using System;

namespace SecTor_Demo_Basic
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello SecTor!!!");
            Console.ReadKey(true);
        }
    }
}
IN THIS TALK

What is .NET
C# for attacking
Attack demos
Detecting malicious C#
Detection demos
FYI…

PowerShell is **NOT** dead ;)

It just has such a heavy amount of defensive research
WHOAMI

- Lee Kagan
- Sr. Principal OffSec R&D @ Symantec
- Co-founder @ RedBlack Security
- Co-founder @ The C3X
- Twitter @InvokeThreatGuy
WHAT IS .NET?

- Software development framework from Microsoft (v1.0 in 2002)
- Common Language Runtime
  - Runtime environment
- Managed code (when targeting .NET)
WHAT IS .NET?

- Threading, type-safety, garbage collection managed for you
- Framework Class Library (FCL)
  - Object-oriented and reusable libraries
  - Assemblies in .NET speak
WHAT IS .NET?

- Supports a lot of development languages
  - C#, C++
  - JScript, VBScript
  - PowerShell, IronPython
  - …and soooo much more!
WHAT IS .NET?

- Not just for Windows
- Mono, Xamarin and .NET Core allow for targeting non-Windows platforms (.NET Standard)
WHAT IS .NET?

- Lots of terms and acronyms to know (yuck)
- CLR – Common Language Runtime
- CLI – Common Language Infrastructure
- MSIL / CIL – Microsoft Intermediate Language / Common Intermediate Language
WHAT IS .NET?

- GAC – Global Assembly Cache
- JIT compiler – Just In Time
- FCL – Framework Class Library
- You guessed it…there are more!
WHAT IS .NET

https://prabirchoudhury.wordpress.com/2008/12/15/common-language-runtime-clr/clr_image21/
BASIC C# CODE EXAMPLE

Using statement
namespace SecTor_Demo_Basic
{

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello SecTor!!!");
        Console.ReadKey(true);
    }

    The compiled program

    Hello SecTor!!!
PowerShell has been defensively covered very well over the years (still works great though)
Malicious C# detections are not widely explored territory (just yet)
Learning entry point to writing in C# is very low
Porting PowerShell tools to C# is quite simple

Like PowerShell, staying very native to Windows
  - Still living off the land 😊

Can still access Win32 fun stuff (VirtualAlloc, CreateProcess)
  - Commonly called **pinvoking**
OFFENSIVE C# TOOLING

- Amazing and rapidly growing public set of C# tools
  - GhostPack – Sorta ports of Mimikatz (SafetyKatz, Rubeus), SharpView, SeatBelt
  - Covenant – An entire C2 framework
  - Cobalt Strike’s execute-assembly
    - feature that lets you load a .NET assembly in-memory
A lot of ways to run C# code

- In PowerShell via AddType
- Wrapped and LOLbin’d as a .csproj with MSBuild
- Can dynamically compile raw C# code in memory
Going to demonstrate...

- Malicious .csproj file executed with msbuild
- C# process injection using QueueUserApc
- C# embedded in PowerShell via AddType
- Cobalt Strike’s execute-assembly
DETECTING MALICIOUS C#

- *No silver bullets*
- AMSI in .NET 4.8
- Managed code running in unmanaged process
- Loading of the CLR and dependent assemblies into a process (similar to above)
DETECTING MALICIOUS C#

- ETW (Event Tracing for Windows)
- Can still hook API calls like anything else
- C# can be “reverse engineered” very easily
- Can decompile back to source code with minimal effort
DETECTING MALICIOUS C# - DEMO

- AMSI detecting malicious Assembly.Load() in .NET 4.8
- Detecting ImageLoads via SilkETW
- Detecting CLR reflection via Get-ClrReflection
- Process injection using Get-InjectedThread
- "Reverse engineering" .NET PE via ILSpy
KEY TAKEAWAYS

For offensive folks:

- Shift to C# from PowerShell for ‘stealth’
- Can operate in-memory
- Bring your own interpreter (check out Silent Trinity)
- .NET APIs are super rich
- Interoperability is huge
KEY TAKEAWAYS

For defensive folks:

- The usual suspects (process creation, image loads, CRT)
- Application baseline and AWL is still really important
- The list/combo of Windows APIs should still be hooked
- AMSI can help a lot (4.8 may take some time)
- Understanding .NET operations artifacts
THANK YOU SO MUCH SECTOR!!! 😊