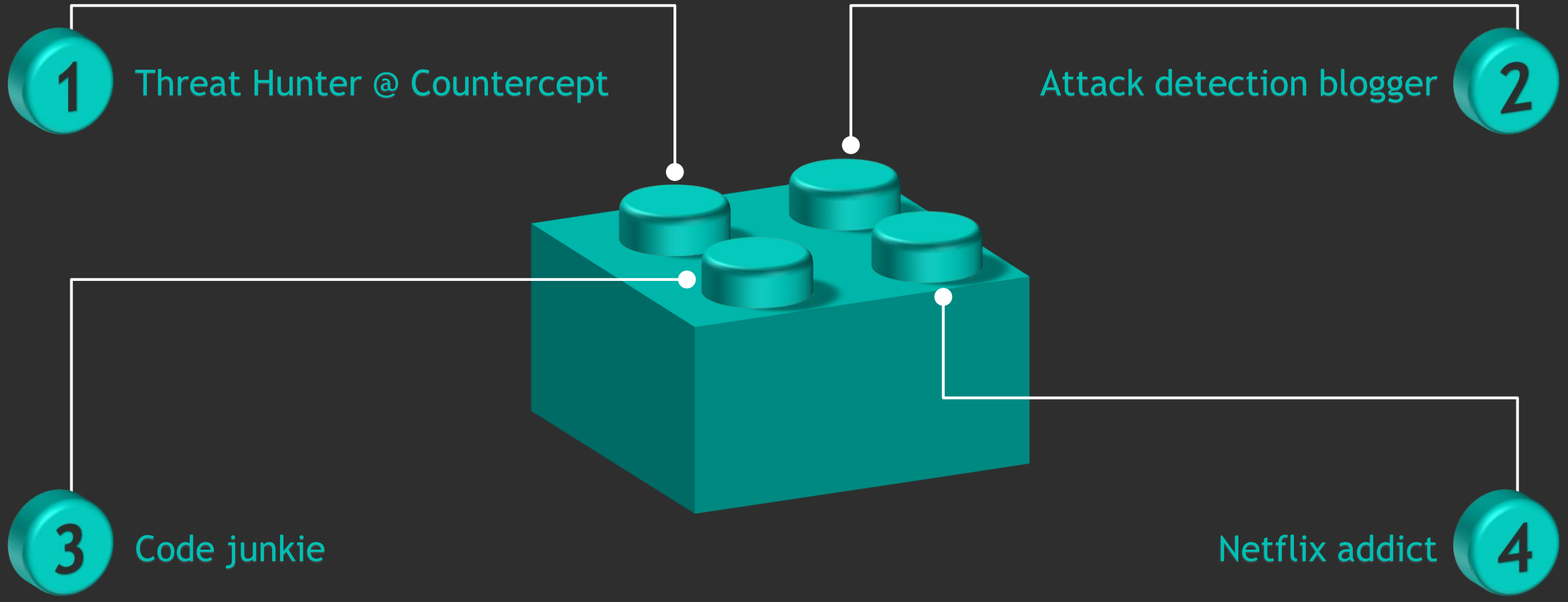


COUNTERCEPT

YOU ARE NOT HIDING FROM ME .NET!

COUNTERCEPT

FOUR THINGS ABOUT MYSELF



THE AGENDA

COUNTERCEPT

Why .NET ?



Detect



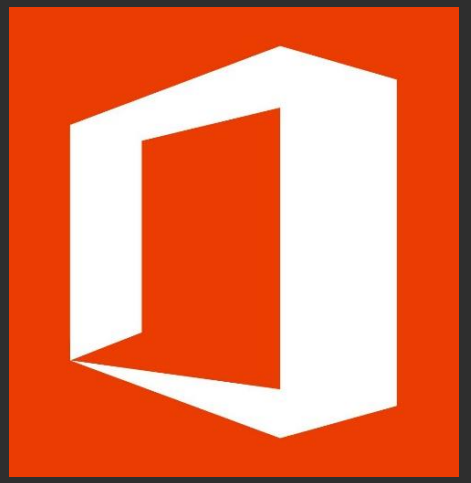
IN THE NOT SO DISTANT PAST



PowerShell

VBScript





Office Macros

POWERSHELL, A HOT FAVOURITE



Powerful

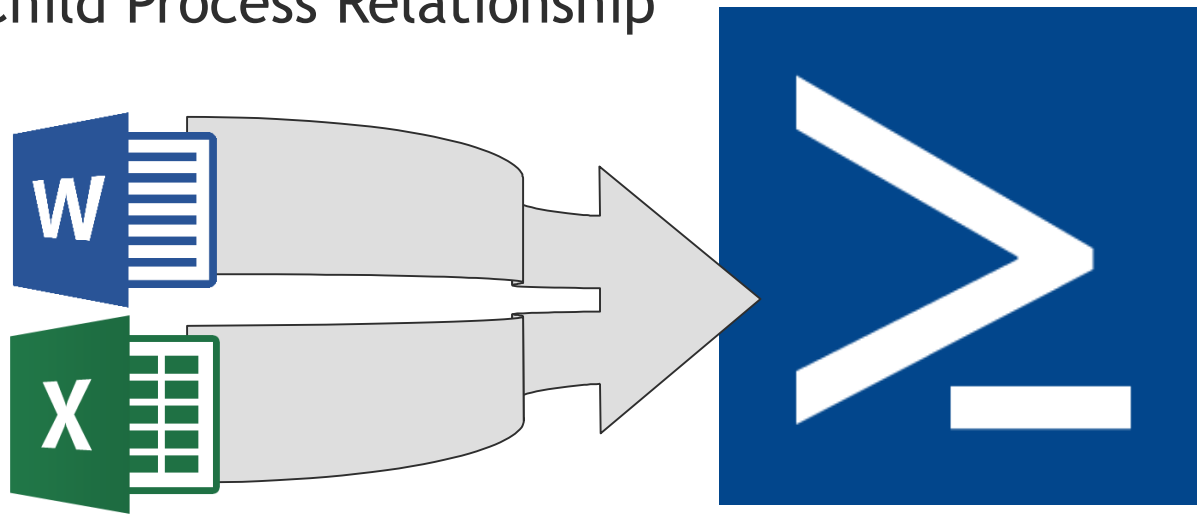
Load shellcode into memory

Call upon .NET API

Call upon native API

DEFENCES ARE GETTING BETTER

Parent-Child Process Relationship



Command Line Arguments Logging

CommandLine	powershell write host “This is an evil command”



EDR AGENTS

DEFENCES ARE GETTING BETTER

AMSI assisting Anti-Virus with script-based detection



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

PowerShell Script Block Logging to aid with detection





COUNTERCEPT

More opportunities to detect bad PowerShell

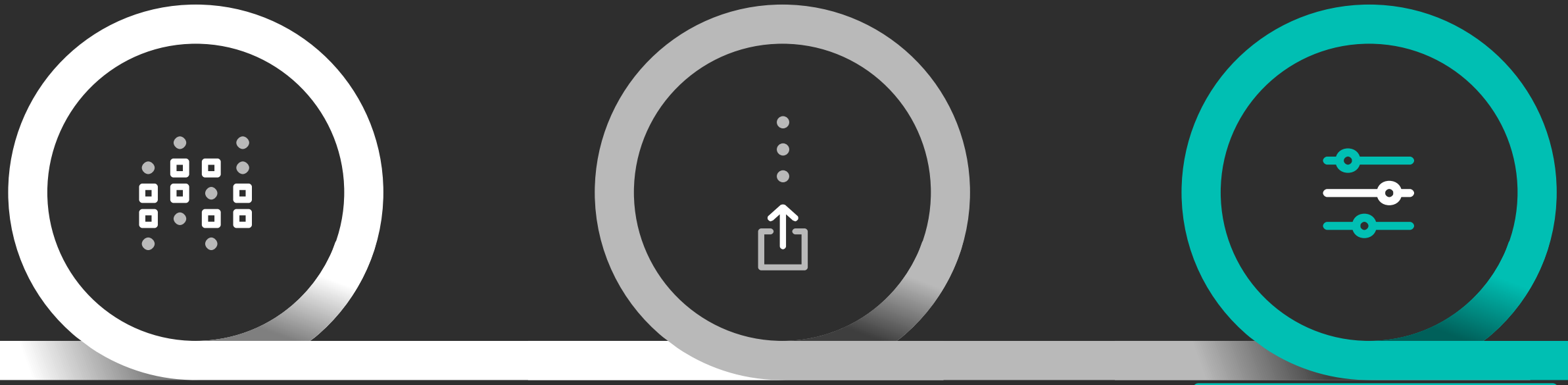


COUNTERCEPT



Invoke .NET directly instead of via PowerShell

WHY .NET THOUGH?



Powerful functions

Installed by default

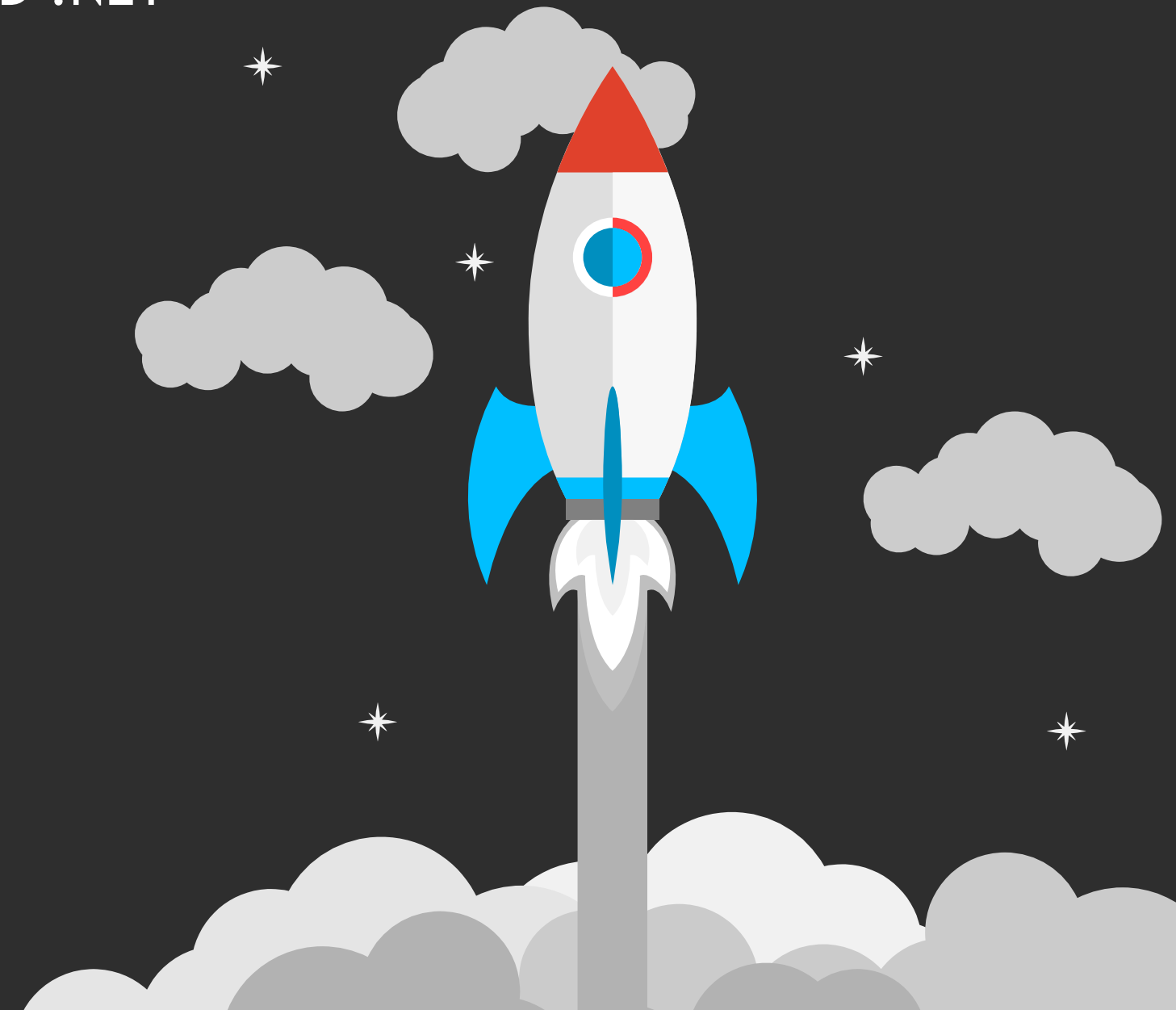
Lack of telemetry

LET'S COMPARE POWERSHELL AND .NET

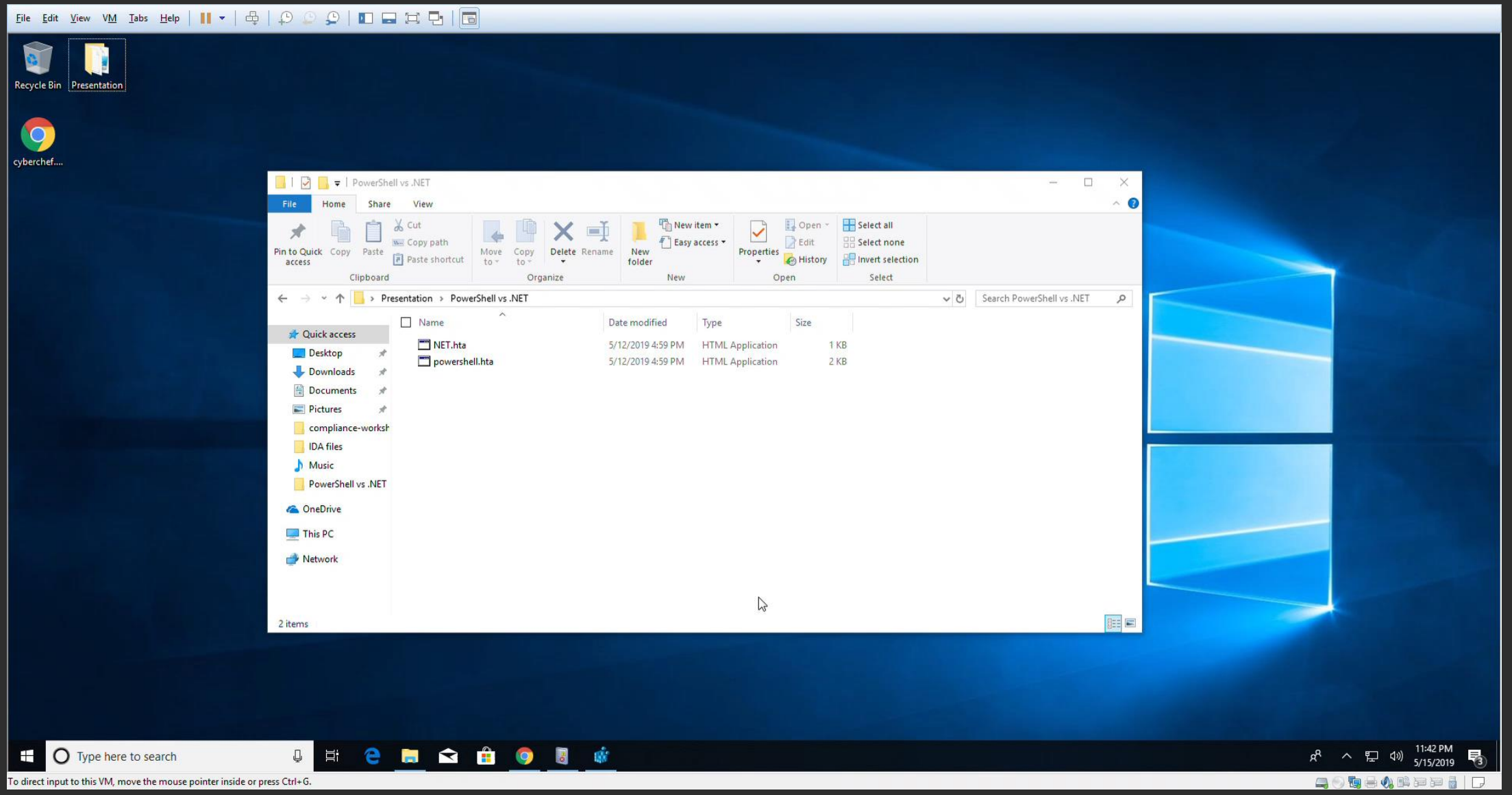
COUNTERCEPT

Write a registry key through a .NET API

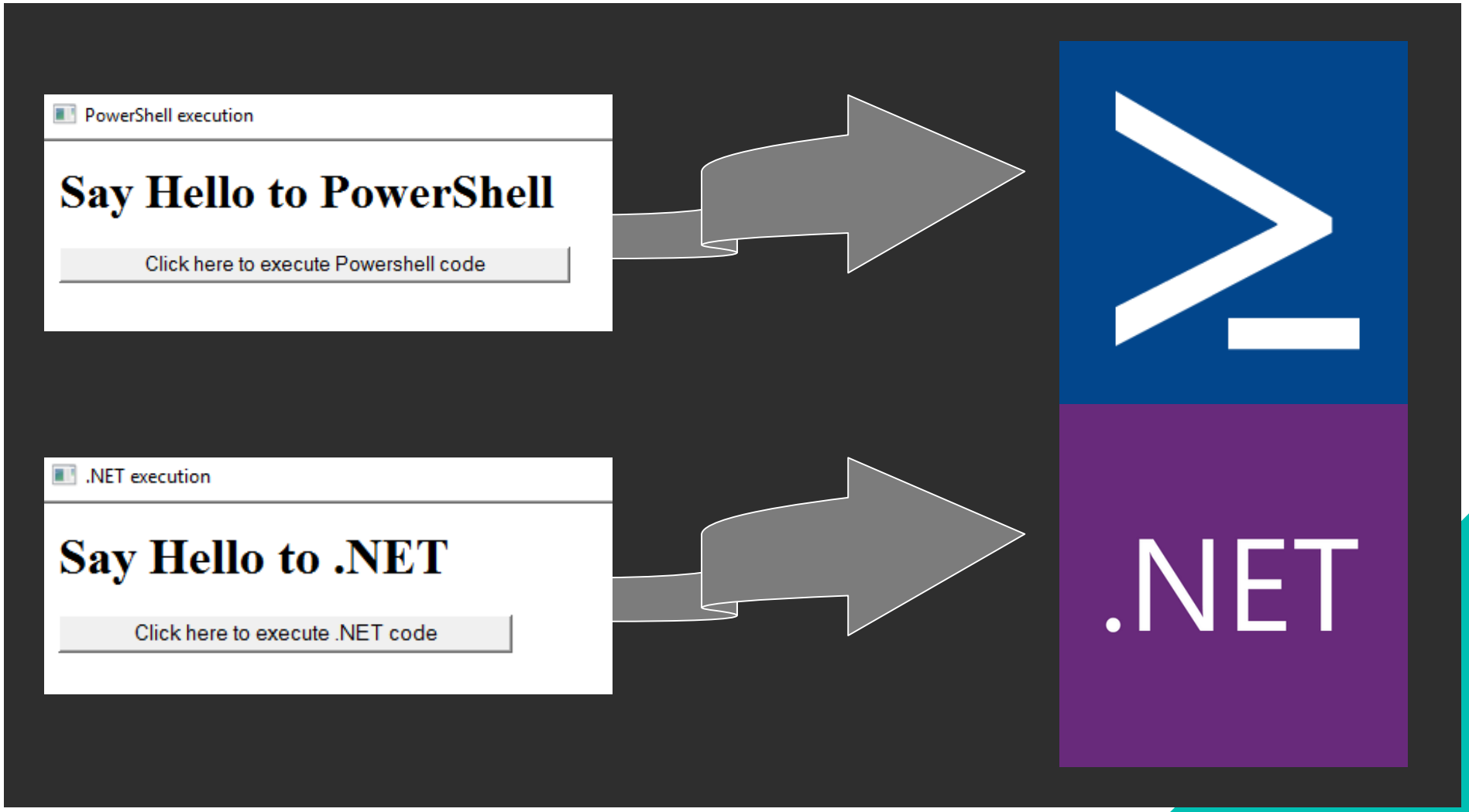
Pop a Message box with a native API



POWERSHELL VS .NET DEMO



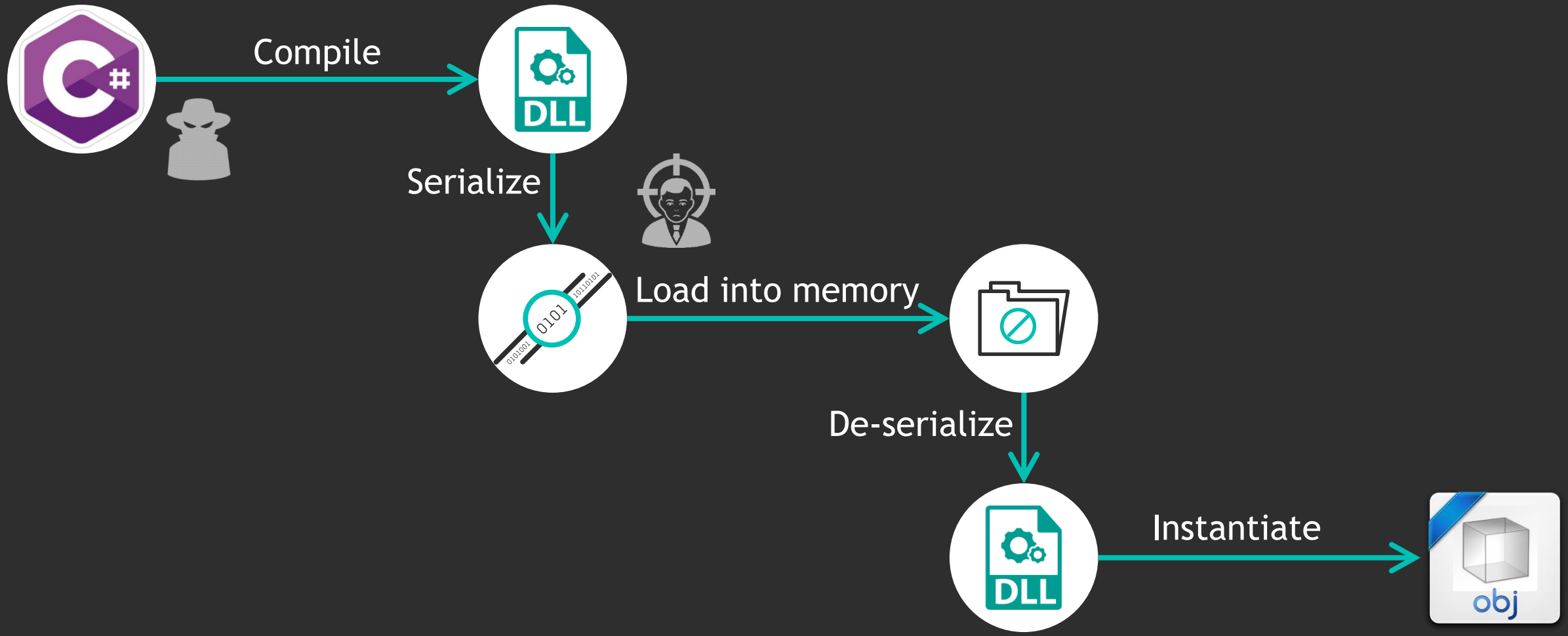
POWERSHELL VS .NET



COUNTERCEPT

In-memory assembly loading

IN MEMORY .NET ASSEMBLY LOADING



A FUN FACT

```
using MyAssembly;

Program newProgram = new Program();
newProgram.main();
```

This does exactly the same thing

COUNTERCEPT

WHAT CAN THE LOADED OBJECT DO?



Load shellcode into memory

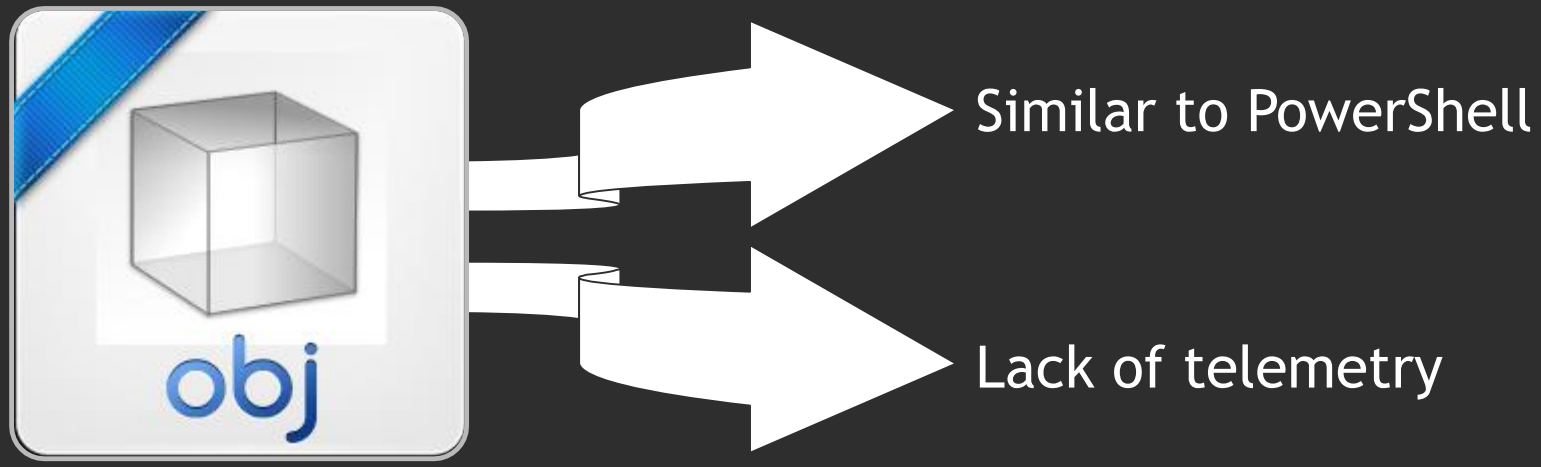
Call upon .NET API

Call upon native API

Similar to PowerShell

COUNTERCEPT

MID-POINT CHECK



Challenge: Can we detect this?

THE AGENDA

COUNTERCEPT

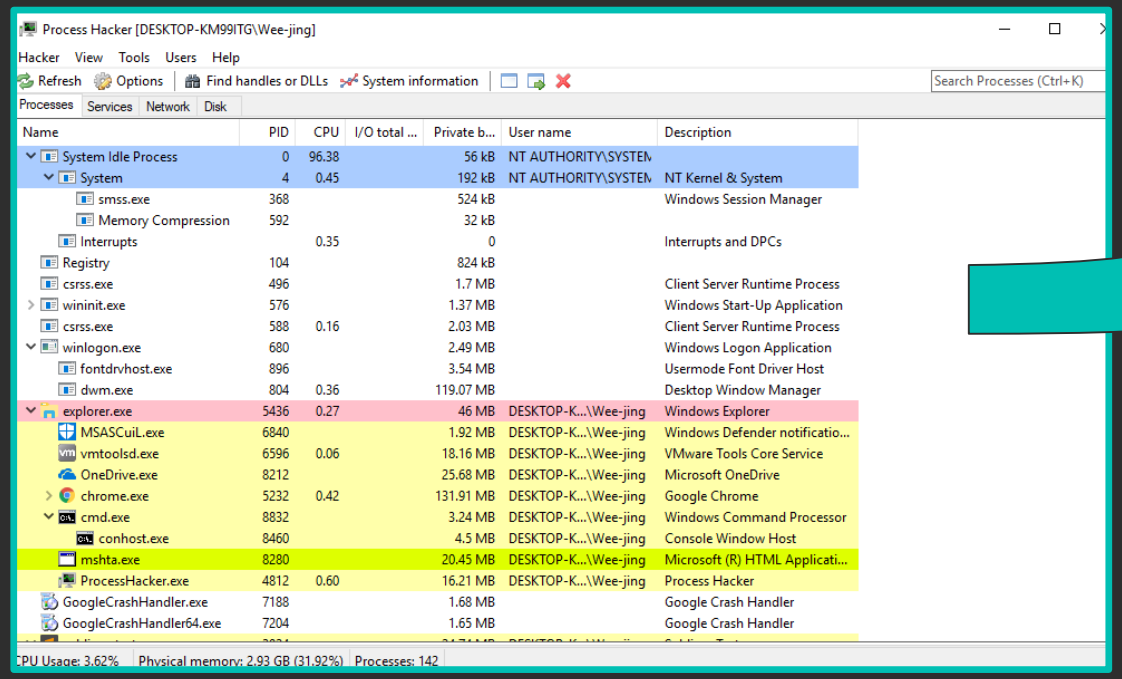
Why .NET ?



Detect



..... WITH PROCESS HACKER



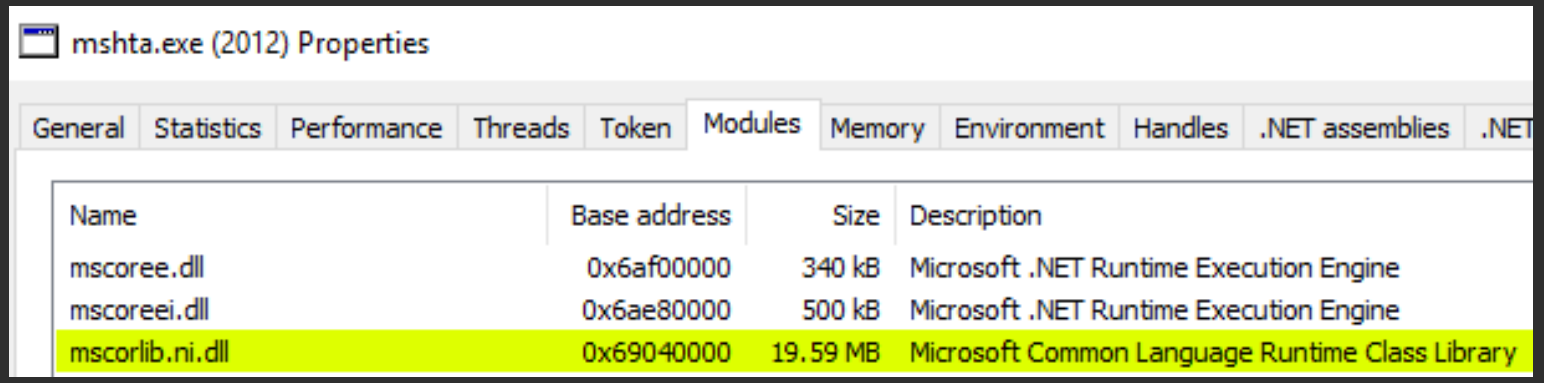
Process Hacker

Analyze



Mshta.exe

DETECTING .NET LOADED DLLS



Loading of .NET runtime DLLs can be observed

COUNTERCEPT

MSHTA typically only runs HTML or JavaScript code!

COUNTERCEPT

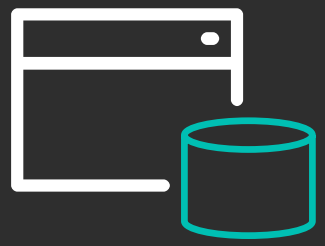
HOLDS TRUE FOR OTHER BINARIES



WHAT IF A BINARY RELATED TO .NET WAS USED



Msbuild.exe



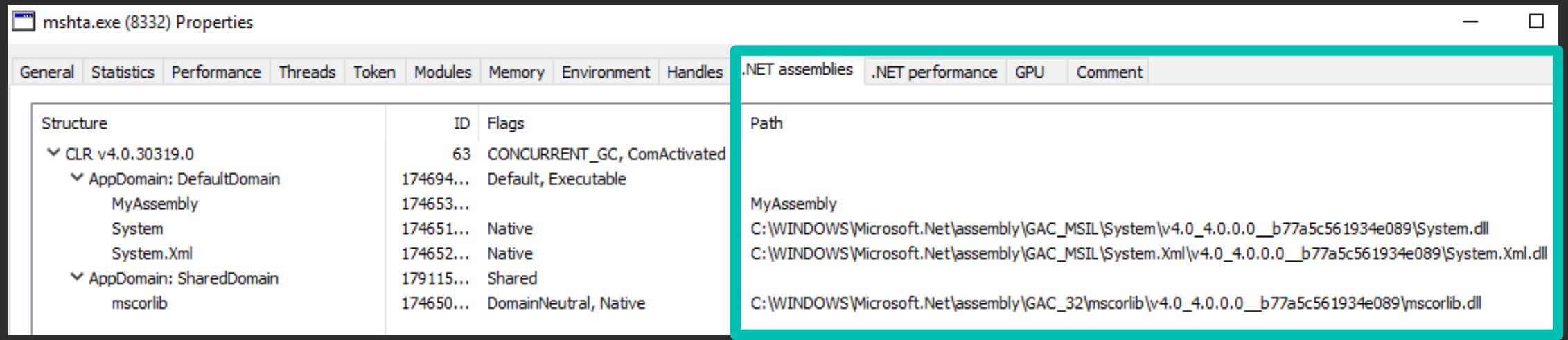
3rd Party Application

Not uncommon to have .NET runtime DLL

COUNTERCEPT

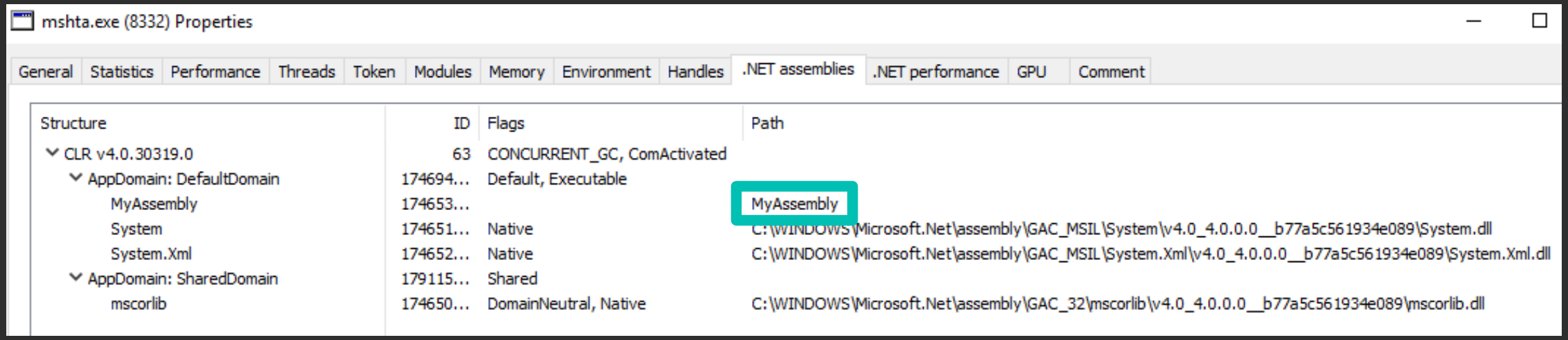
And the answer lies deep within Process Hacker

.NET ASSEMBLIES



Events of assembly loads

.NET ASSEMBLIES



Lack of a path indicates potential in-memory assembly loads

COUNTERCEPT

How did Process Hacker achieve this?

DEEP WITHIN PROCESS HACKER

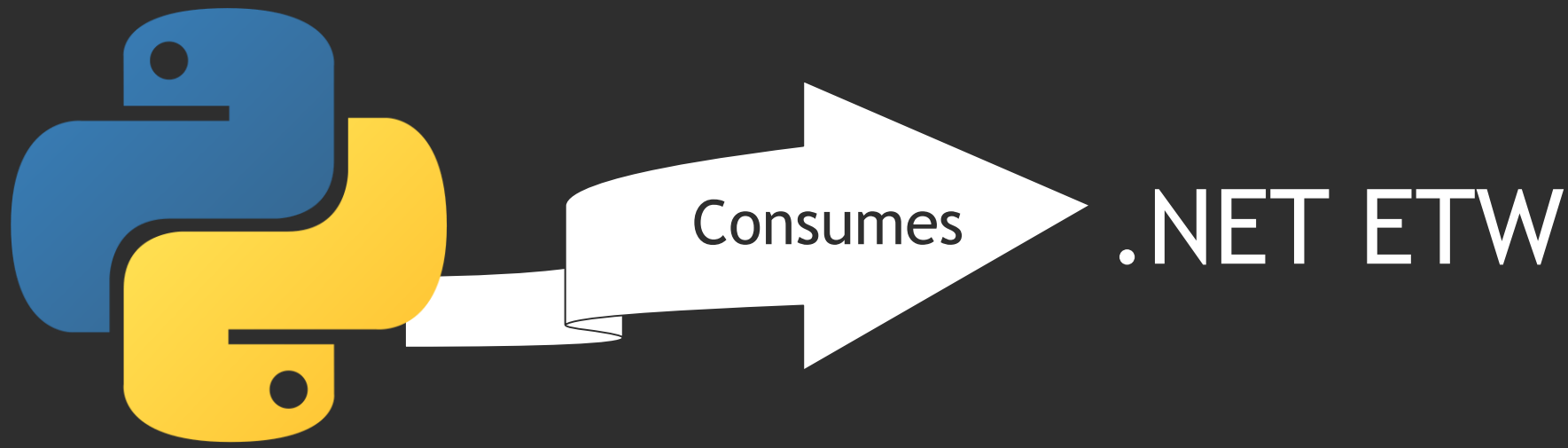
COUNTERCEPT

Microsoft-Windows-DotNETRuntime

Microsoft-Windows-DotNETRuntimeRundown

A set of .NET ETW providers

PROOF-OF-CONCEPT



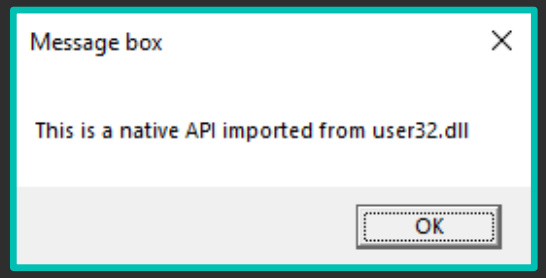
LET'S TRY TO DETECT MY ATTACK



Indicators for in-memory assembly load



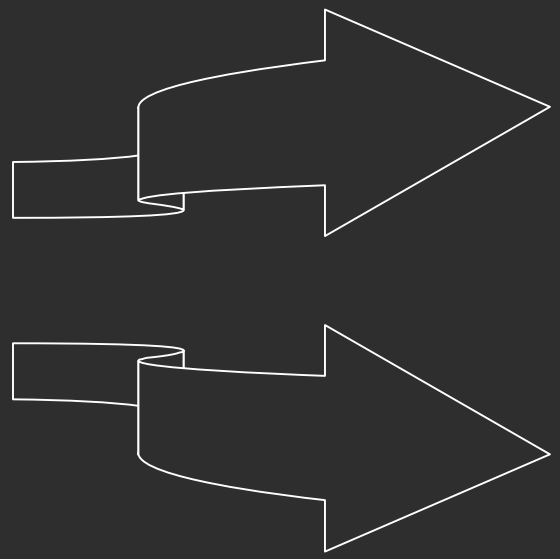
Indicators for .NET API related to registry creation



Indicators for invoking of native API

IN-MEMORY ASSEMBLY LOAD

Events related to in-memory assembly load



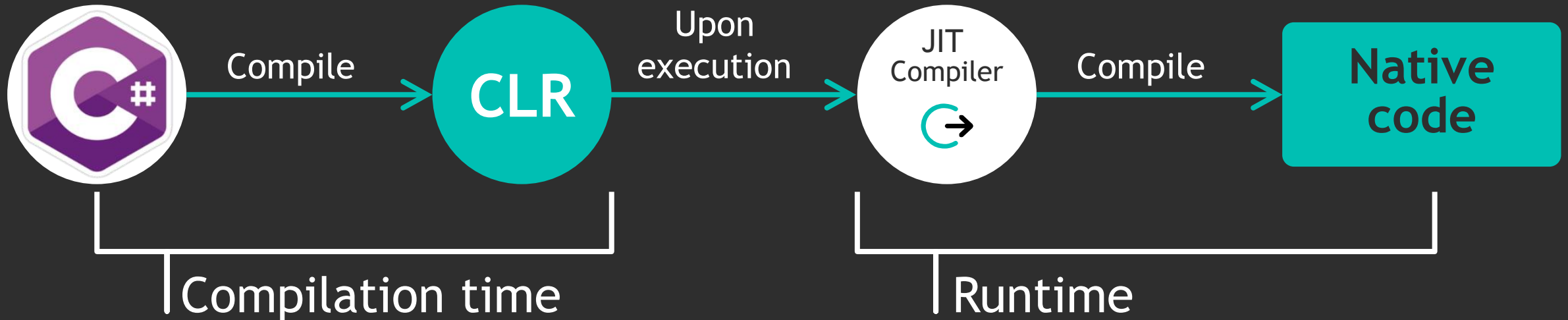


Loading of .NET assemblies



Just In Time compilation

.NET CODE COMPILATION ARCHITECTURE



Events generated whenever a .NET method is first utilized

IN-MEMORY ASSEMBLY LOAD INDICATORS



Loading of .NET assemblies





Just In Time compilation

IN-MEMORY ASSEMBLY LOAD INDICATORS

```
12068, DomainModuleLoad_V1, 0x2701978, Manifest , MyAssembly,
12068, MethodJittingStarted, 0x6145FF4, MyAssembly.Program, .ctor
12068, MethodJittingStarted, 0x6145FF4, MyAssembly.Program, main
```

Detect execution of the MyAssembly constructor

REMEMBER THIS?

```
using MyAssembly;

Program newProgram = new Program();
newProgram.main();
```

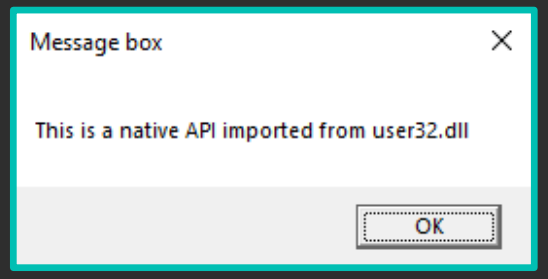
In-memory loading of assembly attempts to replicate the above behavior

DETECTION SUMMARY



Indicators for in-memory assembly load

Indicators for .NET API related to registry creation



Indicators for invoking of native API

COUNTERCEPT



Just In Time compilation

Can we use this?

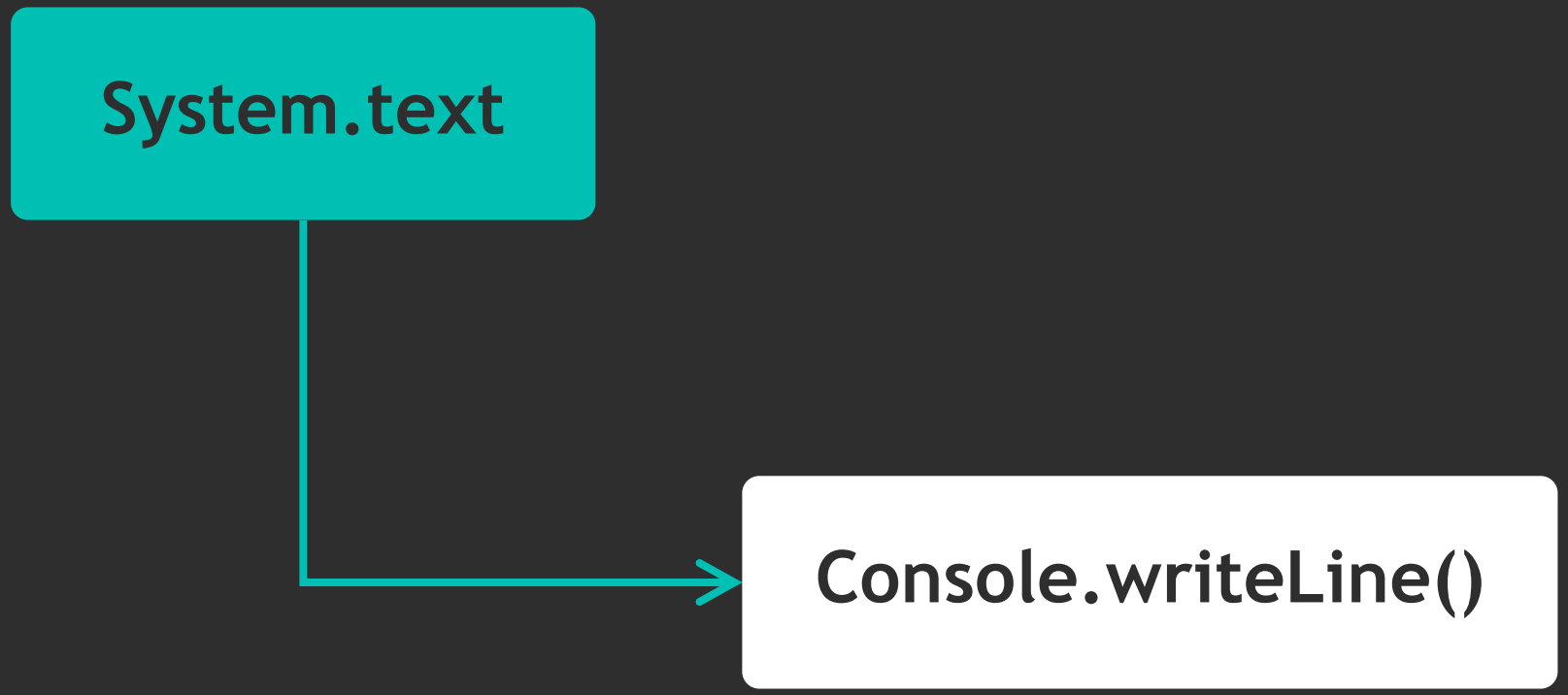


UNFORTUNATELY....

COUNTERCEPT

JIT compilation doesn't occur for native .NET assemblies

NATIVE .NET ASSEMBLIES?



WHY THOUGH?



Native Image Generator (NGEN) compiles .NET assemblies to native images, and caches them

COUNTERCEPT

WHY THOUGH?



JIT compilation would not occur

DETECTION SUMMARY



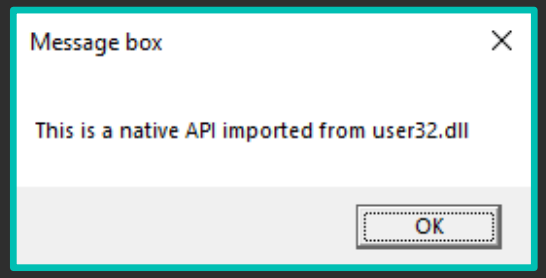


Indicators for in-memory assembly load





Indicators for .NET API related to registry creation



Indicators for invoking of native API



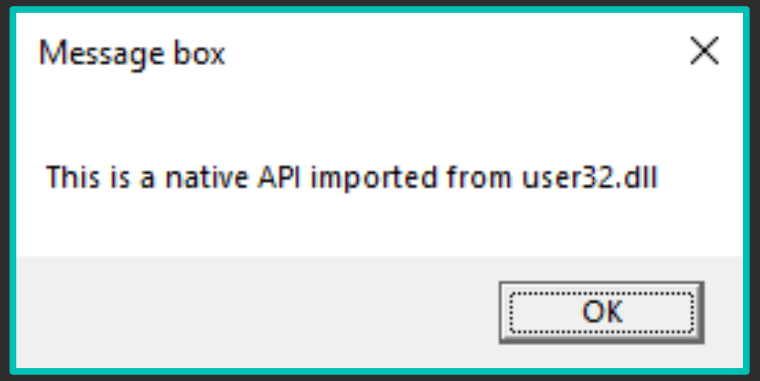
COUNTERCEPT

Interop events

COUNTERCEPT

Events generated during calls made to Window's native API

COUNTERCEPT



Native function imported from User32.dll

```
2044, ILStubGenerated, None, MyAssembly.Program,
MessageBox
```

Detected a call towards MessageBox

COUNTERCEPT

THIS IS REALLY USEFUL



Logging of keystrokes



Credential extraction from memory



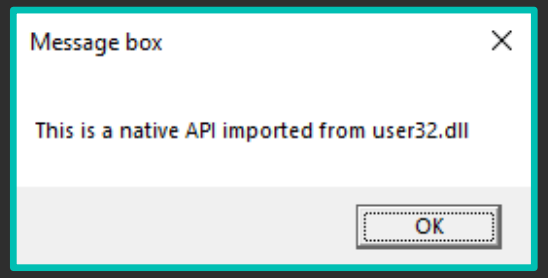
Other malicious activities

DETECTION SUMMARY



Indicators for in-memory assembly load

Indicators for .NET API related to registry creation



Indicators for invoking of native API



REAL WORLD EXAMPLE, SILENTRINITY

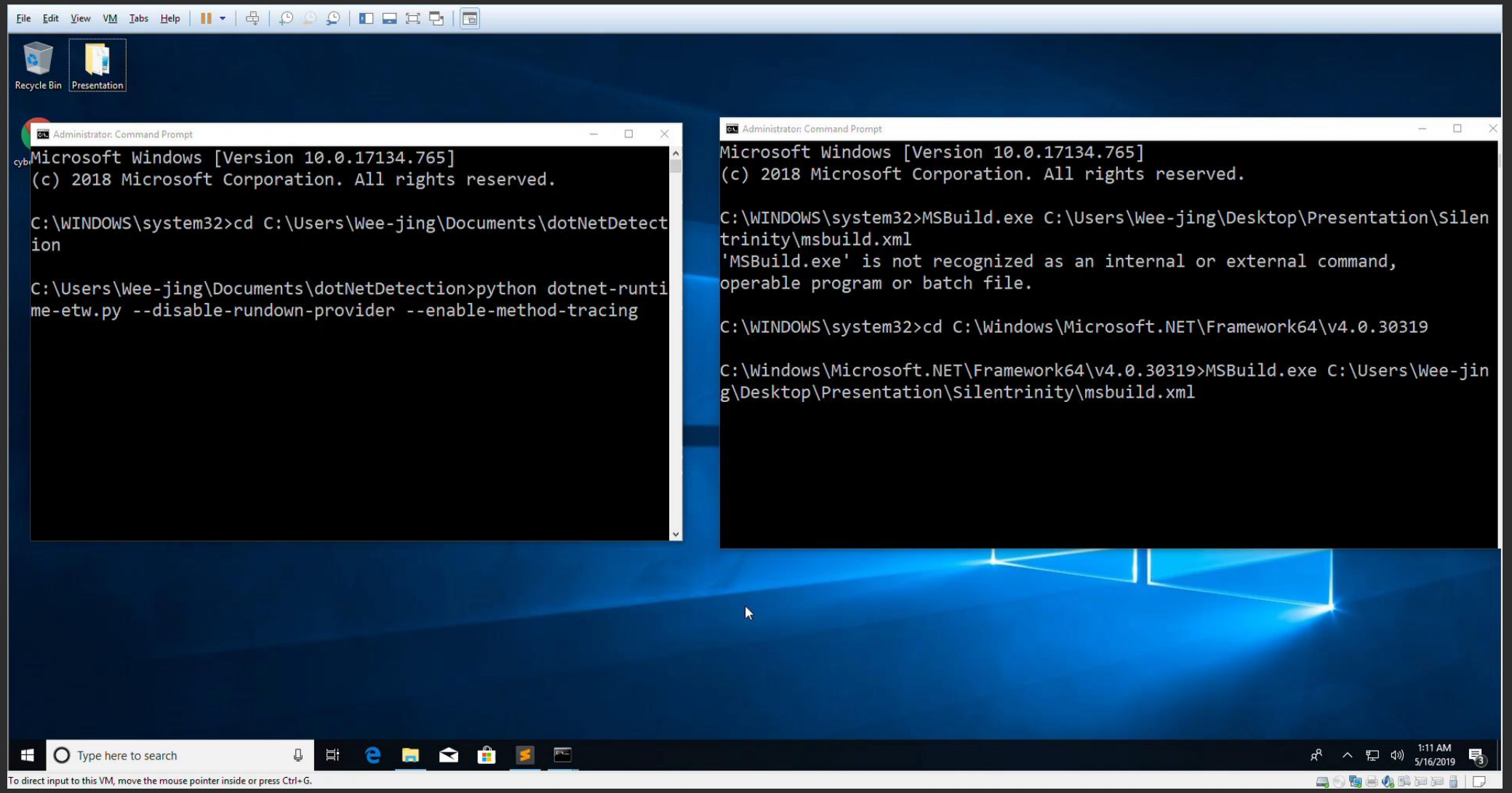
COUNTERCEPT

Launch a .NET assembly

Launch SafetyKatz, a credential extraction tool



SILENTRINITY DEMO



.NET TELEMETRIES



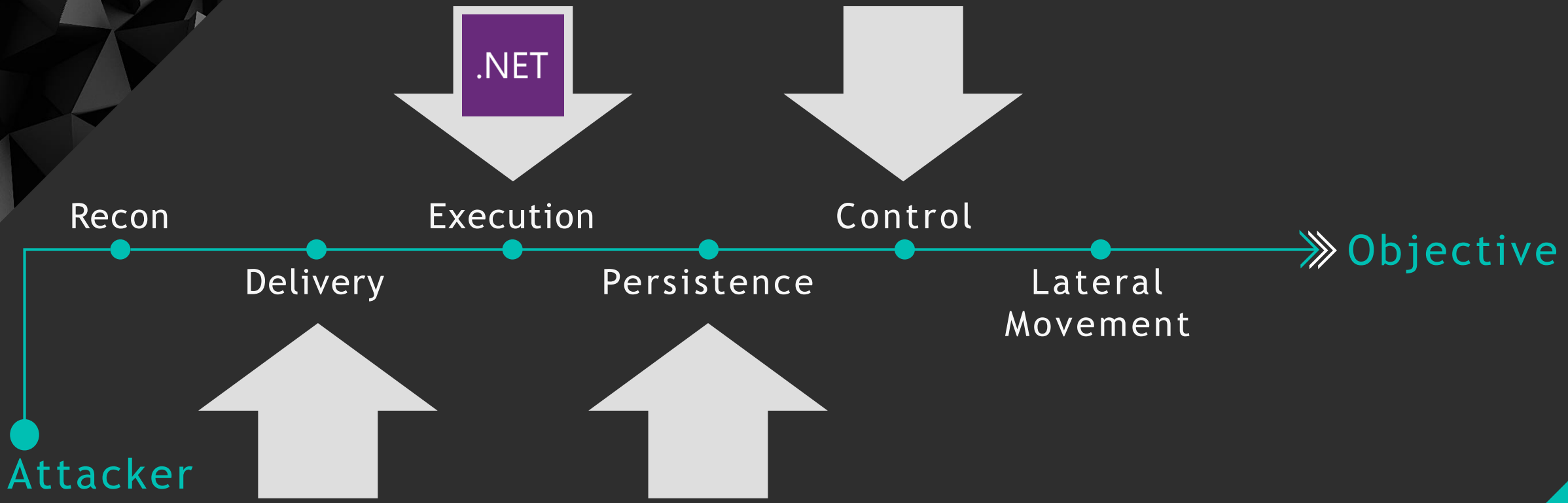
.NET runtime DLLS



.NET ETW events

HOW ABOUT OTHER TELEMETRY?

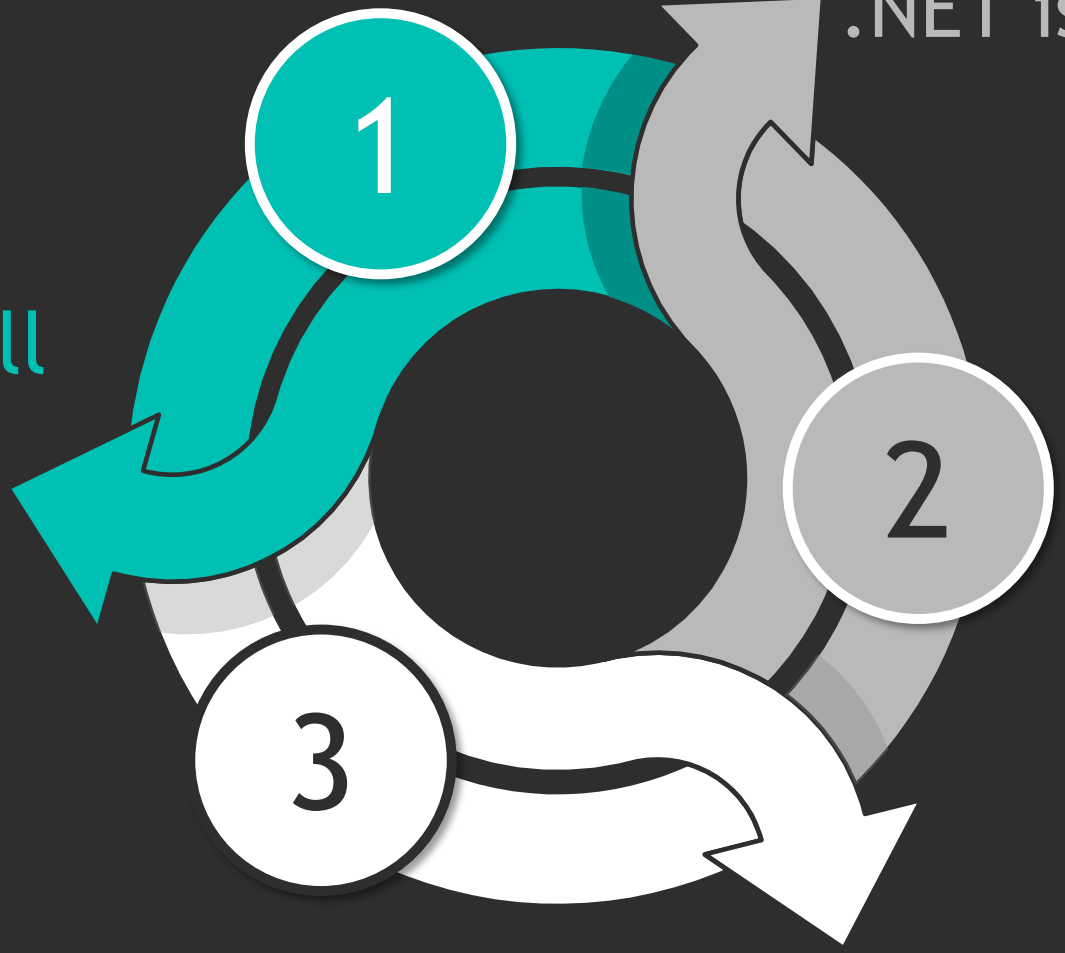
COUNTERCEPT



TO WRAP IT UP

.NET isn’t that invisible

PowerShell, still deadly but…..



Try it yourself!