Bundesamt
für Sicherheit in der
Informationstechnik

Analyse der Telemetriekomponente in Windows 10

Konfigurations- und Protokollierungsempfehlung

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-0
E-Mail: bsi@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Microsoft Telemetrie (im Folgenden mit „Telemetrie“ abgekürzt) ist eine Komponente in Windows 10, die für die automatische Erhebung und Übertragung von Daten an eine von Microsoft betriebene Backend-Infrastruktur (im Folgenden mit „Telemetrie-Backend“ abgekürzt) verantwortlich ist. Bei den erhobenen Daten handelt es sich um unterschiedliche Daten wie z. B.: Daten über die Nutzung des Computers unter Windows 10 und der an ihn angeschlossenen Geräte, Daten über die Performance des Systems, Daten, die bei Fehlern, wie Programm- oder Systemabstürzen erhoben werden, sowie Daten des Windows Defenders und des Malicious Software Removal Tools (MSRT).

Die erhobenen Daten werden von Microsoft auch als Diagnose-Daten („diagnostic data“) bezeichnet und weisen auf den Grund hin, den Microsoft für Erhebung nennt: Dem Benutzer solle eine „Stimme“ [ms_dvc] gegeben werden, damit Microsoft die Betriebssystemqualität, die Benutzererfahrung und die Sicherheit kontinuierlich verbessern kann. Die Sichtweise von Datenschützern und vielen Organisationen auf die in Windows 10 eingebaute Telemetrie ist dagegen von berechtigter Vorsicht und Skepsis gegenüber dieser Aussage von Microsoft geprägt. Vorsicht und Skepsis beziehen sich dabei auf die Art und den Umfang der erhobenen Daten, die Sicherheit ihrer Übertragung sowie ihre Speicherung und Verarbeitung und deren Sicherheit im Telemetrie-Backend.

Das vorliegende Dokument gibt daher zunächst einen Überblick über Architektur und Funktionsweise der Telemetrie und ihrer Komponenten, beschreibt dann die Konfigurations- und Überwachungsmöglichkeiten sowie deren Wirksamkeit und gibt schließlich Empfehlungen zur sicheren Konfiguration und Deaktivierung von Telemetrie in Windows 10. Eine einfache Überwachung des Dienstes (beispielsweise über die `Ereignisanzeige`) ist nicht möglich, weswegen im ein Vorgehen zur Analyse von Telemetrie-Aktivität beschrieben ist.

Das vorliegende Dokument ist im Rahmen des Projekts *Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10 (SiSyPHuS Win10)* als Unterarbeitspaket einer allgemeinen Analyse der Windows 10 Telemetrie-Funktionalität entstanden.

1.1 Datenerhebung und Event Tracing unter Windows

Telemetrie in Windows 10 nutzt die Funktionen von Event Tracing for Windows, um Daten zu erheben. Ein Verständnis der Funktionsweise von Telemetrie in Windows 10 setzt ein (grobes) Verständnis der Funktionsweise von Windows Event Tracing voraus. Daher gibt dieser Abschnitt einen kompakten Überblick über die Komponenten des Windows Event Tracing und ihr Zusammenspiel.

Event Tracing for Windows (im Folgenden abgekürzt mit „ETW“) ist eine Protokollierungsfunktion von Windows 10, die Kernel-Buffer nutzt, um Daten aufzuzeichnen und diese in Form von Echtzeitdaten oder über Logdateien (wie z. B. das Ereignisprotokoll) zur weiteren Verarbeitung bereitzustellen. ETW wurde von Microsoft im Laufe der Jahre von Windows 2000 bis zu Windows 10 zu einem umfangreichen Framework zur Aufzeichnung und Bereitstellung von Ereignissen (Events) entwickelt, mit dem große Mengen an Daten effizient erhoben werden können. ETW wird auch innerhalb der verschiedenen Versionen von Windows 10 kontinuierlich weiter entwickelt. So kommen mit jeder neuen Version zusätzliche sogenannte Provider hinzu, die für die Erhebung und Bereitstellung von bestimmten Daten zuständig sind. In diesem Dokument kann daher nur eine generische Übersicht über ETW unter Windows gegeben werden mit dem Ziel, die für die Telemetrie relevanten ETW-Komponenten in ihrer grundsätzlichen Architektur und Funktion zu verstehen. Für weiterführende Information sei auf das entsprechende MSDN-Dokument verwiesen. [ms_etw]

ETW besteht aus verschiedenen Komponenten mit unterschiedlicher Funktion, deren Zusammenwirken in Abbildung 1 [ms_etw] dargestellt und im Folgenden beschrieben ist.

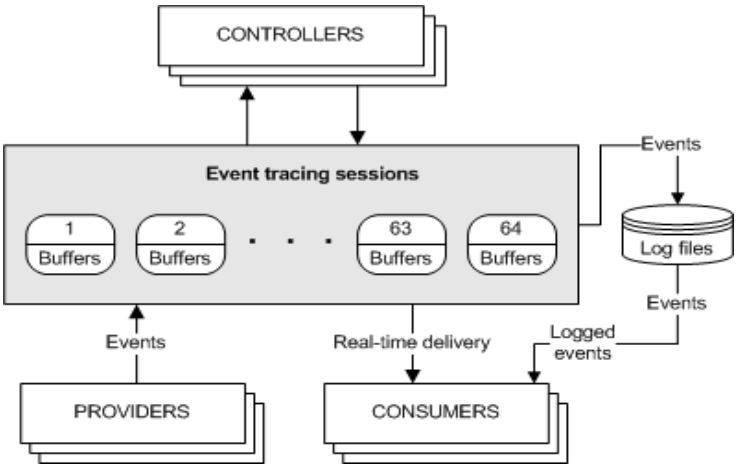

Abbildung 1: Komponenten von Event Tracing for Windows

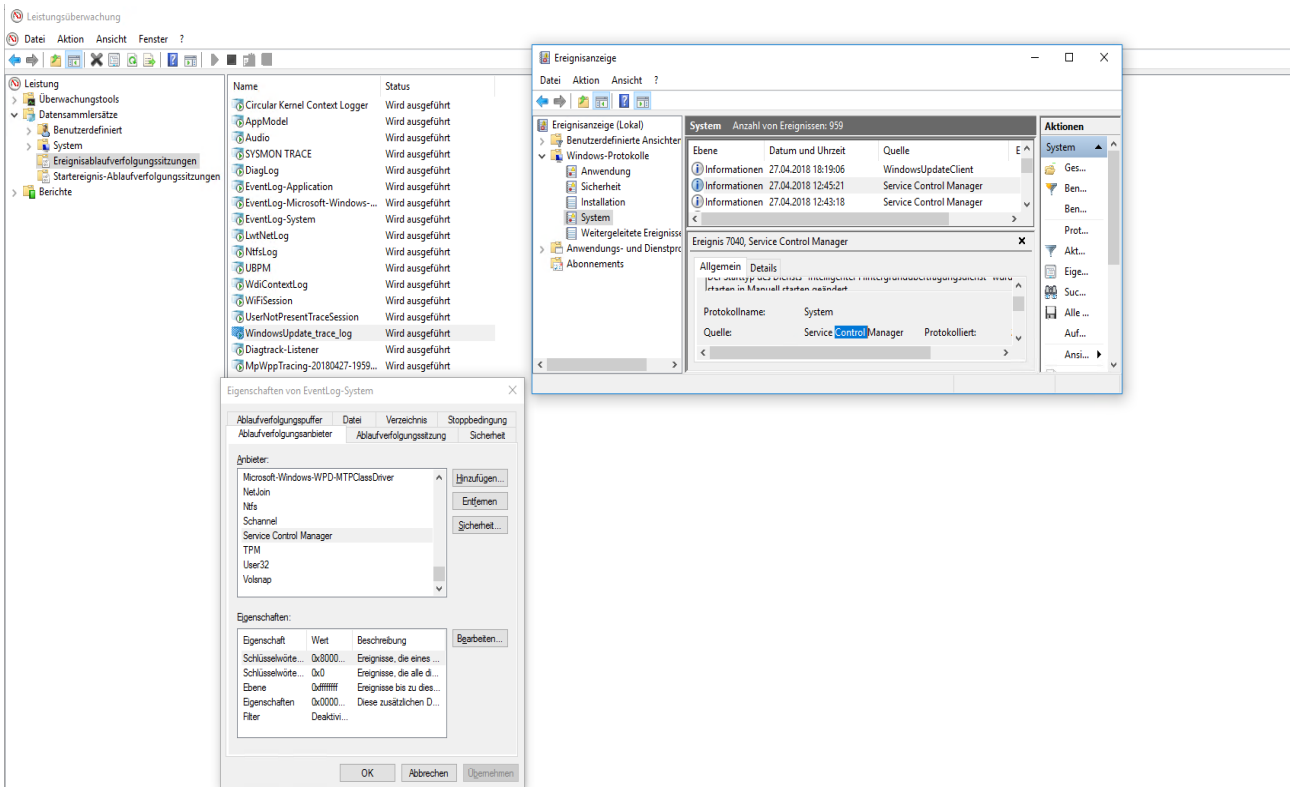

Abbildung 2: ETW Komponenten dargestellt im Performance Monitor

(ETW-) Controller (ETW-Kontrollinstanzen) sind Entitäten (z. B. `perfmon.exe` oder `logman.exe`), die Größe und Ort der Protokolldatei definieren, Provider aktivieren und eine Session initialisieren und beenden können.

(ETW-) Providers (ETW-Anbieter) sind Entitäten, die Events (Ereignisse) schreiben und diese über eine (ETW-) sogenannte Session einem (ETW-) Consumer zur weiteren Verarbeitung anbieten können, wenn sie von einem Controller vorher aktiviert wurden (so schreibt beispielsweise der ETW-Provider „TPM“ Status-Informationen des Trusted Platform Module (TPM) über die Session „Eventlog-System“ in das „System“-Protokoll).

(ETW-) Consumer (ETW-Konsumenten) sind Entitäten, die Events zur weiteren Verarbeitung über eine (ETW-) Session entweder in Echtzeit oder über Logdatei (z. B. Windows Ereignisprotokoll (eventvwr.exe)) erhalten können. Ein Consumer kann Events aus mehreren Sessions erhalten.

(ETW-) Sessions (ETW-Sitzungen) sind definiert durch Speicherbereiche (Kernel Buffer), in die aktivierte Provider ihre Events schreiben. (ETW-) Sessions zeichnen Ereignisse eines oder mehrere Provider auf, die vorher von dem Controller aktiviert wurden. ETW unterstützt in Windows 10 als Maximum die gleichzeitige Aufzeichnung von 64 Sessions.

Abbildung 2 bietet über das in Windows integrierte Tool `Leistungsüberwachung` (`perfmon.exe`) eine Sicht auf die ETW-Session „EventLog-System“, den hervorgehobenen Provider „TPM“ sowie ein Event (mit der ID 18), das in dieser Session geschrieben wurde und von dem (ETW-) Consumer „System“-Protokoll weiter verarbeitet (gespeichert) wird.

2 Architekturüberblick

Die Telemetrie in Windows 10 bedient sich der Funktionalität von ETW, um Diagnose-Daten (= Telemetrie-Daten) zu erheben (siehe Abschnitt 1.1).

Der Windows-Dienst „Benutzererfahrung und Telemetrie im verbundenen Modus“ (Connected User Experiences and Telemetry Service), auch DiagTrack genannt, ist der zentrale Baustein der Windows Telemetrie-Komponente. Er ist in der Datei `%SystemRoot%\System32\diagtrack.dll` implementiert und startet automatisch mit System-Privilegien im Lauf des Bootprozesses. Der DiagTrack-Dienst startet innerhalb seines eignen Services-Host-Container (`svchost.exe`). Abbildung 3 zeigt einen Ausschnitt aus dem `Process Explorer`, der diesen Sachverhalt darstellt:

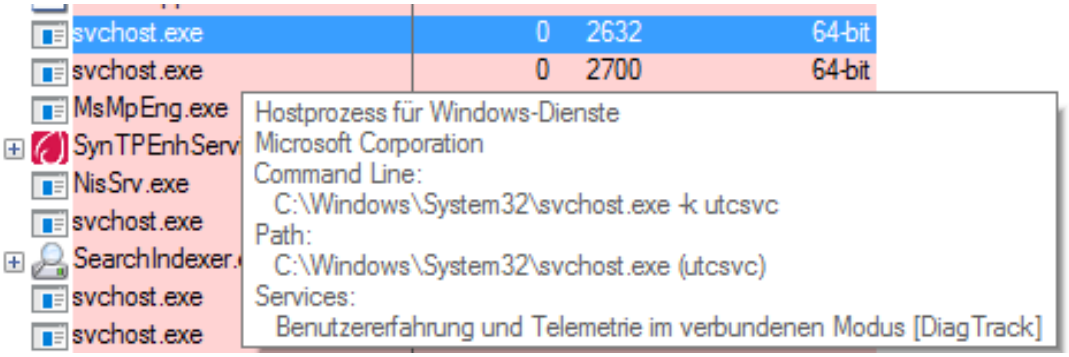

Abbildung 3: DiagTrack dargestellt im Process Explorer

Der DiagTrack-Dienst bezieht seine Daten aus zwei ETW-Sessions: eine mit dem Namen `Diagtrack-Listener` und eine mit dem Namen `Autologger-Diagtrack-Listener`. Von diesen beiden Sessions ist der Autologger-Diagtrack-Listener die zuerst gestartete Sitzung. Sie wird früh im Bootprozess initialisiert[1] und speichert die protokollierten Daten in den beiden Dateien `%ProgramData%\Microsoft\Diagnosis\ETLLogs\AutoLogger\AutoLogger-Diagtrack-Listener.etl` und `%ProgramData%\Microsoft\Diagnosis\ETLLogs\ShutdownLogger\AutoLogger-Diagtrack-Listener.etl`. Der Inhalt dieser Dateien ist in einem binären Format gespeichert und wird vom DiagTrack-Dienst nach dessen Start weiter verarbeitet. Die Session des `Autologger-Diagtrack-Listener` wird deaktiviert, sobald der DiagTrack-Dienst vom Betriebssystem in einer späteren Phase des Bootprozesses gestartet wird.[2] Die Session des Diagtrack-Listener wird beim Start des DiagTrack-Dienstes initialisiert. Der `Diagtrack-Listener` liefert die in seiner Sitzung aufgezeichnete Daten in Form eines Echtzeit-Protokoll-Feeds (*real time log feed*) an den DiagTrack-Dienst. Der DiagTrack-Dienst sendet die erhobenen Daten dann an verschiedene Hosts des Telemetrie-Backends von Microsoft. Die Übertragung der Daten vom DiagTrack-Dienst zum Telemetrie-Backend erfolgt stets TLS-verschlüsselt und ist vor Man-in-the-Middle-Angriffen durch Zertifikats-Pinning auf der Client-Seite geschützt.[3]

Abbildung 4 zeigt eine kompakte Übersicht über die Architektur der Telemetrie in Windows 10. Das als `Microsoft Data Management Service` bezeichnete Telemetrie-Backend enthält exemplarisch drei Hosts aus diesem Backend:

1 Die Initialisierung erfolgt im Zuge der sog. „Pre-Session Initialization“.
2 Dies geschieht kurz bevor der Anmeldebildschirm zur Verfügung steht.
3 Dabei vergleicht DiagTrack die von den Servern im Telemetrie-Backend bereitgestellten Zertifikate mit den (in `crypt32.dll`) fest einprogrammierten. Wenn sie nicht übereinstimmen, wird die Verbindung vom DiagTrack-Dienst beendet.

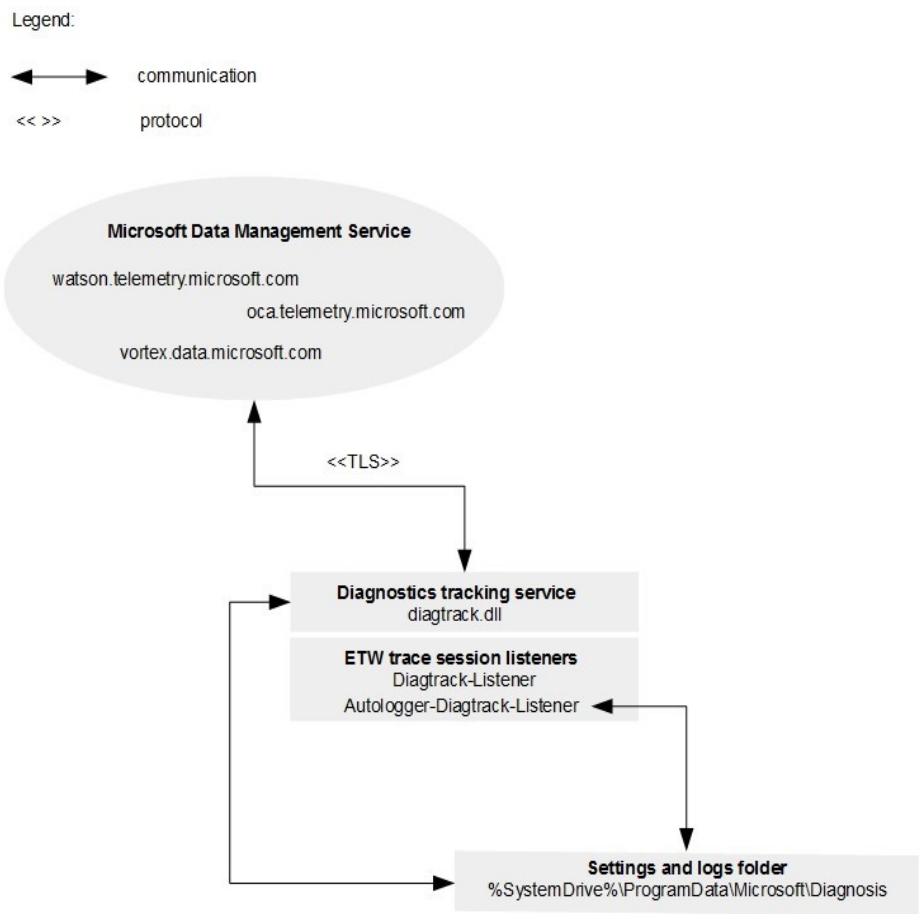

Abbildung 4: Telemetrie-Architektur

Abbildung 5 stellt einen Screenshot des mit dem `Microsoft Network Monitor` aufgenommenen Netzwerkverkehrs zwischen einem exemplarischen Windows 10-System mit der Internet Protocol(IP)-Adresse 172.18.1.184 und typischen Hosts des Telemetrie-Backends dar. Der Umfang der über die Telemetrie erhobenen Daten hängt maßgeblich von dem Wert für die Konfiguration des Telemetrie-Levels ab. Nur in Windows 10 Enterprise und einigen speziellen anderen Versionen steht der niedrigste Telemetrie-Level `Security` [4] zur Verfügung. Damit stehen diesen Versionen insgesamt die folgenden vier Telemetrie-Level zur Verfügung: `Security`, `Basic`, `Enhanced` und `Full`. In allen anderen (d. h. den Consumer-) Windows-Versionen ist der niedrigst mögliche Level `Basic`.

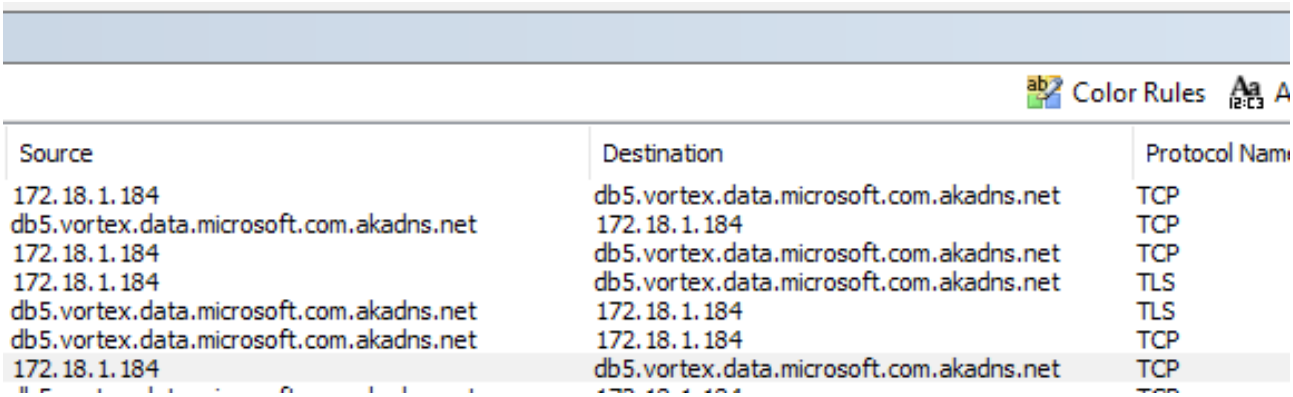

Abbildung 5: Aufzeichnung eines Netzwerkverkehrs zum Telemetrie-Backend

4 Der Telemetrie-Level „`Security`“ steht in den folgenden Windows Versionen zur Verfügung: Windows 10 Enterprise, Education, Mobile Enterprise und IoT Core

Der Umfang der Datenerhebung nimmt aufsteigend von `Security` bis hin zu `Full` zu und hängt technisch von der Anzahl der ETW-Provider ab, die für jeden Level in den beiden ETW-Sessions (`Autologger-Diagtrack-Listener` und `Diagtrack-Listener`) aktiviert sind. Eine generische Übersicht, welche Arten von Daten auf welchem Telemetrie-Level erhoben werden, findet sich bei Microsoft. [ms_configdiag]

Für die Autologger-Diagtrack-Listener-Session stellt sich die Anzahl der ETW-Provider pro Telemetrie-Level wie folgt dar:

Telemetrie-Level	ETW-Provider-Anzahl
`Security`	9
`Basic`	93
`Enhanced`	105
`Full`	112

Tabelle 1: Autologger-Diagtrack-Listener-Session

Für die Diagtrack-Listener-Session stellt sich die Anzahl der ETW-Provider pro Telemetrie-Level wie folgt dar:

Telemetrie-Level	ETW-Provider-Anzahl
`Security`	4
`Basic`	410
`Enhanced`	418
`Full`	422

Tabelle 2: Diagtrack-Listener-Session

Grundsätzlich lässt sich die Anzahl der Provider über den `Performance Monitor` einsehen. Die folgende Abbildung zeigt einen Ausschnitt der für den Diagtrack-Listener konfigurierten Provider:

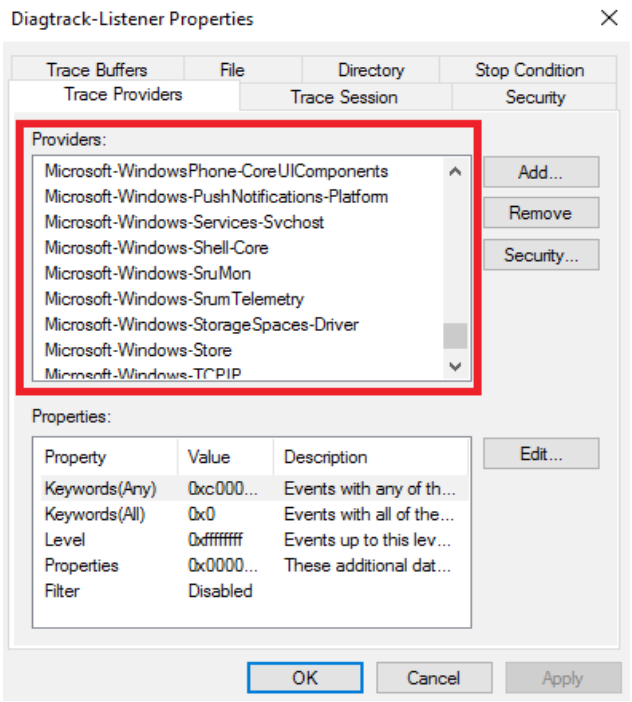

Abbildung 6: ETW-Provider Diagtrack Listener

Ein wichtiger Aspekt hinsichtlich der Bewertung der Quantität der erhobenen Daten sowie des grundsätzlichen Einflusses den Microsoft auf die Erhebung (und damit auch auf das System) hat und damit ein wichtiger Aspekt der Gesamtarchitektur ist: Die auf einem Windows System tatsächlich konfigurierte Anzahl von ETW-Providern eines bestimmten Telemetrie-Levels ist dynamisch und hängt von dem Inhalt der folgenden Datei ab: `%ProgramData%\Microsoft\Diagnosis\DownloadedSettings\utc.app.json`. Diese Datei definiert den Inhalt der pro Telemetrie-Level aktivierten ETW-Provider, sie wird von Microsoft kontrolliert, in Abständen über automatische Downloads erneuert und kann von Betriebssystemversion zu Betriebssystemversion sowie von Installation zu Installation (selbst bei gleicher Betriebssystemversion) variieren. Telemetrie bietet Microsoft darüber hinaus die Funktionalität, zusätzliche Programme oder auch einzelne Funktionen in Bibliotheken aufzurufen, um weitergehende Informationen wie bspw. Speicher-Dumps zur Ergänzung einer Fehlermeldung zu erheben (was der Telemetrie-Dienst kann, da er mit System-Privilegien läuft). Darüber hinaus ist es technisch möglich, dass beliebige Prozesse (die über administrative Rechte oder höher verfügen) ETW-Provider registrieren und mit den DiagTrack ETW-Sessions assoziieren können.

3 Deaktivierung und Reduktion

Diese Kapitel beschreibt die unterschiedlichen Varianten zur Deaktivierung und/oder Reduktion der Erhebung von Telemetrie-Daten. Die Maßnahmen dazu sind in System-basierte sowie Netzwerk-basierte Maßnahmen unterteilt. System-basierte Maßnahmen nutzen Möglichkeiten eines Windows 10 Systems, Netzwerk-basierte Maßnahmen bedienen sich Funktionalität typischer zentraler Netzwerkkomponenten wie beispielsweise Proxy- und Domain Name System(DNS)-Servern.

3.1 System-basierte Maßnahmen

Die in diesem Abschnitt beschriebenen Konfigurationsmöglichkeiten werden auf dem System selbst vorgenommen. Sie dienen dazu, die Kommunikation der DiagTrack-Komponente zu Microsofts Backend zu unterbinden oder – falls dies nicht möglich ist - bestmöglich einzuschränken. Alle Änderungen (der Weg über die Windows 10 `Einstellungen`) müssen mit administrativen Rechten vorgenommen werden. Die Konfiguration der Einstellungen wird (falls jeweils verfügbar) über die Windows 10 `Einstellungen`, die lokale Gruppenrichtlinie (`gpedit.msc`), die Registry (`regedit.exe`) oder alternative Möglichkeiten beschrieben. Sofern nicht anders angemerkt haben die unterschiedlichen Konfigurationsmöglichkeiten eine identische Auswirkung.

3.1.1 Konfiguration des niedrigst möglichen Telemetrie-Levels

Um die Anzahl der ETW-Provider, die in die `DiagTrack-Listener` Daten schreiben, zu reduzieren, kann das Telemetrie-Level konfiguriert werden. Unter Windows 10 Enterprise und Long Term Servicing Branch (LTSB) gibt es die Möglichkeit das Level auf "`0 – Security`" zu setzen. Bei Non-Enterprise Windows 10 Versionen kann der Wert auch gesetzt werden, entspricht dann aber mindestens dem Level "`1 – Basic`". Die folgende Tabelle beschreibt die unterschiedlichen äquivalenten Konfigurationsmöglichkeiten:

3.1.2 Deaktivierung von Telemetrie-Dienst und ETW-Sessions

Wie in Kapitel 2 beschrieben schreiben die ETW-Provider ihre Daten in die ETW-Sessions `Autologger-DiagTrack-Listener` und `DiagTrack-Listener`. Diese Sessions sind die Quelle der Telemetrie-Daten. Durch die Deaktivierung der Sessions wird die Telemetrie-Datensammlung unterbunden. Um die beiden Sessions sowie die Übertragung von Telemetrie-Daten zu deaktivieren, muss zuerst der Dienst `Benutzererfahrung und Telemetrie im verbundenen Modus` (`Connected User Experience and Telemetry`) deaktiviert werden. Dadurch wird die Initiierung der `DiagTrack-Listener Session` verhindert. Zusätzlich muss die `Autologger-DiagTrack-Listener Session` deaktiviert werden. Die Deaktivierung kann in der Registry vorgenommen werden; dazu muss der Wert des entsprechenden Registrierungsschlüssels auf `0` gesetzt werden.

Schnittstelle	Pfad/Befehl
services.msc	`Benutzererfahrung und Telemetrie im verbundenen Modus → Eigenschaften → Starttyp → Deaktiviert`
Registry	`HKLM\SYSTEM\CurrentControlSet\Services\DiagTrack\Start = 4`
Powershell	`Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\ Services\DiagTrack\ -name Start -Value 4`

Tabelle 3: Schritt 1: Deaktivierung Benutzererfahrung und Telemetrie im verbundenen Modus

Schnittstelle	Pfad/Befehl

Registry	`HKLM\SYSTEM\CurentControlSet\Control\WMI\Autologger\AutoLogger-DiagTrack-Listener\Start = 0`
	Alternativer Powershell Befehl:
	`Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\WMI\Autologger\AutoLogger-Diagtrack-Listener\ -name Start -Value 0`

Perfmon.exe	`Datensammlersätze` → `Startereignis-Ablaufverfolgunssitzungen` → `AutoLogger-Diagtrack-Listener` → `Eigenschaften` → `Ablaufverfolgungssitzung` → Haken bei `Aktiviert` entfernen

Tabelle 4: Schritt 2: Deaktivierung Autologger-Diagtrack-Listener

3.1.3 Deaktivierung Telemetrie nach Microsoft Empfehlung

Microsoft stellt eigene Empfehlungen bereit um die Erhebung von Telemetrie-Daten zu reduzieren[ms_configdiag]. Neben der Konfiguration des Telemetrie-Levels auf `0-Security` (Siehe Kapitel 3.1.1), sollen folgende Komponenten deaktiviert werden, die auch auf Telemetrie-Level 0 Daten senden:

- `Windows Update`
- `Cloud-Based-Protection (Windows Defender)`

Durch die Nutzung von einem `Windows Server Update Service` oder einem `System Center Configuration Manager` können die Updates von einer lokalen Instanz bezogen werden. Die Komponente `Cloud-Based Protection` des `Windows Defender` dient dazu Informationen über verdächtige Software an Microsoft zu übermitteln.

Schnittstelle	Pfad/Befehl
Services.msc	Windows Update → Eigenschaften → Starttyp → Deaktiviert
Registry	`HKLM\SYSTEM\CurrentControlSet\Services\wuauserv\Start = 4`
Powershell	`Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\wuauserv\ -name Start -Value 4`
GPO	Via GPO kann lediglich das automatisierte Update konfiguriert, allerdings nicht der Update-Dienst deaktiviert werden.

Tabelle 5: Deaktivierung Windows Update

Schnittstelle	Pfad/Befehl
Einstellungen	Update und Sicherheit → Windows Defender → Cloudbasierter Schutz = Aus
GPO	Via GPO kann die Kommunikation mit Cloud-basierten Backend eingeschränkt, allerdings nicht komplett unterbunden werden.

Tabelle 6: Deaktivierung cloudbasierter Schutz (Windows Defender)

3.1.4 Lokale Firewall-Regeln

Mit der in das Betriebssystem integrierten `Windows Defender Firewall mit erweiterter Sicherheit` lassen sich unter anderem Netzwerkverbindungen von ausführbaren Dateien blockieren. Da für die Übermittlung der DiagTrack-Dienst verantwortlich ist, soll dieser an der Ausführung gehindert werden. Viele Windows-Dienste, darunter auch der DiagTrack-Dienst, werden von dem Dienst-Host-Prozess `svchost.exe` gestartet. Ein Blockieren von Netzwerkverbindungen dieser ausführbaren Datei würde also nicht nur den DiagTrack-Dienst blockieren, sondern auch alle anderen Dienste die `svchost.exe` nutzen. In den folgenden Schritten ist beschrieben wie sich der DiagTrack-Dienst von den anderen Diensten isolieren lässt, um nur diesen Dienst an der Initiierung von Netzwerkverbindungen zu hindern.

1. Erstellung eines Hardlinks auf `svchost.exe` mit anderem Namen (in `%SystemRoot%\System32\`). Hierfür ist eine Anpassung der Berechtigungen notwendig. Alle entsprechenden Befehle (inklusive der nächsten Schritte) sind im Anhang unter Erstellung lokale Firewall-Regel für Telemetrie-Dienst beschrieben.
2. Änderung des Pfads der Ausführung in der Registrierungsdatenbank . Hierzu zum Pfad `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DiagTrack` navigieren und den Wert des Schlüssels `ImagePath` in `%SystemRoot%\System32\utc_myhost.exe -k utcsvc -p` ändern. Damit wird der Dienst mit der kopierten Datei gestartet.
3. Anlegen einer neuen ausgehenden Regel. Die Regel so definieren dass das Programm `%SystemRoot%\System32\utc_myhost.exe` blockiert wird Netzwerkverbindungen aufzubauen.
4. Das System neustarten.

Anlegen ausgehenden Regel in der Windows Defender Firewall mit erweiterter Sicherheit

Schnittstelle	Pfad/Befehl
wf.msc	Ausgehende Regel → Neue Regel → Programm → `%SystemRoot%\System32\utc_myhost.exe -k utcsvc -p` → Verbindung blockieren
Powershell	`New-NetFirewallRule -DisplayName "BlockDiagTrack" -Name "BlockDiagTrack" -Direction Outbound -Program "%SystemRoot%\System32\utc_myhost.exe" -Action block`

Tabelle 7: Windows Defender Firewall Regel

3.1.5 Lokale DNS Einträge

Vor jedem Verbindungsaufbau zum Telemetrie-Backend von Microsoft erfolgt zunächst die Auflösung des DNS-Namens des Telemetrie-Backend-Hosts in dessen zugehörige IP-Adresse. Windows Betriebssysteme versuchen DNS-Namen über Einträge in der lokalen Hosts-Datei aufzulösen, bevor sie einen DNS-Server abfragen. Diesen Umstand kann man sich zunutze machen, um allen bekannten Hostnamen von Telemetrie-Endpunkten von Microsoft die IP-Adresse 0.0.0.0 zuzuordnen. Da die IP-Adresse 0.0.0.0 nicht routing-fähig ist, zeigen die dieser Adresse zugeordneten DNS-Namen „ins Leere“, und können damit nicht erreicht werden. Dementsprechend können die dort aufgelisteten Telemetrie-Backend-Hosts nicht erreicht werden. Die Hosts-Datei befindet sich unter: `%windir%\system32\drivers\etc\hosts[ms_dnsres]` und muss mit Admin-Rechten editiert werden. Die Syntax der Datei ist:

```
<IP> <HOSTNAME>
```

Die Liste der bekannten Hostnamen ist:

Hostname	Location
geo.settings-win.data.microsoft.com.akadns.net, db5-eap.settings-win.data.microsoft.com.akadns.net, settings-win.data.microsoft.com, db5.settings-win.data.microsoft.com.akadns.net, asimov-win.settings.data.microsoft.com.akadns.net	Ireland, Dublin
db5.vortex.data.microsoft.com.akadns.net, v10-win.vortex.data.microsft.com.akadns.net, geo.vortex.data.microsoft.com.akadns.net, v10.vortex-win.data.microsft.com	Ireland, Dublin
us.vortex-win.data.microsoft.com	Virginia (US), Boyton
eu.vortex-win.data.microsoft.com	Netherlands, Amsterdam
vortex-win-sandbox.data.microsoft.com	California (US), Los Angeles
alpha.telemetry.microsoft.com	California (US), Los Angeles
oca.telemetry.microsoft.com	Wyoming (US), Cheyenne

Tabelle 8: Bekannte Hostnamen

Eine entsprechende `hosts` Datei würde dementsprechend folgendermaßen aussehen:

```
<bestehender Inhalt der Datei>
0.0.0.0 geo.settings-win.data.microsoft.com.akadns.net
0.0.0.0 db5-eap.settings-win.data.microsoft.com.akadns.net
0.0.0.0 settings-win.data.microsoft.com
0.0.0.0 db5.settings-win.data.microsoft.com.akadns.net
0.0.0.0 asimov-win.settings.data.microsoft.com.akadns.net
0.0.0.0 db5.vortex.data.microsoft.com.akadns.net
0.0.0.0 v10-win.vortex.data.microsft.com.akadns.net
0.0.0.0 geo.vortex.data.microsoft.com.akadns.net
0.0.0.0 v10.vortex-win.data.microsft.com
0.0.0.0 us.vortex-win.data.microsft.com
0.0.0.0 eu.vortex-win.data.microsft.com
0.0.0.0 vortex-win-sandbox.data.microsoft.com
0.0.0.0 alpha.telemetry.microsft.com
0.0.0.0 oca.telemetry.microsft.com
```

3.2 Netzwerk-basierte Maßnahmen

In diesem Abschnitt werden Konfigurationsmöglichkeiten für zentrale Netzwerkdienste beschrieben, welche die Kommunikations der DiagTrack-Komponente Netzwerk-basiert für alle angeschlossenen Client-Systeme unterbinden oder einschränken können.

3.2.1 HTTP-Proxy

Die Bereitstellung eines Hypertext Transfer Protocol(HTTP)-Proxy Servers, der verpflichtend von allen Client Systemen genutzt werden muss, ermöglicht die Filterung des Netzwerkverkehrs und damit die Unterbindung von unerwünschter Kommunikation. Zur zentralen Blockierung der Netzwerkverbindungen der Telemetrie Dienste muss der eingesetzte Proxy so konfiguriert werden, dass alle HTTP Anfragen für die mit dem Telemetrie Dienst verbundenen DNS Namen zurückgewiesen werden. Gleichzeitig muss der DNS Server bei allen Windows 10 Clients als systemweiter Proxy Server eingetragen werden.

Exemplarisch wird hier der offene Proxy-Server `Squid` mit der folgenden Konfiguration genutzt. Zuerst werden die zu blockierenden DNS Namen in einer Datei eingetragen, die in einem beliebigen Verzeichnis (beispielsweise als `/etc/squid/telemetry-domains.squid`) gespeichert wird. Die Datei muss alle zu blockierenden DNS Namen enthalten, wobei jeder DNS Name in einer neuen Zeile eingetragen wird, beispielsweise:

```
domain.de  # Blockiert nur Zugriffe auf domain.de
.domain.de # Blockiert Zugriffe auf domain.de und alle Subdomains
```

Eine Datei `/etc/squid/telemetry-domains.squid` , die jede mit dem Telemetrie Dienst verbundene Kommunikation über den Proxy blockiert, muss daher folgende Einträge enthalten:

```
geo.settings-win.data.microsoft.com.akadns.net
db5-eap.settings-win.data.microsoft.com.akadns.net
settings-win.data.microsoft.com
db5.settings-win.data.microsoft.com.akadns.net
asimov-win.settings.data.microsoft.com.akadns.net
db5.vortex.data.microsoft.com.akadns.net
v10-win.vortex.data.microsoft.com.akadns.net
geo.vortex.data.microsoft.com.akadns.net
v10.vortex-win.data.microsoft.com
us.vortex-win.data.microsoft.com
eu.vortex-win.data.microsoft.com
vortex-win-sandbox.data.microsoft.com
alpha.telemetry.microsoft.com
oca.telemetry.microsoft.com
```

In der Konfigurationsdatei `/etc/squid/squid.conf` kann nun die erstellte Datei referenziert werden:

```
acl telemetry dstdomain “/etc/squid/telemetry-domains.squid”
```

Die in der Datei enthaltenen Domain Namen können dann durch eine Deny-Regel in der Konfigurationsdatei `/etc/squid/squid.conf` explizit blockiert werden:

```
http_access deny telemetry
```

Zu beachten ist, dass diese Regel **vor** allen anderen `http_access allow` Regeln, die bestimmten Netzwerkverkehr erlauben, stehen muss. Nur dann ist gewährleistet, dass die Blockierung der DNS Namen Vorrang gegenüber anderen Regeln erhält.

Die folgende Grafik zeigt das Ergebnis eines Verbindungsversuchs des Telemetrie Dienstes, wenn ein entsprechend konfigurierter Proxy in den Windows-Einstellungen eingetragen wurde.

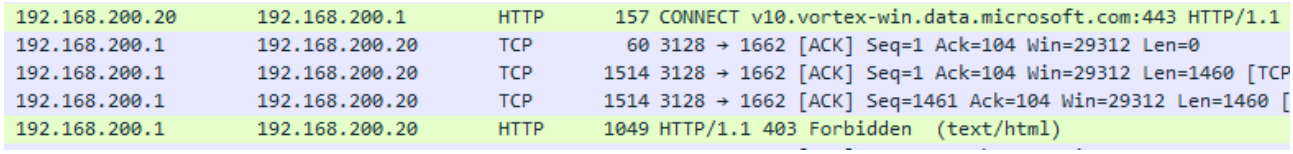

Abbildung 7: Durch Proxy blockierte Telemetrie-Kommunikation

Die entsprechenden Anfragen werden daraufhin bereits vom Proxy mit dem HTTP Statu Code 403 beantwortet und eine Verbindung kommt nicht zustande.

3.2.2 Firewall

Die Blockierung der Kommunikationsverbindungen der Telemetrie-Dienste kann auch auf einer zentralen Firewall erfolgen. Die Problematik hierbei ist, dass die meisten kommerziellen und frei verfügbaren Firewalls Netzwerkverbindungen nicht basierend auf DNS Namen filtern können, sondern die Filterung technisch anhand der IP-Netzwerkadressen vornehmen. Da die für den Telemetrie Dienst verwendeten DNS Namen allerdings auf Content Delivery Networks (CDNs) verweisen können, deren IP Adressen regelmäßig wechseln können, müssen auch die tatsächlichen Filter-Regeln regelmäßig angepasst werden. Anderenfalls kann es passieren, dass die Kommunikation des Telemetrie Dienstes über neu zugewiesene IP Adressen nicht mehr blockiert wird oder mittlerweile anderweitig verwendete IP Adressen trotzdem weiterhin blockiert werden. Die verwendete Firewall muss daher entweder eine Filterung anhand DNS Namen erlauben (und die regelmäßigen Anpassungen selbst im Hintergrund vornehmen) oder von einer eigenen Komponente wie einem Skript in regelmäßigen Abständen angepasst werden.

Die offene `iptables` Firewall erlaubt beispielsweise die Konfiguration von Firewall Regeln anhand von DNS Namen, allerdings werden diese nur bei der Erstellung der Regel zu den dazugehörigen IP Adressen aufgelöst und nicht regelmäßig aktualisiert. Der Befehl

```
iptables -A OUTPUT -s oca.telemetry.microsoft.com -j DROP
```

erzeugt beispielsweise den folgenden Eintrag in der Firewall:

```
-A OUTPUT -s 51.143.10.67/32 -j DROP
```

Ein Skript, das durch DNS Anfragen prüft ob sich die IP Adressen der zu blockierenden DNS Namen geändert haben und ggf. die `iptables` Regeln anpasst, kann dann in regelmäßigen Abständen ausgeführt werden, beispielsweise durch einen entsprechenden Eintrag in `/etc/crontab`.

Zusätzlich können auf der Firewall auch die entsprechenden DNS Anfragen selbst blockiert werden, sodass Verbindungsversuche von Clients bereits an der Auflösung der DNS Namen zu IP Adressen scheitern. Hierbei ist zu beachten, dass dies nur die DNS Anfragen, nicht aber die tatsächlichen Datenverbindungen des Telemetrie Dienstes unterbinden kann, beispielsweise wenn die zu den DNS Namen gehörigen IP Addressen bereits im Cache des Clients vorhanden sind oder dieser sie auf einem anderen Kommunikationskanal erhält.

Mit `iptables` kann eine solche Blockierung von bestimmten DNS Anfragen umgesetzt werden, indem der Netzwerkverkehr zu dem Port 53, der für DNS verwendet wird, auf die Zeichenketten der zu blockierenden DNS Namen gefiltert wird. Wie im folgenden Beispiel gezeigt, müssen dabei die Domain Namen in einem bestimmten Format angegeben werden, das in den DNS Anfragen verwendet wird. Die Punkte im DNS Namen werden weggelassen und dafür muss vor jedem Teil des DNS Namens die Länge der Zeichenkette in hexadezimaler Notation angegegeben werden:

```
iptables -A OUTPUT -p udp --dport 53 -m string --hex-string “|03|oca|09|
telemetry|09|microsoft|03|com” -algo bm -j DROP
```

3.2.3 DNS-Einträge auf zentralem Resolver

Wird ein zentraler DNS Server eingesetzt kann dieser konfiguriert werden die eingehenden Anfragen für zu blockierende DNS Namen mit nicht-routing-fähigen Einträgen zu beantworten, wie beispielsweise `0.0.0.0`.

Sofern anfragende Client Systeme und Anwendungen die entsprechenden IP Adressen nicht in ihrem Cache haben und auch keine andere Möglichkeit besitzen DNS Namen zu IP-Adressen aufzulösen führt dies dazu, dass sie keine Verbindung zum eigentlichen Ziel-System aufbauen können. Die Effektivität der Maßnahme

zum Blockieren der Kommunikation der Telemetrie-Dienste hängt dabei maßgeblich von der konkreten Implementierung des Dienstes ab. Akzeptiert der Dienst die “falschen” Antworten, so wird er die gewünschten Verbindungen nicht aufbauen können. Dieses Verhalten kann allerdings jederzeit durch eine entsprechende Anpassung der Software geändert werden. Beispielsweise könnte der Dienst eine selbstständige Auflösung von DNS Namen über einen separaten und möglicherweise verschlüsselten Kommunikationskanal vornehmen und anschließend trotzdem Verbindungen zu den korrekten IP Adressen aufbauen, was nur durch andere Maßnahmen verhindert werden kann.

Exemplarisch wird diese Maßnahme mit dem offenen DNS-Server `bind` getestet. Hierzu werden die zu blockierenden DNS Namen in der Konfigurationsdatei `/etc/named.conf` als eigene Zonen eingetragen, die auf eine Zonen-Datei mit den nicht-routing-fähigen IP Adressen verweist. Die Zonen-Einträge in der Konfigurationsdatei zum Blockieren des Telemetrie Dienstes lauten dabei wie folgt:

```
zone “geo.settings-win.data.microsoft.com.akadns.net” { type master; file “dummy-zone”; };
zone “db5-eap.settings-win.data.microsoft.com.akadns.net” { type master; file “dummy-zone”; };
zone “settings-win.data.microsoft.com” { type master; file “dummy-zone”; };
zone “db5.settings-win.data.microsoft.com.akadns.net” { type master; file “dummy-zone”; };
zone “asimov-win.settings.data.microsoft.com.akadns.net” { type master; file “dummy-zone”; };
zone “db5.vortex.data.microsoft.com.akadns.net” { type master; file “dummy-zone”; };
zone “v10-win.vortex.data.microsoft.com.akadns.net” { type master; file “dummy-zone”; };
zone “geo.vortex.data.microsoft.com.akadns.net” { type master; file “dummy-zone”; };
zone “v10.vortex-win.data.microsoft.com” { type master; file “dummy-zone”; };
zone “us.vortex-win.data.microsoft.com” { type master; file “dummy-zone”; };
zone “eu.vortex-win.data.microsoft.com” { type master; file “dummy-zone”; };
zone “vortex-win-sandbox.data.microsoft.com” { type master; file “dummy-zone”; };
zone “alpha.telemetry.microsoft.com” { type master; file “dummy-zone”; };
zone “oca.telemetry.microsoft.com” { type master; file “dummy-zone”; };
```

In dem Verzeichnis von `bind`, das in der Konfigurationsdatei als `directory` hinterlegt ist (standardmäßig ist dies `/var/named`) muss dann die entsprechende Zonen Datei `dummy-zone` angelegt werden. Damit Anfragen für die in der Konfigurationsdatei definierten DNS Namen und alle ihre Subdomains mit der nicht routingfähigen IP Adresse 0.0.0.0 beantwortet werden und Clients diese Information für 24 Stunden speichern, könnte die Zonen-Datei die folgenden Einträge enthalten:

```
@     IN    SOA   dnsserver.example.com. hostmaster.example.com.
                  (201703030086400 300 604800 3600)
@     IN    NS    dnsserver.example.com
@     IN    A     0.0.0.0
*     IN    A     0.0.0.0
```

Die Zeichenkette `dnsserver.example.com` muss dabei an den DNS Namen des verwendeten DNS Servers angepasst werden. Zudem muss `hostmaster.example.com` durch die Email-Addresse des verantwortlichen Administrators ersetzt werden, wobei das @ Zeichen durch den ersten Punkt der Zeichenkette ohne davorstehenden Backslash („\“) beschrieben wird. Die vorletzte Zeile sorgt dafür, dass die in der Konfigurationsdatei `/etc/named.conf` angegebenen DNS Namen mit `0.0.0.0` beantwortet werden, wobei die letzte Zeile dies auch für alle Subdomains des DNS Namens sicherstellt.

4 Wirksamkeit und Empfehlung

Diese Kapitel beschreibt die Wirksamkeit der einzelnen in Kapitel 3 beschriebenen Maßnahmen zur Reduktion und Deaktivierung von Telemetrie-Aktivität. Die Wirksamkeit wurde dabei über einen Zeitraum von mindestens 48 Stunden pro Maßnahme verifiziert. Die genaue Vorgehensweise sowie detaillierte Ergebnisse sind im Anhang unter Verifikation beschrieben.

Allgemein kann die Telemetrie-Auswertung Bestandteil bestimmter Support-Modelle mit Microsoft sein, so dass evaluiert werden muss, ob der IT-Support nach Deaktivierung des Telemetrie-Dienstes noch gewährleistet werden kann. Weiterhin wird durch die Deaktivierung ein direkter Kanal zu Microsoft für die Übermittlung von Nutzer-Feedback entfernt, so dass beispielsweise Informationen über fehlerhafte Systemkomponenten weniger zeitnah behoben werden können.
Die folgende Tabelle beschreibt die Wirksamkeit der einzelnen Maßnahmen und listet Aspekte auf, die für den Betrieb (sowohl eines Endnutzer- als auch Unternehmens-System) relevant sind. Es gibt verschiedene Maßnahmen die Telemetrie-Aktivität wirksam unterbinden, die Auswahl der geeignetsten Maßnahme hängt von der betrieblichen Umgebung des Zielsystems ab.

Konfigurationsvariante	Wirksamkeit	Operationelle Aspekte
System-basierte Maßnahmen		
Konfiguration des niedrigst möglichen Telemetrie-Levels	Reduziert Telemetrie-Aktivität	Diese Option hat keine bekannten betrieblichen Auswirkungen, da die Konfiguration explizit von Microsoft unterstützt wird und per Gruppenrichtlinie verteilt werden kann.
Deaktivierung von Telemetrie-Dienst und ETW-Sessions	Vollständig	Die Option stellt eine Konfiguration mit Bordmitteln dar, die auch automatisiert verteilt werden kann. Die Deaktivierung hat keine bekannten Auswirkungen auf die Windows Systemstabilität.
Deaktivierung Telemetrie nach Microsoft Empfehlung	Vollständig	Die Option wird explizit von Microsoft unterstützt und kann per Gruppenrichtlinie verteilt werden. Diese Variante kann allerdings nur empfohlen werden, wenn Updates trotz Deaktivierung des Windows Update Dienstes zeitnah ausgerollt werden können.
Lokale Firewall-Regeln	Vollständig	Diese Option erfordert die Modifikation von Zugriffsrechten einer relevanten Systemdatei, die Ausführung des DiagTrack-Dienstes in einem umbenannten `svchost` Prozess und die Erstellung einer angepassten Firewall-Regel. Diese Modifikationen können Seiteneffekte zukünftiger Systemaktualisierungen nach sich ziehen, die bisher nicht bekannt sind.
Lokale DNS Einträge	Vollständig	Diese Maßnahme erfordert die kontinuierliche Pflege der DNS-Namen in der `hosts` Datei, da sich die Kommunikationspunkte über Updates hinweg verändern können. Weiterhin existiert bisher keine vollständige Liste aller Endpunkte, die kontinuierlich öffentlich gepflegt wird.
Netzwerk-basierte Maßnahmen		
HTTP-Proxy und DNS-	Vollständig	Diese Maßnahme erfordert die kontinuierliche Pflege der

Konfigurationsvariante	Wirksamkeit	Operationelle Aspekte
Einträge auf zentralem Resolver		zu blockierenden DNS-Namen, da sich die Kommunikationspunkte über Updates hinweg verändern können. Weiterhin existiert bisher keine vollständige Liste aller Endpunkte, die kontinuierlich öffentlich gepflegt wird.
Firewall	Vollständig	Diese Maßnahme erfordert die kontinuierliche Pflege der zu blockierenden DNS-Namen, da sich die Kommunikationspunkte über Updates hinweg verändern können. Weiterhin existiert bisher keine vollständige Liste aller Endpunkte, die kontinuierlich öffentlich gepflegt wird. Darüber hinaus arbeiten zentrale Firewalls typischerweise auf IP-Adressen und lösen DNS-Namen nur beim initialen Anlegen auf. Damit wird nicht sichergestellt, dass alle IP-Adressen blockiert werden, die sich beispielsweise in einem CDN hinter einem DNS-Namen verbergen können.

Tabelle 9: Wirksamkeit der Empfehlungen

Die Tabelle zeigt Vor- und Nachteile der unterschiedlichen Konfigurationsmöglichkeiten auf. Sowohl für Unternehmensumgebungen als auch Endnutzer bietet unter Betrachtung typischer Betriebsanforderungen die Deaktivierung von Telemetrie-Dienst und ETW-Sessions das beste Verhältnis aus wirksamer Telemetrie-Unterbindung und operationellen Auswirkungen. Als zusätzliche Maßnahme auf Netzwerk-Ebene bietet sich die Konfiguration eines DNS-Resolvers an, da diese in den meisten Unternehmensumgebungen und auch in Umgebungen versierter Endnutzer vorhanden sind und die benötigten Filterungsanforderungen effektiv und mit geringem Aufwand erfüllen.

Anhang

Verifikation

Um die Effektivität und eventuelle Auswirkungen der unterschiedlichen Konfigurationen aus der Konfigurationsempfehlung zur Telemetrie auf den Test-Systemen zu überwachen, wurde ein Monitoring-Framework implementiert, welches relevante Information sammelt, speichert und entsprechend darstellt. In diesem Kapitel wird dieses Framework sowie die Beobachtungen der unterschiedlichen Konfigurationen beschrieben.

Motivation

Windows-Systeme unterscheiden sich je nach Version in ihren Konfigurationsmöglichkeiten. Um ein sich ständig änderndes Betriebssystem zu überwachen wird ein entsprechendes Monitoring implementiert. Dieses soll die Basis zur Überwachung des Windows-Betriebssystem und der einzelnen Komponenten bilden. Die zentralisierte Speicherung ermöglicht dabei eine Korrelation von Daten aus verschiedenen Quellen.

Gliederung

Die Inhalte dieses Kapitels werden wie folgt zusammengefasst:

- Verschiedene Komponenten bilden zusammen das *Monitoring-Framework*. In diesem Abschnitt werden die Funktion und Relevanz der einzelnen Komponenten beschrieben und dargestellt.
- Die Testumgebung besteht aus dem Framework, dem untersuchenden Windows-System und der Anbindung an das Internet. Dieser Abschnitt beschreibt den Aufbau der Umgebung und die notwendige Konfiguration des Windows-System.
- Der letzte Abschnitt beschreibt die Beobachtungen basierend auf den unterschiedlichen Konfigurationsmöglichkeiten.

Monitoring-Framework

Das Framework basiert auf dem `Elastic-Stack [elk_stack]` und dient der Überwachung von System- und Netzwerkaktivitäten von Windows-Systemen sowie der nachfolgenden Analyse. Das Framework bezieht Daten aus drei verschiedenen Quellen:

- Systemdefinierte Log-Kanäle
- Benutzerdefinierte `ETW-Session`
- Netzwerkaufzeichnungen

Für diese Informations-Quellen werden sogenannte Collectors definiert die dafür zuständig sind, diese Daten zu sammeln, beziehungsweise aufzuzeichnen und dann an das Framework zu übermitteln. Beispielsweise ist `Winlogbeat[elk_wb]` ein Dienst, der auf dem Test-System installiert wird und dafür zuständig ist relevante Log-Kanäle an das Framework zu übermitteln. Die Relevanz dieser Kanäle basiert auf der Fragestellung der Untersuchung und kann variabel angepasst werden. So können beispielsweise selbstausgewählte Providers einer ETW-Session hinzugefügt werden und an das Framework übermittelt werden. Im Folgenden sind relevante Log-Kanäle in Bezug auf die Telemetrie-Komponente aufgelistet:

- `System`

- `Microsoft-Windows-DNS-Client/Operational`
- `Microsoft-Windows-Winsock-AFD/Operational`
- `Microsoft-Windows-CAPI2/Operational`
- `Microsoft-Windows-WindowsUpdateClient/Operational`
- `Microsoft-Windows-Diagnostics-Performance/Operational`
- `Microsoft-Windows-Sysmon/Operational[ms_sysmon]`

Alle diese Kanäle, bis auf den Sysmon-Kanal, sind standardmäßig im Windows-Betriebssystem definiert und müssen eventuell noch aktiviert werden. Sysmon muss zusätzlich installiert werden, und liefert umfangreiche Details zur Prozesserstellung und Aufbau von Netzwerkverbindungen.

Ein weiterer Collector stellt `Molochcapture` dar, eine Komponente aus dem `Moloch`-Framework. `Molochcapture` zeichnet den gesamten Netzwerkverkehr an einer definierten Netzwerkschnittstelle (sogenanntes *Full Packet Capture*) des Frameworks auf. Wie später im Abschnitt der Test-Umgebung beschrieben wird über eine zentrale Netzkomponenten (also nicht das zu überwachende System selbst) gewährleistet, dass der Netzwerkverkehr der durch das Test-Systems erzeugt wird an die Netzwerkschnittstelle des Frameworks gespiegelt wird.

Vor Beginn jedes Test-Szenario wird eine eigene ETW-Session definiert, welche gesammelte Daten der Diagtrack-Komponente protokolliert. Diese gibt Aufschluss über die systeminterne Aktivität der Telemetrie-Komponente. Diese Daten müssen nicht mit den anderen Daten korreliert werden, welche im Framework gespeichert werden. Somit werden diese Daten lokal gespeichert und nur bei Bedarf beziehungsweise zur Bestätigung oder Widerlegung von Annahmen geprüft. Jedoch können solche benutzerdefinierten ETW-Sessions nach belieben auch in das Framework eingespeist werden.

Die Collectors speichern die gesammelten und aufgezeichneten Daten in der gemeinsamen Datenbank `ElasticSearch`. Um auf die gesammelten Daten nun zuzugreifen, werden sogenannte Viewer definiert. Mit den Viewern können komplexe Filter auf die Daten angewendet werden und so individuelle Visualisierung erstellt werden. Die Log-Kanäle werden dabei von `Kibana` dargestellt. `Kibana` ist Teil des Elastic-Stack und kann unterschiedlichste Datensätze darstellen. Der aufgezeichnete Netzwerkverkehr wird durch `Molochview` dargestellt. `Molochview` bietet dabei umfangreiche Filter- und Visualiserungsmöglichkeiten speziell für Netzwerkverkehr.

Leistung und Grenzen

Dieses Framework bietet die Basis für ein nahezu beliebig granulares automatisiertes Monitoring von Windows-Systemen. Die zentrale Speicherung von ausgewählten Windows Eventlogs ermöglicht dabei eine sehr tiefgreifende Analyse von Systemaktivitäten. Durch die umfangreiche Filterfunktion der Viewer lässt sich zielgerichtet nach einzelnen Events suchen und diese lassen sich visualisieren. Durch die zusätzliche Netzwerküberwachung werden Netzwerkverbindungen protokolliert und dabei Informationen über verwendete Protokolle oder Größe der übertragenen Daten geliefert. Die Kombination aus System- und Netzwerküberwachung lässt Aktivitäten einzelner Windows-Komponenten bis zur Netzwerkschicht sehr detailreich aufzeigen. Das Framework ist dabei auf die Analyse von Systemen/Komponenten ausgelegt, die nicht versuchen Aktivität zu verbergen (wie beispielsweise von Malware), da die Überwachung umfassend auf Informationen basiert, die durch das System selbst geliefert werden.

Test-Umgebung

In diesem Abschnitt wird die Test-Umgebung, das Test-System und die exemplarische Telemetrie-Kommunikation beschrieben. Die exemplarische Telemetrie-Kommunikation dient als Ausgangspunkt für die anderen Test-Szenarien. Die Testumgebung besteht aus dem Framework selbst, einem oder mehreren

Test-Systemen und einem Gateway zum Internet. Ein Switch verbindet die Test-Systeme mit dem Monitoring-Framework und dem Gateway. Der Switch ist dabei so konfiguriert das der Netzwerkverkehr der an den Switch-Port des Test-Systems passiert, an den Switch-Port, an den das Framework angeschlossen ist, gespiegelt wird. Dies ermöglicht dem Framework, dass der Netzwerkverkehr umfassend aufgezeichnet werden kann. In der folgenden Grafik sind die einzelnen sowie ihrer Verbindungen zu einander schematisch dargestellt.

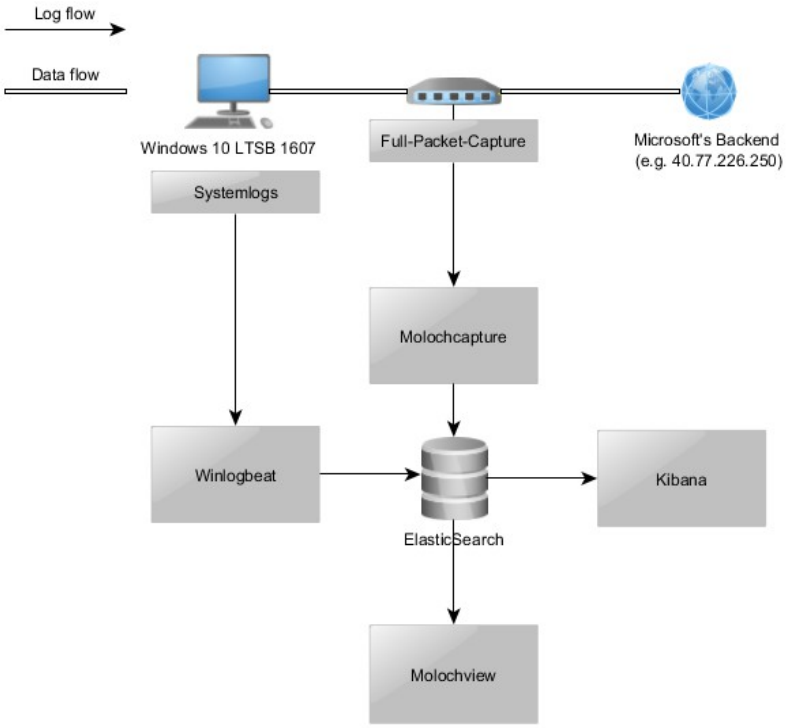

Abbildung 8: Architektur Monitoring Framework

Im Folgenden wird der Installations- beziehungsweise Provisionisierungsprozess der Test-Systeme dargestellt. Initial wird das Betriebssystem Windows 10 LTSB 1607 in Deutsch installiert. Während der Installation werden alle vordefinierten Einstellungen übernommen und anschließend das System vollständig gepatcht. Die getesteten Systeme haben einen Patchstand vom 01.02.2018. Anschließend wird das Test-System für das Test-Szenario vorbereitet: Dazu werden relevante Log-Kanäle in Bezug auf die jeweilige Fragestellung aktiviert oder definiert. `Winlogbeat` wird als Dienst auf dem Test-System installiert um die einzelnen Log-Kanäle an das Framework zu übermitteln. Im letzten Schritt wird eine benutzerdefinierte ETW-Session, im weiteren `UserTrace` genannt, erzeugt zu der verschiedene Provider in Bezug auf die Telemetrie-Komponente assoziiert werden. Diese benutzerdefinierte ETW-Session überwacht die Aktivität der Telemetrie-Komponente und kann als Indikator für gesammelte Informationen der Komponente dienen. Abschließend wird das System entsprechend des Test-Szenarios konfiguriert und für 48 Stunden überwacht. Als erstes wird beobachtet wie das Test-System mit dem Microsoft-Backend kommuniziert, wenn keine Konfigurationsänderungen an der Telemetrie-Komponente vorgenommen wurde . Die Telemetrie-Komponente kommuniziert mit den Hostnamen `settings-win.data.microsoft.com` und `*.vortex-win.data.microsoft.com`. Die folgende Grafik illustriert das Kommunikationsintervall und die übermittelten Daten. Der dunkelgraue Balken zeigt dabei die Größe der gesendeten Daten, während der hellgraue die Größe der empfangenen Daten darstellt. Diese Verbindungen werden in der Regel alle 20 bis 25 Minuten aufgebaut.

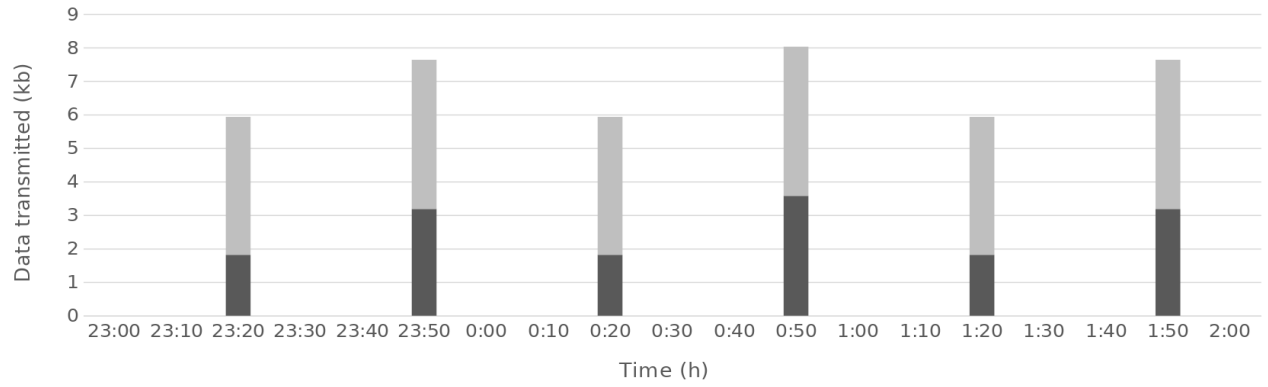

Abbildung 9: Kommunikationsintervall und übertragene Daten (Host: 40.77.226.250)

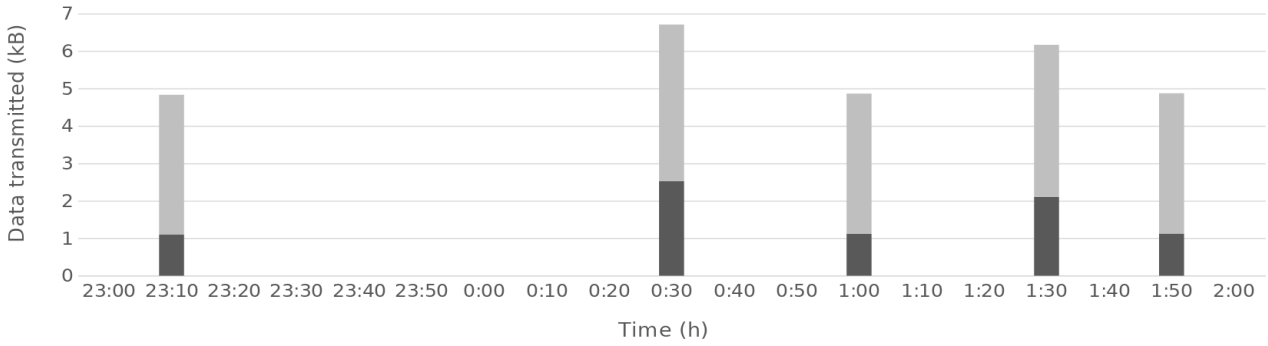

Abbildung 10: Kommunikationsintervall und übertragene Daten (Host: 40.77.226.249)

Beobachtungen

Die folgende Tabelle beschreibt die Beobachtungen der Telemetrie-Aktivität unter Konfiguration der unterschiedlichen Maßnahmen aus Kapitel 3 der Konfigurationsempfehlung zur Telemetrie:

Konfigurationsoption	Wirksamkeit	Verifikation
Konfiguration des niedrigst möglichen Telemetrie-Levels	Reduziert	Die Anzahl an Verbindungen reduzierte sich enorm. Im Testzeitraum wurde nur eine Verbindung pro Tag zu der IP-Adresse 40.77.226.250 aufgebaut. Im ganzen Test-Zeitraum wurde nur eine Verbindung zu dem Endpunkt mit der IP 40.77.226.249. Das Kommunikationsintervall hat sich von knapp 25 Minuten auf ca. 24 Stunden erhöht. Die Größe der übertragenen Daten hat sich nicht verändert.
Lokale DNS Einträge	Vollständig	Nach setzen der lokalen DNS-Einträge werden diese in den DNS Cache geschrieben. Versucht die Diagtrack-Komponente einen Verbindungsaufbau, initiiert diese eine DNS-Abfrage. Bevor diese jedoch an den DNS weitergeleitet wird, wird geprüft ob schon ein Eintrag zu diesem Hostnamen im DNS Cache vorhanden ist. Da dieser gesetzt wurde, erhält die DiagTrack-Komponente die Antwort 0.0.0.0 zurück. Da ein Verbindungsaufbau zu dieser Adresse nicht möglich ist, werden 2-3 erneute Anfragen an den DNS-Server beziehungsweise an den DNS-Cache gesendet.

Konfigurationsoption	Wirksamkeit	Verifikation
		Diese Informationen können über den Log-Kanal `Microsoft-Windows-DNS-Client/Operational` verifiziert werden. In den Netzwerkaufzeichnungen sind dementsprechend keine DNS-Abfragen oder Verbindungsaufbauten zu verdächtigen Endpunkten zu sehen.
Deaktivierung von Telemetrie-Dienst und ETW-Sessions	Vollständig	Weder DNS-Abfragen noch Verbindungsaufbauten konnten auf dem Netzwerk oder auf dem System selbst erkannt werden. Die benutzerdefinierte ETW-Session `UserTrace` bestätigt die Annahme, dass keine Daten im Kontext der Diagtrack-Komponente gesammelt worden sind.
Deaktivierung Telemetrie nach Microsoft Empfehlung	Vollständig	Im Test-Zeitraum konnte keine Netzwerkverbindung dem DiagTrack-Prozess zugeordnet werden. Es wurde in dem Zeitraum auch keine Daten in die benutzerdefinierte ETW-Session `UserTrace` geschrieben.
Lokale Firewall-Regeln	Vollständig	Durch die Isolierung des DiagTrack-Dienstes und der Blockierung durch die Firewall, können von dem Prozess, welcher durch den DiagTrack-Dienst gestartet wird, keine Netzwerkverbindungen aufgebaut werden. Wie schon bei der lokalen DNS-Konfiguration werden nach dem erfolglosen Verbindungsversuch 2-3 DNS-Abfragen zusätzlich gesendet.

Tabelle 10: Beobachtung der Empfehlungen

Erstellung lokale Firewall-Regel für Telemetrie-Dienst

Das folgende PowerShell-Skript enthält alle Befehle um einen Hardlink auf `svchost.exe` zu setzen:

```
# Besitzer der Datei aendern
$Account = New-Object -TypeName System.Security.Principal.NTAccount -ArgumentList 'VORDEFINIERT\
Administratoren';
$ACL = $null
$ACL = Get-Acl -Path C:\Windows\System32\svchost.exe
$ACL.SetOwner($Account)
Set-Acl -Path C:\Windows\System32\svchost.exe -AclObject $ACL
# Abfrage der Access Control Liste
$ACL = $null
$ACL = Get-Acl C:\Windows\System32\svchost.exe
# Zugriffsrechte setzen
$Ar = New-Object System.Security.AccessControl.FileSystemAccessRule($Account, "Write", "Allow")
$ACL.SetAccessRule($Ar)
Set-Acl C:\Windows\System32\svchost.exe $ACL
# Wechsle in Zielverzeichnis
Set-Location -Path C:\Windows\System32\
# Erstellung Hardlink
New-Item -ItemType hardlink -Name hard.exe -Value .\svchost.exe
```

```
# Anpassung der Registry

Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\DiagTrack" -Name "ImagePath" -Value
"%SystemRoot%\System32\hard.exe -k utcsvc"

# Hinzufuegen der Firewall Regel

New-NetFirewallRule -DisplayName "Block_Diagtrack" -Name "Block_Diagtrack" -Direction Outbound -
Program "%SystemRoot%\System32\hard.exe"
```

Referenzen

ms_dvc https://docs.microsoft.com/en-us/windows/configuration/configure-windows-diagnostic-data-in-your-organization [30/4/2018]
ms_etw https://msdn.microsoft.com/en-us/library/windows/desktop/aa363668(v=vs.85).aspx [30/4/2018]
ms_configdiag https://docs.microsoft.com/en-us/windows/configuration/configure-windows-diagnostic-data-in-your-organization [30/4/2018]
ms_dnsres https://technet.microsoft.com/en-us/library/bb962068.aspx#ID0ELHAC [30/4/2018]
elk_stack https://www.elastic.co/elk-stack [30/4/2018]
elk_wb https://www.elastic.co/downloads/beats/winlogbeat [30/4/2018]
ms_sysmon https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon [30/4/2018]

Abkürzungsverzeichnis