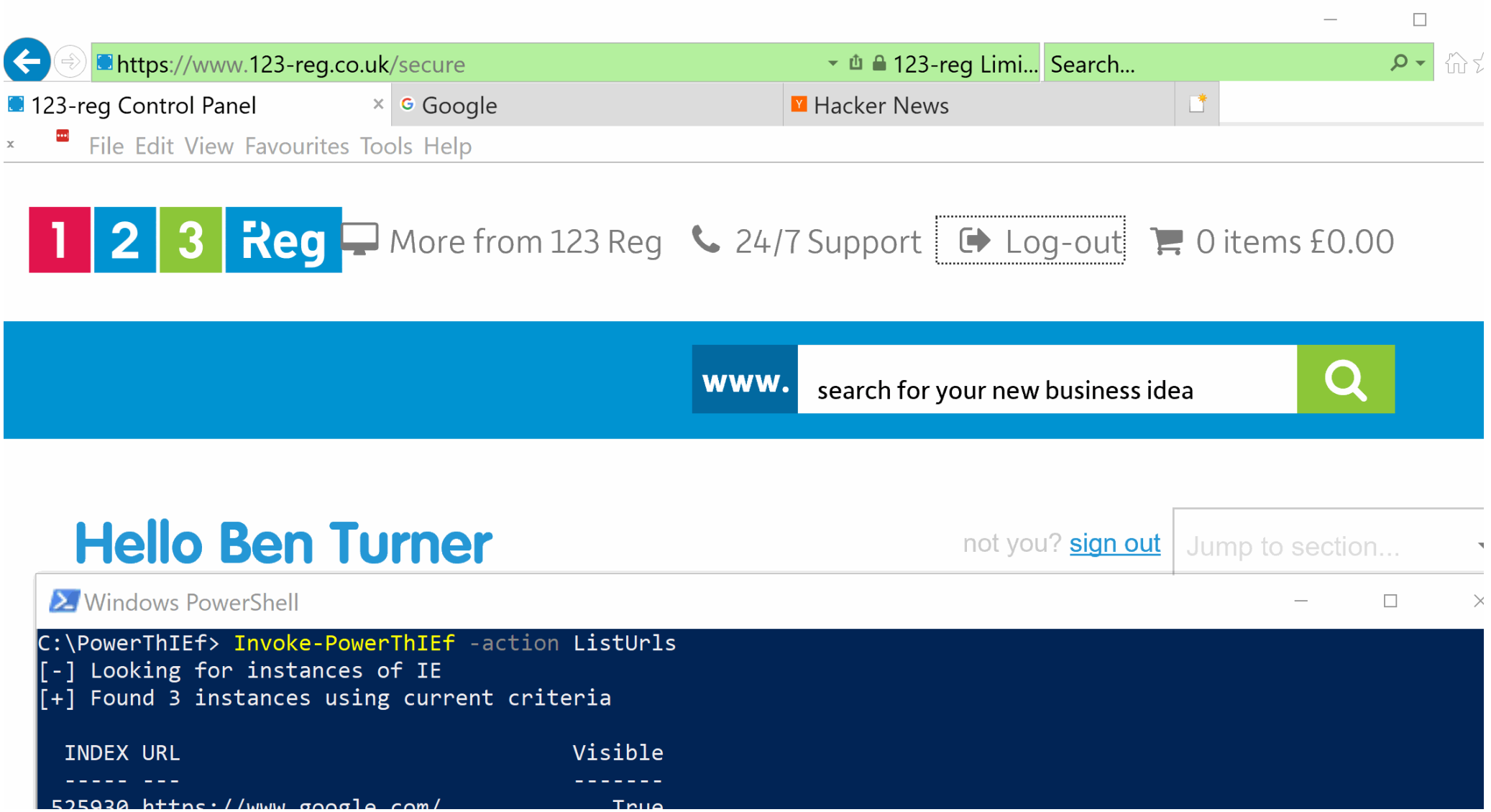
Current Functionality

- **DumpHtml**: retrieves HTML from the DOM, can use some selectors just not jQuery style yet…
- **ExecPayload**: Uses the migrate technique from earlier to launch a payload DLL in IE
- **Show/HideWindow**: Shows or Hides a browsing window
- **HookLoginForms**: Steals credentials by hooking the login form and monitoring for new windows.
- **InvokeJS** : Executes JavaScript in the window of your choice
- **ListUrls**: Lists the urls of all currently opened tabs/windows
- **Navigate**: Navigate to another URL
- **NewBackgroundTab**: Creates a new tab in the background

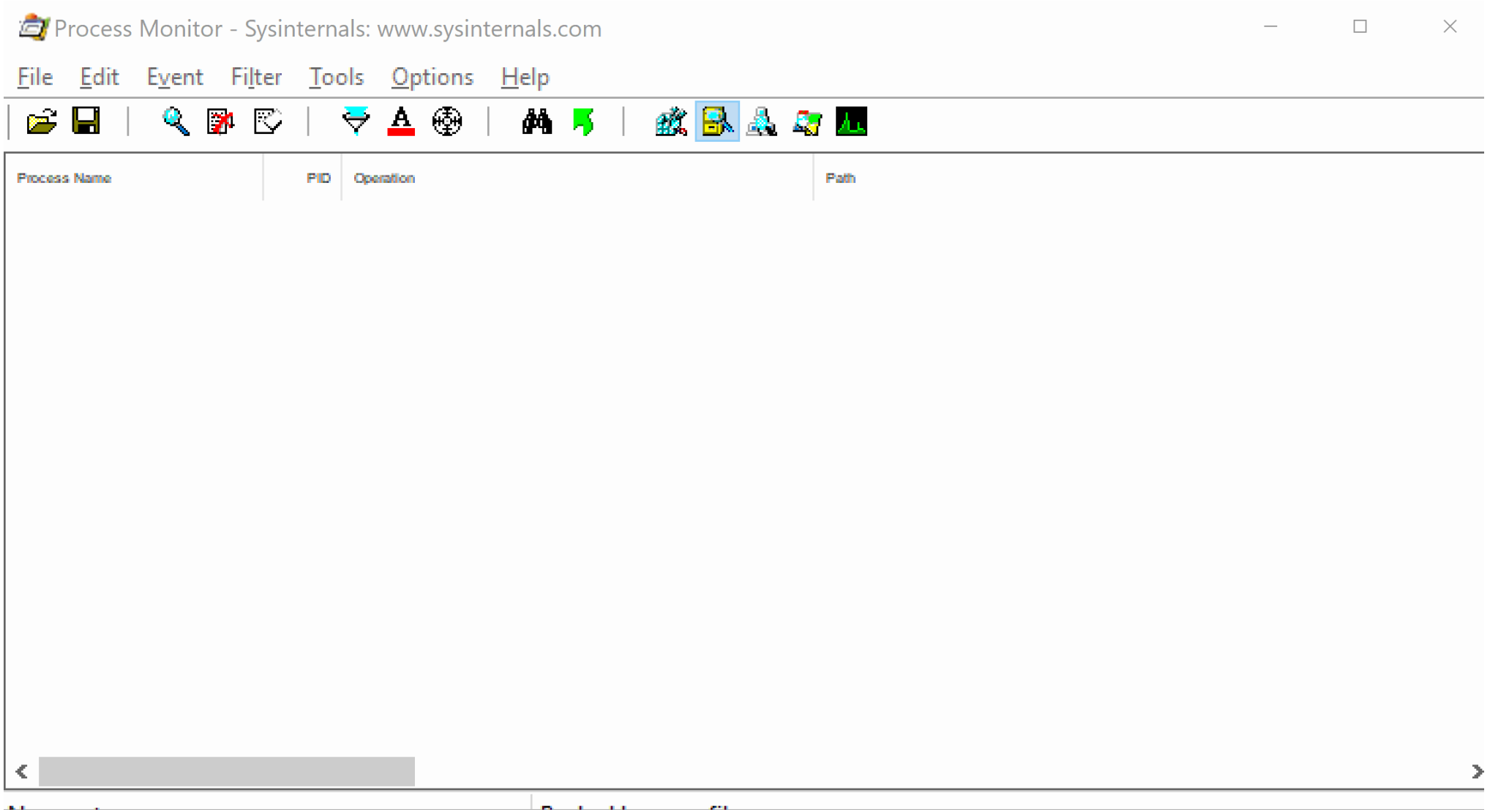
Demo – Extracting HTML from the page



Demo – Logging the user out



Demo – PoshC2 Payload via Migrate



Automated Credential ThIEvery

What we would like (high level)

- IE is tracked for when windows/tabs are opened & closed
- When a tab/window loads a page it is automatically scanned to check if there any indicators of a login form
- Monitor when the user clicks 'Submit' or 'Login' and capture the creds in transit
- Avoid rewriting the target of the form to be an attacker controlled server (BlackHat Python's method). When testing this I found LastPass could detect it.

Automated Credential ThIEvery

Is this possible? Well yes

- Opening and closing of windows can be tracked via the events *WindowRegistered* and *WindowRevoked* events
- When a window is opened the it's DOM can be obtained and the *DocumentComplete event* hooked
- When the *DocumentComplete* event fires, the content of the DOM could be extracted and scanned for any forms or input fields that look like will be for a logon.
- *onSubmit* could then be hooked and any details posted (username/creds) are recorded
- Yeah that sounds really easy, right!!

Hmm maybe not

Need access to these events to track opening and closing

DShellWindowsEvents object

31/05/2018 • 2 minutes to read

Receives IShellWindows window registration events.

Members

The **DShellWindowsEvents** object has these types of members:

- Methods

Methods

The **DShellWindowsEvents** object has these methods.

Method	Description
WindowRegistered	Called when a window is registered as a Shell window.
WindowRevoked	Called when a Shell window's registration is revoked.

Credential ThIEvery

This is where it gets a bit more complex

If you activate ShellWindows by it's GUID you can't see them.

```
$ishellGuid = "{9BA05972-F6A8-11CF-A442-00A0C90A8F39}"
$ishllwn = [System.Guid]::Parse($ishellGuid)
$typeShwin = [System.Type]::GetTypeFromCLSID($ishllwn)
$shwin = [System.Activator]::CreateInstance($typeShwin)
```

Credential ThIEvery

It exists but we can't register to handle to it

```
C:\> $ishllwn = [System.Guid]::Parse("{9BA05972-F6A8-11CF-A442-00A0C90A8F39}")
>> $typeShwin = [System.Type]::GetTypeFromCLSID($ishllwn)
>> $shwin = [System.Activator]::CreateInstance($typeShwin)
>>
C:\> $shwin.WindowRegistered | gm


   TypeName: System.Management.Automation.ComInterop.BoundDispEvent

Name                 MemberType Definition
----                 ---------- ----------
Equals               Method     bool Equals(System.Object obj)
GetDynamicMemberNames Method     System.Collections.Generic.IEnumerable[string] GetDynamicMemberNames()
GetHashCode          Method     int GetHashCode()
GetMetaObject        Method     System.Dynamic.DynamicMetaObject GetMetaObject(System.Linq.Expressions.Expression p...
GetType              Method     type GetType()
```

```
C:\> Register-ObjectEvent -InputObject $shwin[0] -EventName WindowRegistered -SourceIdentifier "ShWindowRegistered" -Act
ion { write-host "yeah" }
Register-ObjectEvent : Cannot register for the specified event. An event with the name 'WindowRegistered' does not
exist.
Parameter name: eventName
```

Credential ThIEvery

ShellWindows activated like this only contains a subset of types. What we actually need is much more detailed type information

- COM has the concept of a type library, these can be in the format of a *.tlb* file, a *.dll* or as a resource in DLL.
- They can contain
  - Information about data types, such as aliases, enumerations, structures, or unions.
  - Descriptions of one or more objects
  - References to type descriptions from other type libraries
- https://msdn.microsoft.com/en-us/library/windows/desktop/ms221355(v=vs.85).aspx

Credential ThIEvery

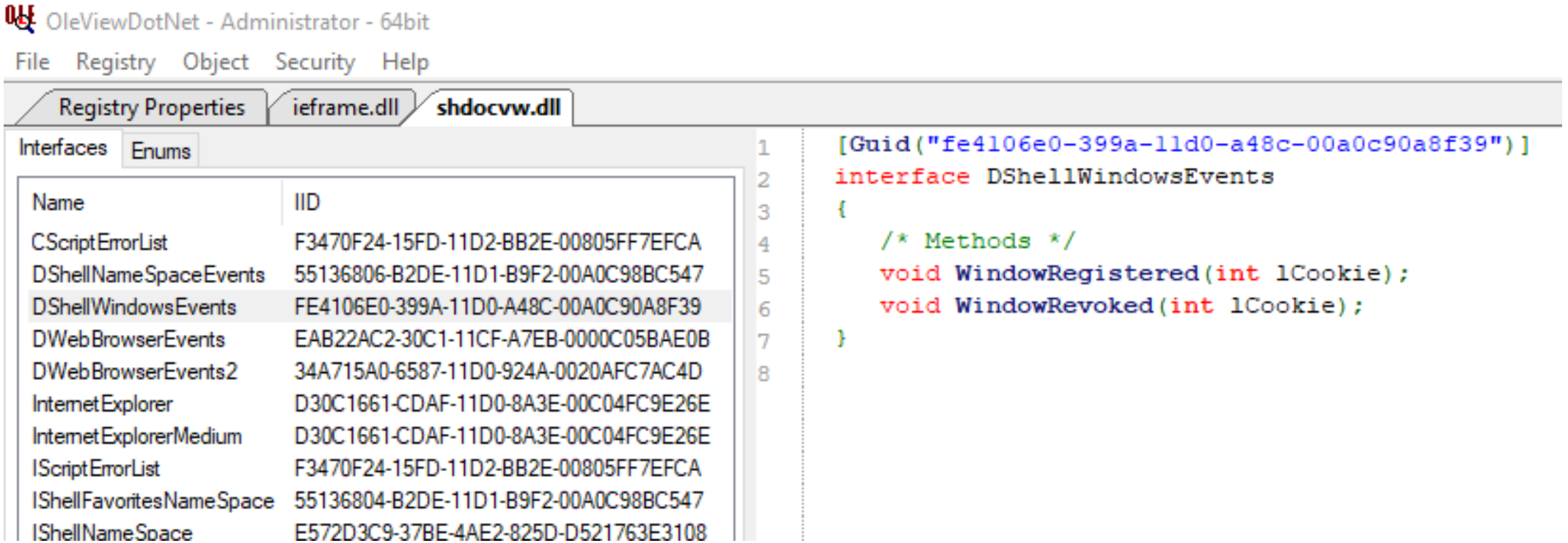
There are two libraries that we need to load which are two key components of Internet Explorer

- IEFrame.dll
  - Appears to contain a subset of functions from BrowseUI and SHDocVW.dll
- SHDocVW.dll
  - "In a very real sense, SHDOCVW.DLL is Internet Explorer." https://www.geoffchappell.com/studies/windows/ie/shdocvw/index.htm?tx=4,5,7,42,83,84,87-89

Credential ThIEvery

Open shdocvw.dll in OleViewDotNet

https://github.com/tyranid/oleviewdotnet



Credential ThIEvery

In order to get access to the types we need to

- Load IEFrame.dll as a type library
- Generate a .Net wrapper around the types in SHDocVW.dll
- Thankfully there are helpers functions

Credential ThIEvery

To load ieframe use pinvoke to call this function

LoadTypeLibEx function

04/05/2018 • 2 minutes to read

Loads a type library and (optionally) registers it in the system registry.

Syntax

Copy

```
HRESULT LoadTypeLibEx(
  LPCOLESTR szFile,
  REGKIND   regkind,
  ITypeLib  **pptlib
);
```

Parameters

`szFile`

The type library file.

Credential ThIEvery

This function makes the type accessible to .Net by generating a wrapper assembly for SHdocVw.dll

Syntax

C# | C++ | F# | VB

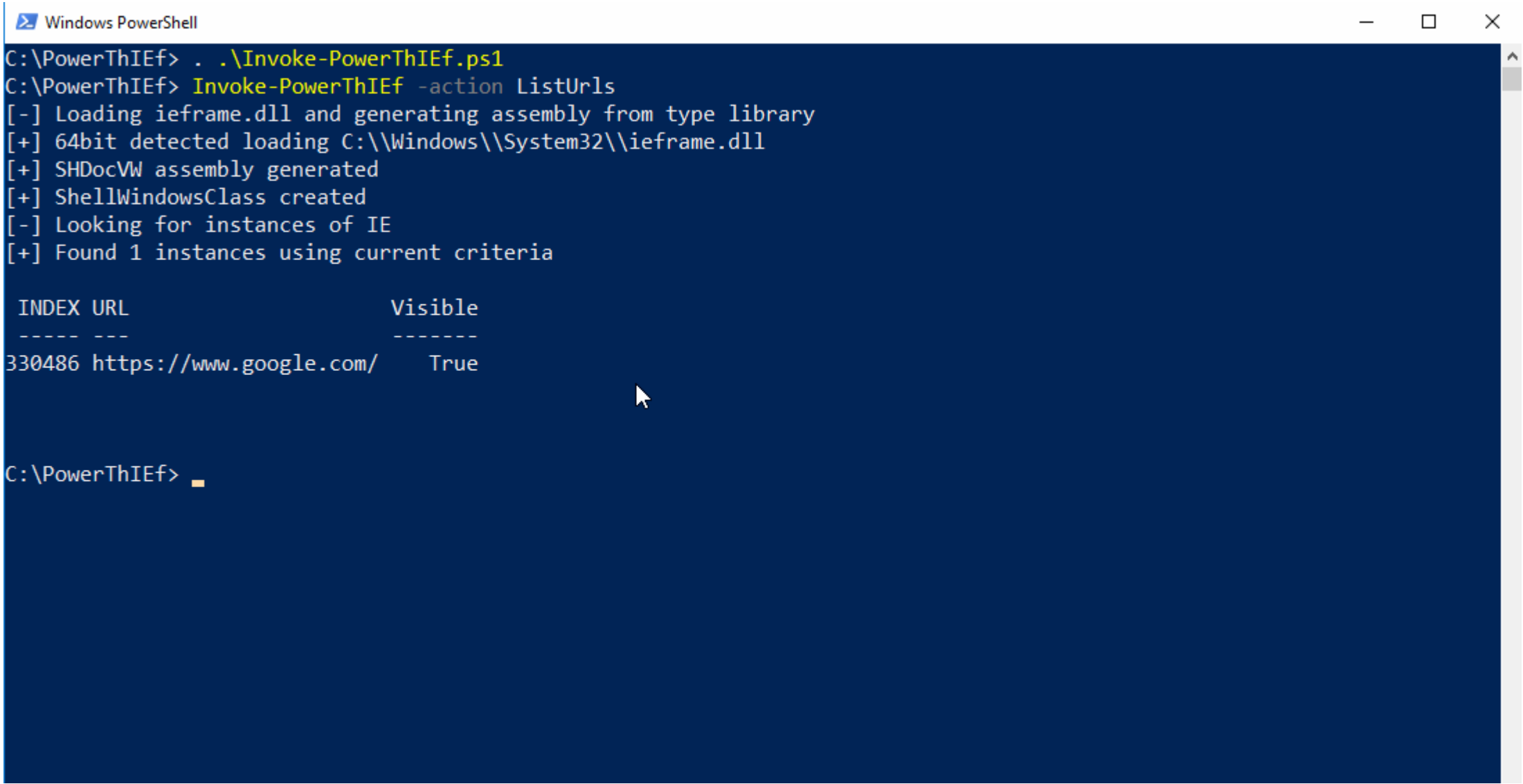
```
[SecurityPermissionAttribute(SecurityAction.Demand, Flags = SecurityPermissionFlag.UnmanagedCode)]
public AssemblyBuilder ConvertTypeLibToAssembly(
        object typeLib,
        string asmFileName,
        int flags,
        ITypeLibImporterNotifySink notifySink,
        byte[] publicKey,
        StrongNameKeyPair keyPair,
        bool unsafeInterfaces
)
```

Parameters

typeLib
Type: System.Object
The object that implements the **ITypeLib** interface.

asmFileName
Type: System.String
The file name of the resulting assembly.

Now can we register the event handler?

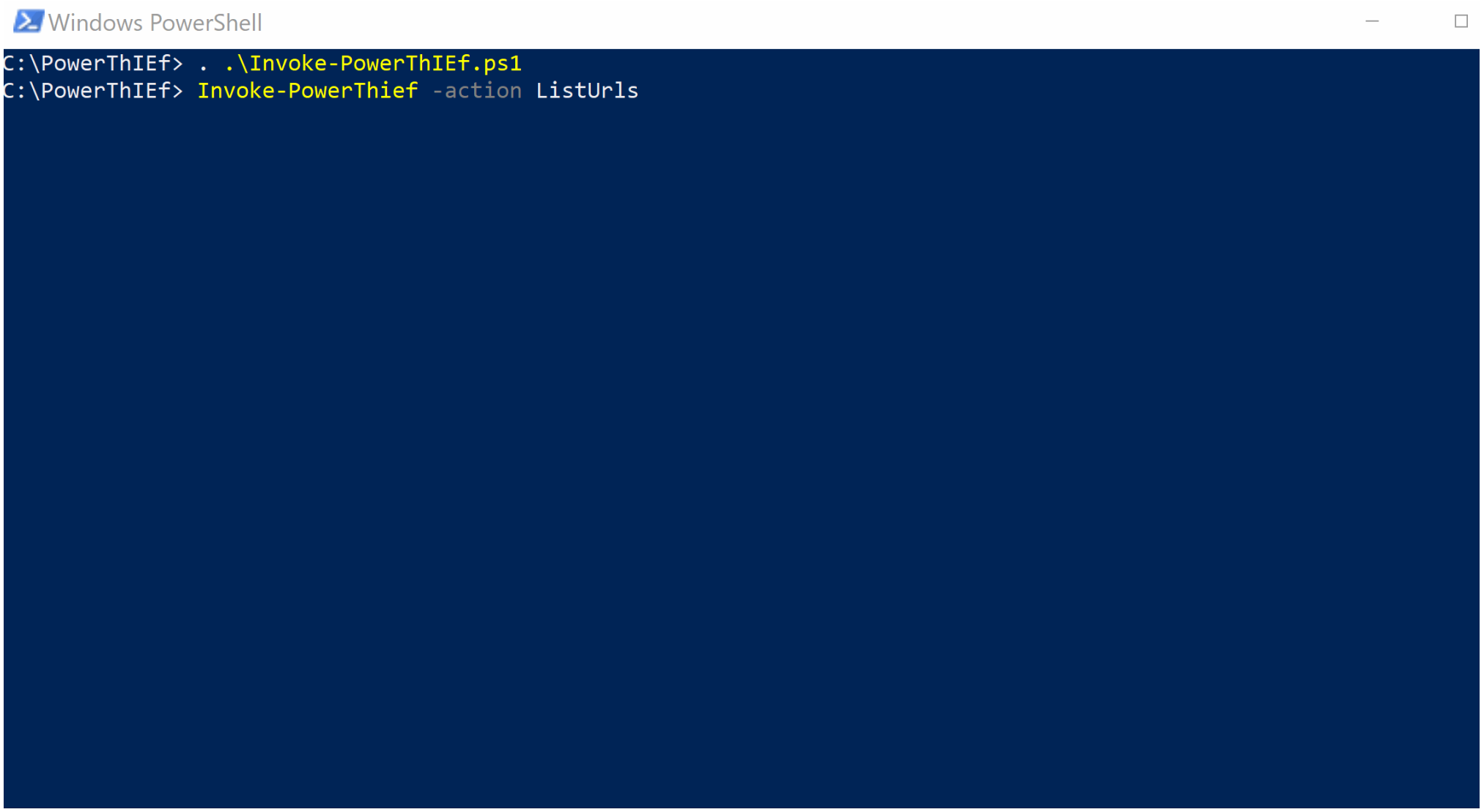


So we are all good then??

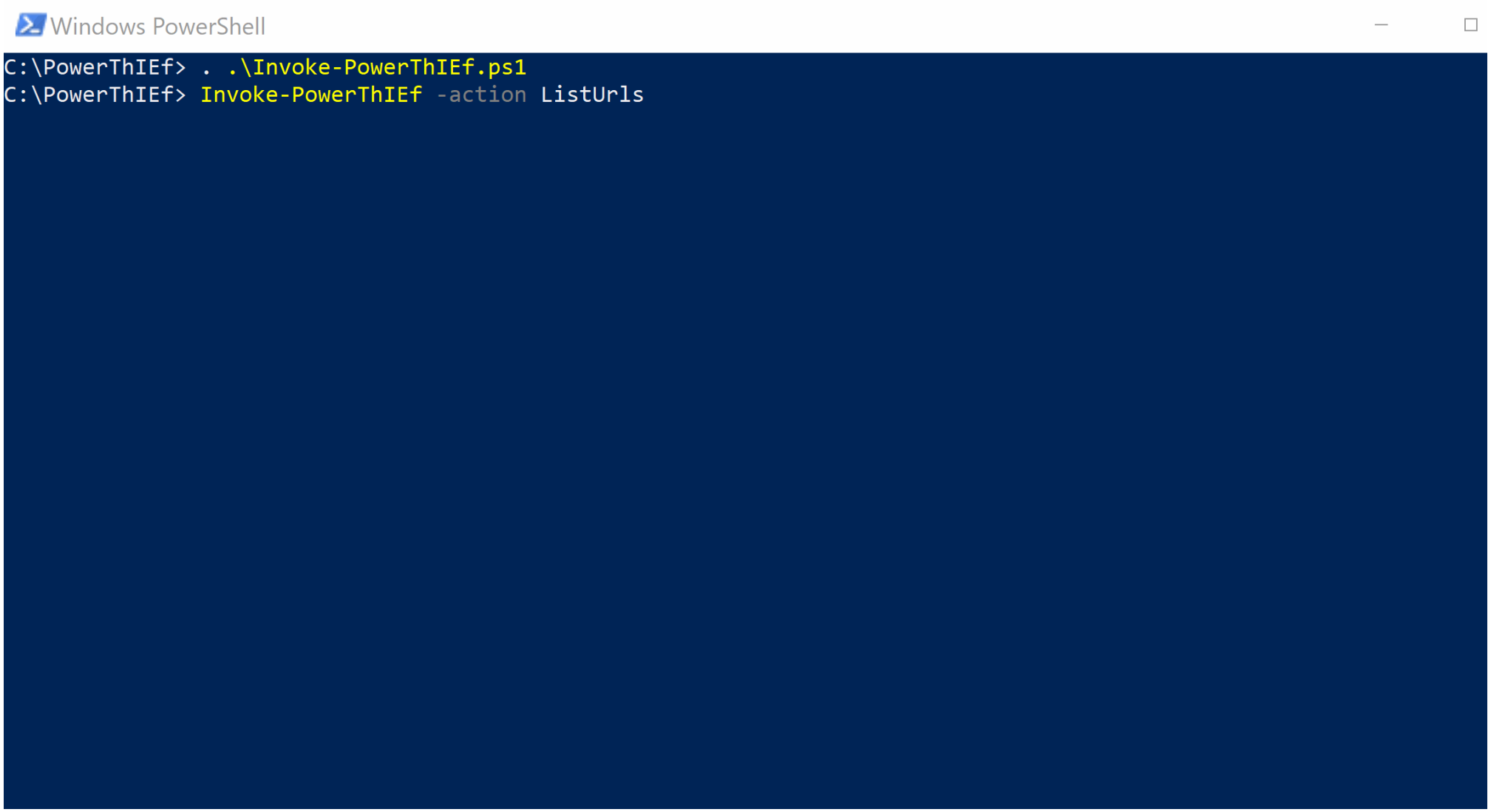
Well kinda in order to handle the events like *onSubmit* we have to use the IDispatch pattern. How this works would pretty much be a talk on its own.
Some good links to understand IDispatch and some ideas on how to handle in .Net (I will blog about this in the future)

- https://docs.microsoft.com/en-gb/previous-versions/windows/desktop/api/oaidl/nn-oaidl-idispatch
- https://www.codeproject.com/Articles/25769/Handling-HTML-Events-from-NET-using-C

Demo – Capturing typed creds in transit



Demo – Capturing Last Pass entered creds in transit



PowerThIEf

Supports

- IE 11
- Win 7-10
- .Net 4.0+
- Powershell 4.0

.Net version is mainly a limitation on the C# code embedded in the PowerShell.

Plan on rewriting to pure C# at some point which would allow support of .Net 2.0-3.5.

PowerThIEf

Roadmap

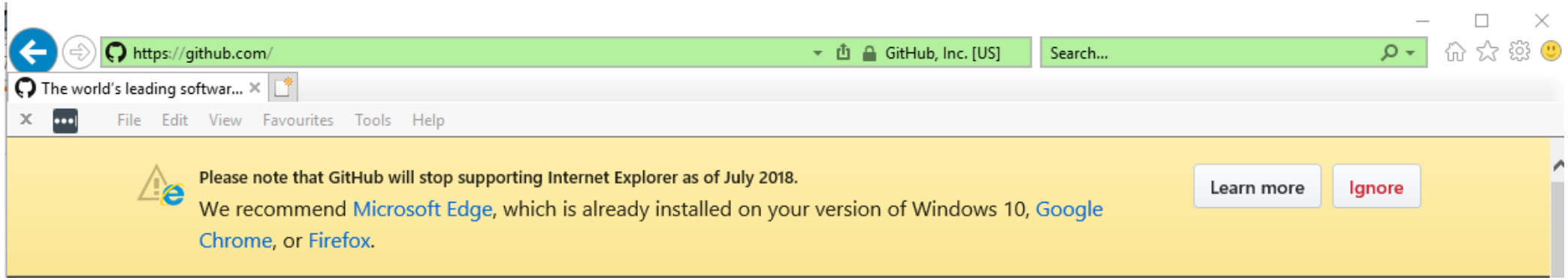
Planned functionality

- **Screenshot**: Screenshot all the browser windows
- **CookieThief**: Steal cookies (session cookies).
- **Refactor DOM event handling**: Developing Complex C# wrapped in Powershell is not ideal.
- **Pure C# Module**
- **Support for .Net 2.0-3.5**

NETTITUDE
A member of the Lloyd's Register group

Spend time looking into IE

And then start seeing deprecation messages, thanks!! (even though I wholly agree with the sentiment)



Creds are cool but what about MFA....

You're right fair point it's all about them cookies

- Yes you can just dump the process or.....
- You can come to my next talk, follow **@rbmaslen** and **@nettitude_labs** for details ☺

Availibility

Where I can get this amazing tool for my totes non-nefarious purpose?
Glad you asked

- GITHUB: https://github.com/nettitude/Invoke-PowerThief

NETTITUDE

A member of the Lloyd's Register group

Questions?

NETTITUDE

A member of the Lloyd's Register group

Thank You

@rbmaslen

@nettitude_labs