ERNW
Wir leben IT-Security

ERNW Newsletter 16 / April 2007

Liebe Partner, liebe Kollegen,

willkommen zur 16. Ausgabe des ERNW-Newsletters mit dem Thema:

Logging und Logauswertung im Windows-Umfeld als Stütze der IT-Sicherheitsarchitektur

Version 1.1 vom 16. April 2007

von: Friedwart Kuhn (fkuhn@ernw.de)

Dieser Newsletter gibt eine Überblick Logging und Logauswertung im Windows-Umfeld und beschreibt damit verbundene sicherheitsrelevante Aspekte.

INHALTSVERZEICHNIS

1 EINLEITUNG

Das Thema Logging und Logauswertung wird in den meisten Umgebungen unabhängig von der konkreten Implementierung nach wie vor stiefmütterlich behandelt. Dies, obwohl das Erfassen, Speichern und Verarbeiten von Logdaten heute einen kritischen Faktor innerhalb jeder IT-Sicherheitsarchitektur darstellt und gesetzliche Rahmenbedingungen, wie sie im KonTraG, in Basel II oder im Sarbanes-Oxley Act formuliert werden, das Vorhandensein von zuverlässigen Protokolldaten direkt oder indirekt verlangen – ganz abgesehen davon, dass im Verdachts- oder Schadensfall Protokolldaten unabdingbar für eine forensische Analyse sind. Hinzu kommt, dass das Thema Logging im Windows-Umfeld aufgrund des Microsoftschen Designs und der Implementierung hinsichtlich Konfiguration und Auswertung nicht immer besonders transparent ist und daher zu Verwirrung bei der Umsetzung konkreter Anforderungen führt. Der vorliegende Artikel will deshalb einen grundsätzlichen und zusammenfassenden Überblick über Logging und Logauswertung im Windows-Umfeld geben.

2 WINDOWS LOGS-BASICS

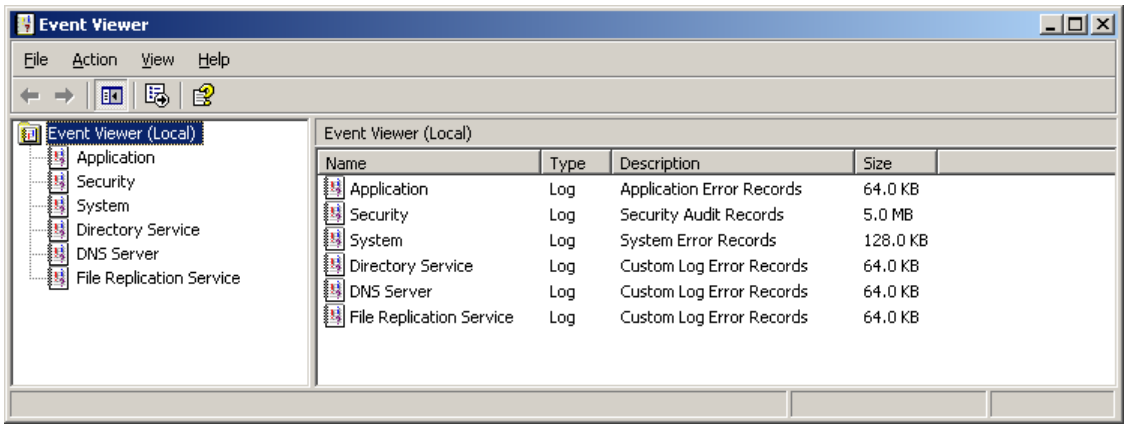
Ereignis-Protokolle

Die Betriebssysteme, die auf dem Kern von Windows NT basieren und über Windows Server 2003 bis zu Windows Vista reichen, kennen eine Ereignisanzeige, die abhängig von der Rolle des Rechners im Netzwerk ein bis sechs verschiedene Protokolle anzeigt. Der Einfachheit halber wird von den aktuellen Betriebssystemversionen, also von Windows XP, von Windows Server 2003 und von Windows Vista die Rede sein; eines dieser Betriebssysteme kann gemeint sein, wenn von „Windows“ gesprochen wird. Abweichungen von Windows Vista gegenüber dem hier Geschilderten finden im nachfolgenden Abschnitt eine eigene Beschreibung.

Jedes Windows verfügt über mindestens drei Protokolle in der Ereignisanzeige: Anwendungen wie z. B. der Exchange Server speichern ihre Ereignisse i. d. R. im Anwendungsprotokoll. Systemnahe Treiber und Dienste wie etwa der Anmeldedienst speichern ihre Ereignisse im Systemprotokoll. Sicherheits-bezogene Ereignisse wie z. B. der Fehlschlag einer Anmeldung werden – im Gegensatz zu den Einträgen in den beiden anderen Protokollen – erst nach erfolgter Konfiguration in das Sicherheitsprotokoll geschrieben. Handelt es sich um einen Windows-DNS Server, so findet sich in der Ereignisanzeige ein zusätzliches DNS Server-Protokoll. Ein Domänencontroller besitzt noch zwei weitere Protokolle, und zwar eines für den Verzeichnisdienst und ein weiteres für den Dateireplikationsdienst. In das Verzeichnisdienstprotokoll schreiben die (Active Directory-) Datenbankengine und wichtige Active Directory-spezifische Dienste wie der Knowledge Consistency Checker (KCC) oder der Inter-Site Topology Generator (ISTG). In dem Dateireplikationsdienst-Protokoll hinterlässt ausschließlich der für die Gruppenrichtlinien- und die Datei-Replikation verantwortliche Dateireplikationsdienst seine Einträge. Format der Protokolle ist das Windows-eigene .evt-Format. Eine Übersicht zusammen mit den Speicherorten zeigt die folgende Tabelle:

Protokoll	**Rolle des Rechners**	**Speicherort und Format** sind (nicht bei Vista) stets: %Systemroot%\System32\Config\
Application	Jede OS-Version von NT bis Server 2003 R2 (auch Vista)	Appevent.evt
Security	ebenso	Secevent.evt
System	ebenso	Sysevent.evt

Directory Service	Domänencontroller ab Windows 2000	NTDS.evt
DNS Server	DNS-Server ab Windows Server 2003	Dnsevent.evt
File Replication Service	Domänencontroller ab Windows 2000	Ntfrs.evt



Ereignisprotokoll eines Domänencontrollers, der gleichzeitig DNS Server ist

Darüber hinaus verfügt jeder Windowsrechner über eine Reihe von (bis zu mehreren hundert) Logdateien im Textformat, in denen Anwendungen oder einzelne Dienste, aber auch Patches und Hotfixes ihre Einträge schreiben. Ein Großteil dieser Logdateien entsteht während der Installation von Windows, ein weiterer entsteht bei jedem Update von Windows. Diese Dateien finden sich vor allem in %Systemroot% und tragen die Endung .log (es können jedoch auch .txt-Dateien sein, die im Namen die Buchstaben „log“ tragen, z. B. ntbtlog.txt). Einzelne Logdateien finden sich auch in %Systemroot%\Debug, %Systemroot%\System32\Config und %Systemroot%\System32.

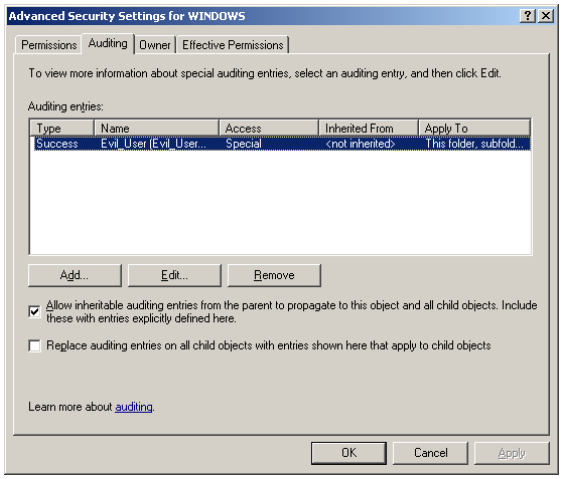
Zugriffsberechtigungen & Konfiguration

Alle Ereignisprotokolle bis auf das Sicherheitsprotokoll können per Default von (gewöhnlichen) Benutzern gelesen, jedoch nicht gelöscht werden. Das Sicherheitsprotokoll kann von Administratoren gelesen (und gelöscht) werden; eine Löschung zieht beim Sicherheitsprotokoll jedoch immer einen Eintrag darüber nach sich, welcher Benutzer das Protokoll wann gelöscht hat. Jedes Protokoll kann für sich nur als Ganzes, einzelne Einträge können nicht gelöscht werden.

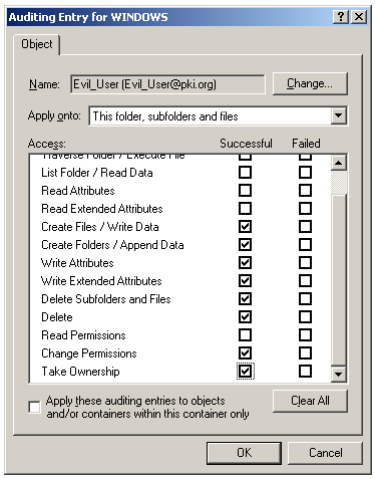
Ereignisse des Sicherheitsprotokolls werden erst protokolliert, nachdem erstens eine Überwachungsrichtlinie für die zu überwachende Kategorie konfiguriert wurde und zweitens, wenn es sich etwa um die Überwachung von Dateizugriffen handelt, auf der SACL (s. u.) des entsprechenden Objekts die zu überwachende Zugriffsart konfiguriert wurden. Die Überwachung der Verwaltung von Konten, der Verwendung von Privilegien oder der Veränderung von Gruppenrichtlinienobjekten (GPOs) ist ähnlich kompliziert. Sollen gezielt einzelne Aktionen im System überwacht werden, dann gestaltet sich die Konfiguration zum einen relativ kompliziert, zum anderen ist die Protokollierung nicht immer so genau, dass sich aus ihr ersehen lässt, welches Konto die in Frage stehende Veränderung vorgenommen hat. Dies betrifft beispielsweise die Veränderung von Gruppenrichtlinien auf Domänencontrollern. Da in Gruppenrichtlinien hochsensible Sicherheitskonfigurationen eingestellt werden, ist ein

genauer Nachvollzug für kritische Situationen wichtig – er lässt sich mit Windows-Bordmitteln jedoch nicht bewerkstelligen.

Das Einsammeln von Ereignissen auf anderen Windows-Computern ist mit der Ereignisanzeige (bis Windows Server 2003 R2) nicht möglich.



Überblick über die Überwachungseinträge



Einträge in einer SACL

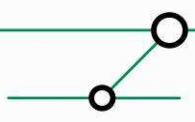
3 Gesetzliche Rahmenbedingungen & Manipulierbarkeit von Windows-Logs

Zweck jeglicher Protokollierung ist zunächst einmal und mindestens die Erfassung von atomaren technischen Vorgängen (in Form von Einträgen in einem Protokoll) für den Nachvollzug von Zugriffen, Konfigurationsänderungen, aber auch von alltäglichen Vorgängen wie etwa der Anmeldung am System. Bei den meisten Organisationen reicht die Forderung nach einem wie auch immer gearteten und nicht näher spezifizierten Nachvollzug nicht mehr aus, sondern die Protokollierung muss heute meistens ‚revisionssicher' sein. ‚Revisionssicher' kann dabei abhängig von der entsprechenden Organisationsrichtlinie unterschiedliche Bedeutungen annehmen: Sie kann einfach die Erfüllung der von der organisationsinternen Revision definierten Anforderungen bedeuten. Sie kann aber auch – und dies betrifft vor allem Unternehmen die im E-Commerce, im öffentlich-rechtlichen Bereich oder auf multinationaler Ebene operieren – die Erfüllung bestimmter rechtlicher Rahmenbedingungen wie etwa des KonTraG oder des Sarbanes-Oxley Acts bedeuten. So folgt etwa für die IT von Gesellschaften aus den Paragraphen §91, §289 HGB, §317 HGB des KonTraG im Wesentlichen, dass ein geeignetes System zur Überwachung der IT und ein geeignetes Risikomanagement betrieben werden muss. Daraus ergibt sich wiederum, dass geeignete Loggingmechanismen implementiert sein müssen. Der Abschnitt 404 (Section 404 (*Management Assessment Of Internal Controls)* des Sarbanes-Oxley Acts, der für alle nach der SEC gelisteten Unternehmen gültig ist (d. h. für alle börsennotierten Unternehmen in den Vereinigten Staaten), verlangt Dokumentation, Implementierung und Audit und damit insbesondere ein vertrauenswürdiges Loggingsystem. Die sich daraus ergebenden Anforderungen nach Integrität, Authentizität, Vertraulichkeit, Güte, Zuverlässigkeit und Vollständigkeit des oder der protokollierenden Systeme wird oft nicht erreicht.

Die diesbezüglichen Probleme sollen kurz umrissen werden:

Design des zugrundeliegenden Betriebssystems

Ein grundsätzliches Problem liegt in dem Design der Logginginfrastruktur und der sie umfassenden Komponenten: Die wenigsten Windows Betriebssysteme bieten fortgeschrittenere Zugriffskontrollmodelle wie MAC (Mandatory Access Control), bei denen jeder Betriebssystemkomponente ein bestimmter Integritätslevel und eine damit verbundene



Zugriffsstufe zugeordnet wird. Microsoft beschreitet erst mit Windows Vista diesen Weg (in Form von Mandatory Integrity Control (MIC)).

Vertraulichkeit der innerhalb der Logginginfrastruktur übertragenen Daten

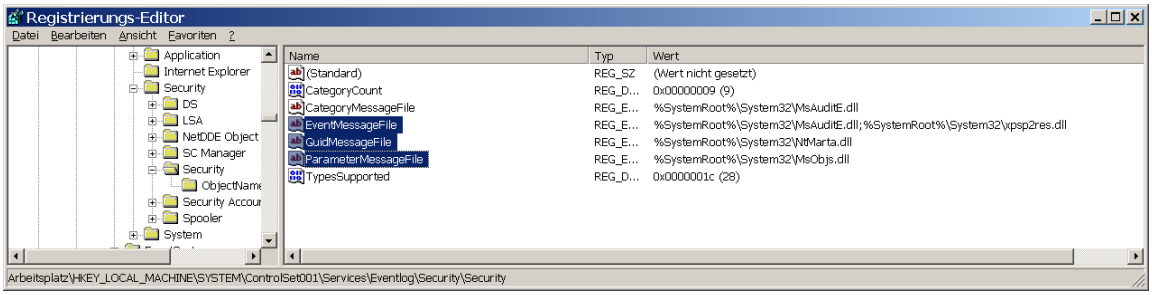
Die Absicherung des Kommunikationskanals der verschiedenen Loggingkomponenten etwa durch die Implementierung kryptografischer Methoden ist bisher wenig verbreitet.

Privilegienfülle von administrativen Konten

Administrative Konten verfügen i. d. R. über nicht oder nur sehr kompliziert einschränkbaren Vollzugriff auf nahezu alle Betriebssystemkomponenten und damit auch auf die Protokolle.

Ganz konkret eröffnen sich bei Windows Betriebssystemen bis Windows Server 2003 R2 folgende Manipulationsmöglichkeiten. Die Ereignisprotokollierung wird einfach ‚abgeschaltet': Der Dienst kann zwar nicht einfach beendet werden, sein Startup kann jedoch auf "deaktiviert" gesetzt werden. Nach einem Neustart des Rechners wird dann gar nicht mehr protokolliert.

Eine subtilere und technisch anspruchsvollere Methode stellt die Veränderung von Windows-‚Schablonen-dll' dar. Schablonen-dll stellen bei Windows eine Zuordnung von (Fehler-) Code zu einem lesbaren Text her, denn Windows speichert in den Ereignisprotokollen aus Platzgründen vor allem (Fehler-) Codes und Variablen und keinen erklärenden Text. Der erklärende Text zu einem Code findet sich in einer dll. Die wichtigste dieser Schablonen-dll ist die Datei msaudite.dll. Wird diese Datei – z. B. mit einem Expeditor – verändert oder wird in der Registrierung der Pfad, der auf diese Datei verweist, verändert, dann können ‚benutzerdefinierte' Texte zu Codes erzeugt werden. Ein solcherart verändertes System wird zwar nach wie vor Protokollieren, jedoch ist dieses Protokoll nicht mehr vertrauenswürdig.



Schablonen-dll in der Registrierung

Darüber hinaus findet sich auf http://www.ntsecurity.nu/toolbox/winzapper/ das Tool Winzapper, mit dem sich die Einträge des Sicherheitsprotokolls unter Windows NT 4.0 und Windows 2000 bearbeiten lassen. Unter Windows XP und Server 2003 ist dieses Tool bisher nicht lauffähig, doch es ist davon auszugehen, dass das Erscheinen von weiteren und potenteren Tools zur Manipulation des Ereignisprotokolls nur eine Frage der Zeit ist.

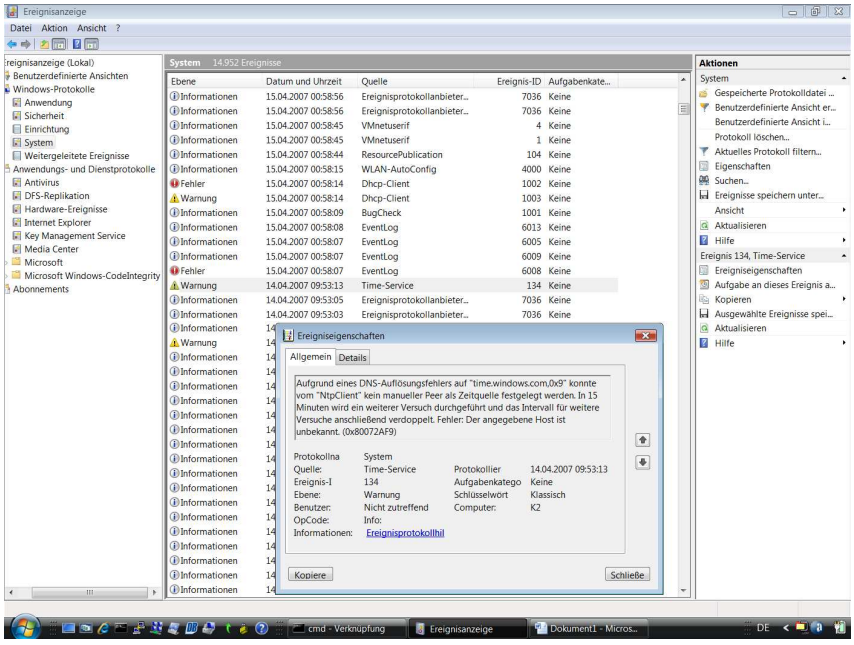
4 WINDOWS VISTA

Auch Windows Vista kennt die Ereignisanzeige mit den drei klassischen Protokollen, allerdings ist schon die Präsentation der Daten anders als bei den Vorgängerversionen, denn als Grundlage dient Windows Vista die Verwaltungskonsole in der Version 3.0 (MMC 3.0). Windows Vista bietet grundsätzlich ein wesentlich granulareres Logging als Windows-Vorgängerversionen, und zwar gibt es bei Vista zum einen neue Logging-Kategorien, zum anderen werden Ereignisse i. d. R. deutlich ausführlicher kommentiert.

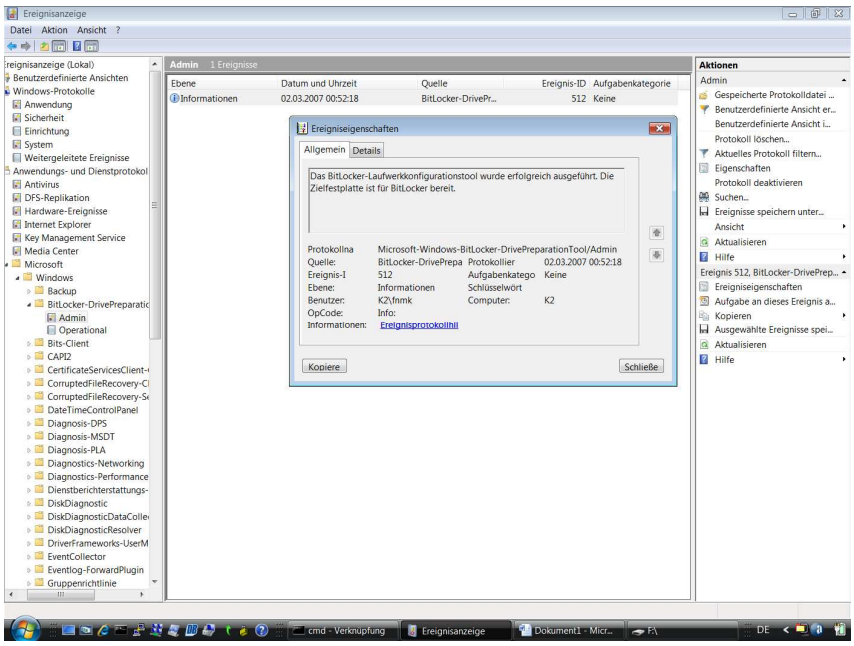
Die klassischen Ereigniskategorien finden sich unter Windows Logs, dazu kommen an dieser Stelle zwei neue Kategorien, nämlich „Einrichtung“ und „Weitergeleitete Ereignisse“. Unter „Einrichtung“ schreiben Vista-fähige Applikationen (künftig) ihr Installationsprotokoll, unter „Weitergeleitete Ereignisse“ lassen sich Ereignisse anderer Computer sammeln. Diese Eigenschaft bedeutet eine wichtige Neuerung, denn ein Vista-Computer kann damit eine Loghost-ähnliche Funktionalität erfüllen. Darüber hinaus kennt Windows Vista unter dem Knoten „Microsoft – Windows“ ein Protokoll für jeden unter Vista registrierten Dienst. Damit kann ein Monitoring einzelner Dienste erfolgen. Es ist zu erwarten, dass Software, die für Vista geschrieben wurde unter einem Knoten mit ihrem Namen ein eigenes Protokoll erstellen wird.

Windows Vista beherrscht die regelbasierte Reaktion auf die von einem Administrator dafür ausgewählten Ereignisse; so kann als Reaktion auf ein solches Ereignis ein Programm gestartet werden oder eine Regel definiert werden, die ein Programm über den Task Scheduler einbindet.

Des Weiteren haben sich Speicherorte und Formate geändert. Die klassischen Protokolle liegen bei Vista unter %Systemroot%\System32\winevt\Logs und tragen die Erweiterung .evtx. Nomen et Omen: Das Format der Protokolldateien ist bei Vista XML und damit endlich plattformübergreifend ohne großen Aufwand weiter verarbeitbar.



Windows Vista Ereignisanzeige: mehr Kategorien und ausführlichere Ereignisbeschreibungen



Vista bietet Protokolle zu einzelnen Diensten (hier am Beispiel von BitLocker)

5 GRUNDLEGENDE ASPEKTE EINER LOGGINGINFRASTRUKTUR

Wenn eine Logginginfrastruktur implementiert werden soll, dann sind einige grundlegende Aspekte zu berücksichtigen:

- **Größe und (hierarchische) Struktur der Organisation**

  Hier gilt es zu klären, ob – da hier das Logging im Windowsumfeld betrachtet wird – es verschiedene Logging-Domänen gibt und ob diese dem zugrundeliegenden Active Directory-Design entsprechen sollen, oder ob ein von der bestehenden Gesamtstruktur abweichendes Design realisiert wird. Aufgrund der in einer Gesamtstruktur bestehenden Rechtestruktur ist es empfehlenswert, das bestehende Active Directory-Design als Ausgangspunkt für die zu implementierende Logginginfrastruktur zu verwenden. Unter diesen Punkt fällt auch die Klärung, wie das Design einer z. B. UNIX-basierten Logginginfrastruktur in die Windowsumgebung einzubinden ist.
- **Sicherheitsanforderungen an die Logginginfrastruktur**

  Unter diesem Punkt gilt es zu klären, ob etwa die übertragenen Daten verschlüsselt werden müssen und wie der administrative Zugriff auf die Logginginfrastruktur und ihre Komponenten gemäß den Sicherheitsanforderungen an die Infrastruktur definiert werden.
- **Erwünschter Automatisierungsgrad bei der Informationssammlung- und –auswertung**

  Der Automatisierungsgrad bei der Informationssammlung und –auswertung bestimmt maßgeblich den Aufwand von administrativen Tätigkeiten bei der Verwaltung der Logginginfrastruktur. Die Möglichkeiten reichen hierbei von manueller Sammlung und Auswertung bis zur Anschaffung von Software, die einen Großteil der anfallenden Tätigkeiten automatisiert erledigt.

- **Homogenität /Heterogenität der IT-Umgebung**
  Eine wichtige Rolle bei der Auswahl einer Lösung spielen die Homogenität oder auch die Heterogenität der Umgebung. In einer reinen Windowsumgebung etwa wird die Zusammenarbeit von Syslogkomponenten mit Windowsprotokollsoftware i. d. R kaum eine Rolle spielen. In heterogenen Umgebungen kann – z. B. wenn eine zentrale Verwaltung gewünscht ist – Software benötigt werden, die zwischen Windows- und UNIX-Protokollen vermittelt.
- **Vorhandenes Know How**
  Da vorhandenes Know How bestmöglich genutzt werden sollte, wird sich die Implementierung i. d. R. – Outsourcing ausgenommen – danach zu richten haben. In homogenen Windowsumgebungen wird Syslog eher sporadisch oder gar nicht eingesetzt werden.
- **Outsourcing**
  Outsourcing von Teilen der IT-Umgebung spielt heute bei der Behandlung von Kosten- und Effizienzfragen i. A. eine wichtige Rolle. Das Outsourcing von Komponenten, die die IT-Sicherheit betreffen sollte jedoch besonders gründlich bedacht werden. Hat eine Organisation gesetzliche Rahmenbedingungen zu wahren, dann ist dies in den entsprechenden SLAs zu berücksichtigen.

6 LÖSUNGEN

Bei der Wahl einer Logginglösung spielt heute vor allem die Automatisierung bei der Informationssammlung, -präsentation und –auswertung eine Rolle. Wenn man von Automatisierung spricht, so gilt es zu unterscheiden:

- **Automatisierungskategorien**
  Automatisierung kann sich auf unterschiedliche Vorgänge, die automatisiert werden, beziehen. Z. B. auf die Informationsauswertung.
- **Automatisierungsgrade**
  Automatisierung kann in verschiedenen Graden /Stufen erreicht werden von manuell bis (voll) automatisiert.

6.1 Automatisierungskategorien

Automatisierungskategorien beziehen sich in einer Logginginfrastruktur meisten auf:

- **Informationssammlung**
  - Einsammlung der Informationen aus den unterschiedlichen Ereignisprotokollen und sonstigen Logs der Rechner /Devices
- **Informationspräsentation**
  - als reine Textdatei unformatiert (z. B. Eventquery), häufig .csv-Format (z. B. Logparser)
  - als HTML-Datei (z. B. Logparser; Frontend einer SQL-Datenbank)
  - über ein eigenes GUI /eigene Verwaltungskonsole (z. B. MOM)
- **Informationsauswertung**
  - findet in einfachen Tools nicht statt (z. B. Eventquery)
  - findet bei diversen Softwarelösungen statt (z. B. MOM, Event Manager)
  - Ereignisse können mit einer Bewertung verknüpft werden (z. B. MOM, Event Manager)
  - Ereignisse können gefiltert werden (z. B. MOM, Event Reporter)
  - Ereignisse können mit Aktionen verknüpft werden (Event Reporter, Event Manager)

6.2 Automatisierungsgrade

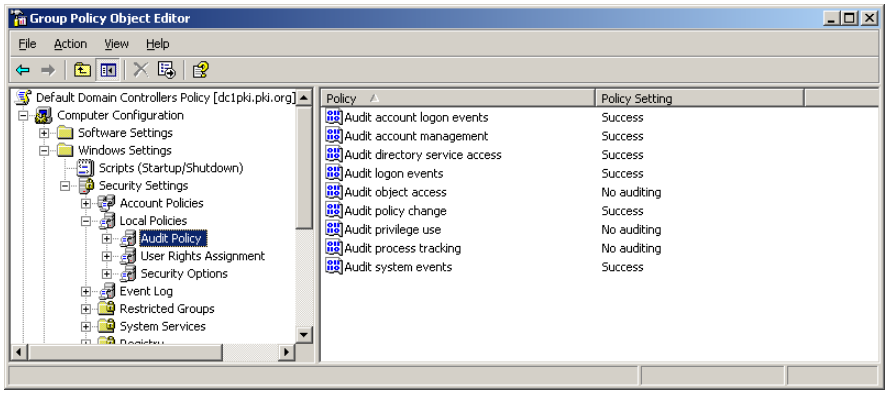
Es lassen sich die folgenden Automatisierungsgrade /-stufen unterscheiden:

- **Manuell**
  - Individuelle Ansicht & Auswertung der Windows-Ereignisprotokolle und sonstiger Logs
- **Halbautomatisiert**
  - Verwendung von meistens kommandozeilen basierten Tools für eine ansatzweise Automatisierung
  - Grenzen zur Vollautomatisierung sind fließend
- **(Voll-) Automatisiert**
  - Daten werden automatisch gesammelt und (meistens über eine eigene Konsole) präsentiert

6.3 Manuelle Lösungen

Manuell & individuell (pro Rechner)

Sollen Protokolle manuelle und individuell (pro Rechner) ausgewertet werden, so geschieht dies in der (klassischen) Ereignisanzeige, die bis zu Windows Server 2003 das .evt-Format (proprietär) und ab Windows Vista das .evtx-Format (XML) besitzt. Bevor jedoch Ereignisse, die über das recht karge Default-Logging hinaus gehen, aufgezeichnet werden, muss in den Überwachungsrichtlinie konfiguriert werden, welche Kategorien (z. B. Kontenverwaltung oder Verzeichniszugriff) protokolliert werden sollen. In der jeweiligen Objekt-SACL (siehe Abbildung in Abschnitt 2) muss dann bestimmt werden, welche Zugriffsebene protokolliert wird. In Active Directory-Umgebungen besteht wegen der Verknüpfungsmöglichkeiten von GPOs dabei eine recht hohe Flexibilität zu bestimmen, welche Ereignisse auf welchen (Windows-) Rechnern aufgezeichnet werden.



Überwachungsrichtlinie

Des Weiteren kann der Zugriff auf Registrierungsschlüssel entweder im Sicherheitsprotokoll oder „live“ über das Tool Regmon nachverfolgt werden. Windows bietet darüber hinaus eine Reihe von textbasierten-Protokollen zu vielen verschiedenen Komponenten wie etwa ein Protokoll, das alle während der ersten Startphase geladenen Treiber listet (ntbtlog.txt in %Systemroot%, aktiviert über den Schalter „/bootlog“ in der boot.ini) oder zwei Protokolle, die Installation und Konfiguration eines Domänencontrollers aufzeichnen (dcpromo.log und dcpromoui.log in %Systemroot%\debug) – um nur zwei von vielen Beispielen zu nennen.

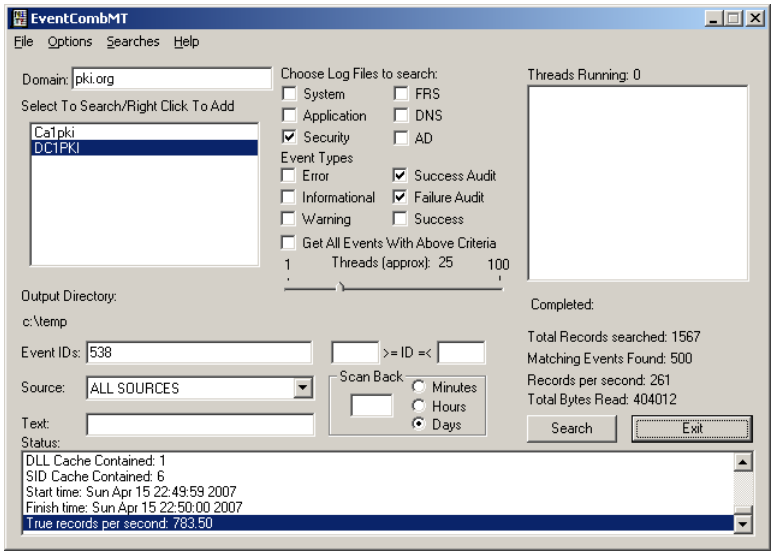
Manuell und zentralisiert in kleinen Umgebungen

Das Snap In für die Ereignisanzeige kann remote ausgeführt werden, so dass in kleinen Umgebungen – administrative Berechtigungen vorausgesetzt – Ereignisprotokolle mehrer Rechner in einer Konsole angezeigt werden können. Solch ein Verfahren bietet sich an, wenn man sich einen schnellen Überblick etwa über die Systemprotokolle aller Domänencontroller verschaffen möchte.

Dumpel und EventCombMT

Dumpel.exe ist ein kommandozeilenbasiertes Tool aus dem Windows 2000 Resource Kit (auch lauffähig unter Server 2003) mit dem die Ereignisprotokolle eines lokalen oder Remote-Rechners ausgelesen werden können. Die Ausgabe erfolgt in eine (tabstopp-getrennte) Textdatei.

EventCombMT.exe ist ein GUI-Tool – das in den Windows-Resource Kits enthalten ist – zum Durchsuchen der Ereignisprotokolle nach definierbaren Ereignissen über beliebige Mitgliedsrechner einer Domäne hinweg. Die Suche erfolgt dabei von dem Rechner aus, auf dem EventComMT ausgeführt wird. Dieses Tool ist besonders gut geeignet, um Ereignisprotokolle nach einer bestimmten Ereignisse-ID auf bestimmten Rechnern zu suchen. Die Ausgabe erfolgt in eine Textdatei.



EventCombMT

6.4 Halbautomatisierte Lösungen

Bei halbautomatisierten Lösungen werden meistens kommandozeilen-basierte Tools oder einzelne Befehle mit Scripts und /oder Geplanten Tasks kombiniert und dadurch automatisiert. Beispiele hierfür sind die Befehle „eventquery“, „eventtriggers“ und „eventcreate“. Bei diesen Befehlen handelt es sich um vbs-Scripte, die einen lauffähigen und voreingestellten Scripthost benötigen. Die Befehle besitzen folgende Funktionen:

- **eventquery**: damit kann das lokale oder das Remote-Ereignisprotokoll durchsucht werden und Ereignisse können gefiltert werden
- **eventtrigers**: damit kann das lokale oder ein Remote-Ereignisprotokoll hinsichtlich definierter Ereignisse überwacht werden, die bei Eintreten als Auslöser für Befehle oder Tasks fungieren können
- **eventcreate**: damit können benutzerdefinierte Ereignisse und Ereigniskategorien im Ereignisprotokoll erstellt werden

Die Ausgabe dieser Befehle kann für eine Weiterverarbeitung in Textdateien erfolgen. So wird etwa mit dem Befehl:

eventtriggers /create /tr "DNS AD Fix" /so "DNS" /eid 4004 /tk c:\admin\scripts\dnsadfix.bat

ein Auslöser erstellt, der das DNS-Server-Ereignisprotokoll auf Ereignisse mit der Quelle "DNS" und der Ereignis-ID 4004 überwacht. Beim Eintreten dieses Ereignisses wird die Batchdatei "dns-adfix.bat" ausgeführt.

Ein weiteres Tool, das zum Einsatz als halbautomatisierte Lösung geeignet ist, ist **Logparser**.exe (siehe www.logparser.com). Hierbei handelt es sich um ein Kommandozeilen-basiertes Tool das SQL-Abfragen gegen eine Vielzahl von Microsoft-spezifischen Protokollen stellen kann und die Ergebnisse in einer Vielzahl von Ausgabeformaten darstellen kann. Insbesondere eignet sich das Tool zum Sammeln, Filtern, Darstellen und Exportieren von Ereignissen. Eingabeformate können u. a. sein: Windows-Ereignisprotokolldateien, IIS-Protokolldateien, div. Textformate (.csv, .tsv, .w3c, .xml u. a.), Metadateien zu Active Directory-Objekten, Netmon-Capturedateien.

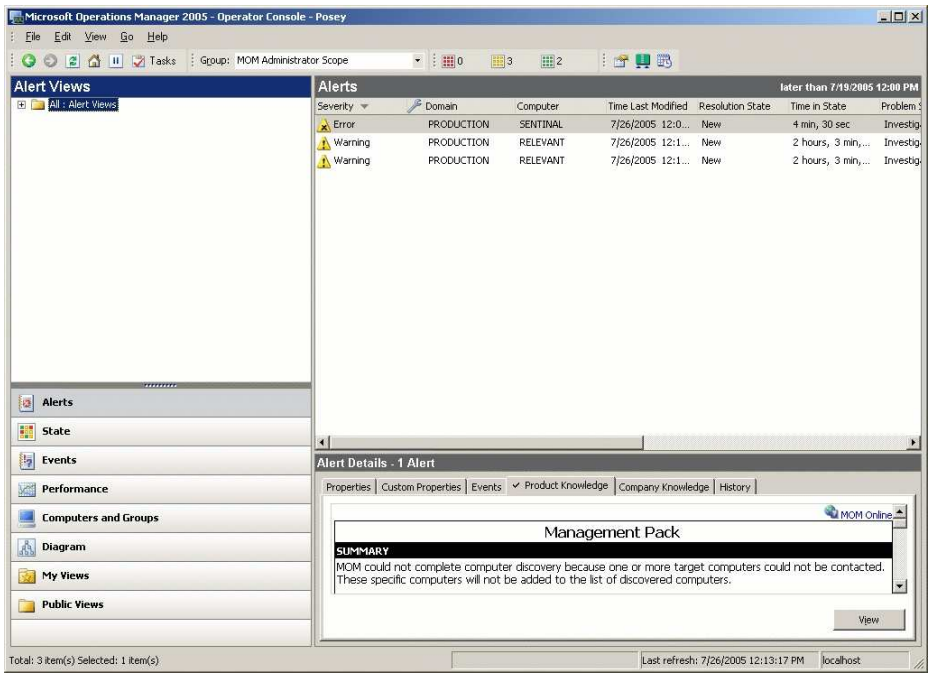
Ausgabeformate können u. a. sein: Textformate (.csv, .tsv, .w3c, .xml u. a.), entsprechend Syslog-Standard, Hochladen in SQL-Datenbank, Erstellen Excel-ähnlicher Diagramme. Durch die vielen unterstützten Formate ist Logparser sehr vielseitig einsetzbar.

6.5 (Voll-) Automatisierte Lösungen

Bei diesen Lösungen handelt es sich häufig – aber nicht ausschließlich – um kommerzielle Lösungen, die i. d. R. bis auf den Enterprise-Bereich skalierbar sind. Exemplarisch seien zwei Vertreter der kommerziellen Lösungen genannt: **Microsoft Operations Manager 2005** und **GFI Events Manager (V.7)**. Das Microsoft-Produkt ist in vielen größeren und großen Windowsumgebungen das Standardprodukt zur automatisierten Sammlung, Präsentation und Auswertung von Ereignisprotokollen und anderen nicht über die Ereignisanzeige erfassten Daten. MOM 2005 gibt es dabei auch in einer kleineren „Workgroup Edition“, die für Arbeitsgruppen-Umgebungen bis 10 Server geeignet ist. MOM 2005 bietet dabei u. a. der Übersicht wegen stichpunktartig aufgeführt:

- Ausführliche Berichterstattung über sprichwörtlich hunderte von Sensoren mit Zusammenfassungen (Management-Berichterstattungen), Trendanalysen für Windows Betriebssysteme
- Management-Packs insbesondere für Microsoft-Serverapplikationen (Exchange-, ISA-, SQL-Server) und für viele Dritthersteller
- Unterstützung der gängigen Protokollformate im Windows-Bereich (.evt und W3C) und über Windows-Bereich hinaus (Syslog) über Management Packs.
- Eine umfangreiche Wissensdatenbank, die Best Practices gemäß ITIL integriert

Es gibt derzeit rund 195 Management Packs für MOM 2005, über die praktisch alle gängigen UNIXe eingebunden werden können.



MOM 2005 (Alarmansicht).

Von der Funktionalität sehr ähnlich ist GFIs Events Manager (V.7). Er unterscheidet sich vom Microsoft-Produkt vor allem durch eine geringere Anzahl von Sensoren, er verfügt zwar auch über eine Wissensdatenbank, jedoch ohne ITIL-gemäße Best Practices-Empfehlungen. GFIs Events Manager unterstützt per Default – d. h. ohne Erweiterungen in Form von Management Packs – die drei gängigen Protokollformate .evt, W3C und Syslog.

Syslogbasierte Lösungen

Von den kommerziellen Drittersteller-Lösungen sind vor allem die nicht-kommerziellen Syslog-Lösungen zu unterscheiden.[1] Dabei gibt es für Windows sowohl Syslog-Server als auch Syslog-Clients. Als eine bekanntere nicht-kommerzielle Syslog-Lösung sei der **Kiwi Syslog Daemon (V. 8.2.8)** genannt (www.kiwi.com). Mit ihm können einfache Protokollsammel-, Darstellungs- und Filterfunktionen ausgeführt werden. Die lizensierte kommerzielle Version wartet darüber hinaus mit einer Reihe verfeinerter Filterfunktionen und mit regelbasierten Aktionen auf. Was die Syslog-Software der Open Source-Gemeinde betrifft, so ist es üblich, dass eine kostenlose Version mit eingeschränkter Funktionalität neben einer kostenpflichtigen mit erweiterter Funktionalität angeboten wird. Als weitere Serverlösung sei **WinSyslog** erwähnt. Diese Lösungen bieten in keinem Fall den Umfang der vorgenannten kommerziellen Lösungen, dafür sind sie deutlich preisgünstiger.

Als recht umfangreiche und den Sicherheitsanforderungen an eine Logginginfrastruktur entgegenkommende Lösung sei Balabits Syslog next generation[2]-Implementierung **syslog ng PE** genannt (www.balabit.com). Da Syslog-Meldungen als Protokollinformationen sensible

[1] *Syslog ein Quasi-Standard zur Übertragung von Log-Nachrichten. Syslog steht im engeren Sinne für das Übertragungsprotokoll selbst; im weiteren Sinne die Anwendung, die Log-Nachrichten sendet und /oder empfängt (über UDP-Port 514).*

[2] *Syslog ng ist die Weiterentwicklung Syslog und in vielen aktuellen Linuxen enthalten.*

Informationen enthalten können, verschlüsselt die Premium Edition von syslog-ng alle Übertragungen per SSL/TLS. Zudem ermöglicht syslog-ng PE die gegenseitige Authentifizierung von Host und Server über X.509-Zertifikate. syslog-ng PE kann darüber hinaus alle eingehenden Meldungen nach Quell-Host, Anwendung und Priorität sortieren und Windows Serveranwendungen integrieren. Umfangreiche Filterfunktionen und regelbasierte Aktionen ermöglichen es, auf vielfältige Art und Weise auf Syslog-Meldungen zu reagieren.

Weitere Syslog-Server finden sich unter:

- *http://www.loganalysis.org/sections/syslog/syslog-nonunix/*
- *http://www.practicallynetworked.com/support/syslog.htm*

Syslog-Clients für Windows gibt es eine ganze Reihe. Sie sind meistens in Form von Agents für Windows realisiert, die Protokollinformationen auslesen, in das Syslog-Format konvertieren und an einen Loghost weiterleiten. An erster Stelle sei der mit den Services for Unix von Windows Server 2003 R2 (SFU 3.5) mitgelieferter Syslog-Daemon genannt. Bekannte Syslog-Agents aus dem Open Source-Umfeld sind: Kiwi Logger, NTSyslog und EventReporter. Eine recht umfassende Liste von Syslog-Clients findet sich unter:

- http://www.loganalysis.org/sections/syslog/windows-to-syslog/

6.6 Logformate und -konvertierungen

Bei der Weiterverarbeitung von Protokollen kann die Frage, welches Format das Protokoll besitzt, eine nicht zu unterschätzende Rolle spielen. Grundsätzlich wünschenswert, in aktuellen Windowsumgebungen aber noch nicht durchgängig realisierbar, ist eine Darstellung in einem plattformunabhängigen Format.

Folgende gängige Protokollformate werden in aktuellen Windowsumgebungen vor allem verwendet:

- .evt (Windows-proprietär)
- .evtx (XML ab Windows Vista)
- W3C-Format, ASCII-textbasiert, (z. B. Default-Log-Format in IIS 6.0)
- Syslog (Textformat mit Leerzeichen)
- (einfaches) Textformat (div. Windows Logdateien, z. B. ntbtlog.txt)

Bei der Weiterverarbeitung der Logdateien in heterogenen Umgebungen spielt die Konvertierung eine wichtige Rolle. Es gibt die folgenden Konvertierungsmöglichkeiten:

- **.evt => div. Formate** wie Textformat, HTML-Format, XML-Format: Dies wird über professionelle kommerzielle Software wie MOM und Events Manager erledigt und findet vollkommen automatisiert statt.
- **.evt => Textformat**
  - dumpel.exe: Konvertierung geschieht manuell, automatisierbar durch Verskriptung
  - EventCombMT.exe: Konvertierung geschieht manuell
  - div. Syslog-Daemons: Protokollierung /Konvertierung erfolgt immer in eine Textdatei
- **Syslog => .evt**
  - auch das ist möglich über einen Syslog-Wrapper (kein ausgereiftes Tool)
  - http://www.loganalysis.org/sections/syslog/syslog-nonunix/
  - kann auch über Management Packs des MOM erledigt werden.

Was die Konvertierung von des .evt-Formats in das Textformat betrifft, so ist anzumerken, dass weder dumpel noch EventCombMT geschrieben wurden, um Protokolldateien für

Weiterverarbeitung im Textformat in dieses zu konvertieren; gleichwohl lassen sie sich in eingeschränktem Maße dazu verwenden.

Für weitere Fragen steht Ihnen das Team von ERNW-Deutschland und ERNW-Portugal gern zur Verfügung.

Mit freundlichen Grüßen,

Friedwart Kuhn.

ERNW GmbH
Friedwart Kuhn
Senior Security Consultant

ERNW Enno Rey Netzwerke GmbH
Breslauer Str. 28
69124 Heidelberg
Tel. +49 6221 480390
Fax +49 6221 419008
Mobil +49 15152411855
Portugal: +351 91 8763637
www.ernw.de
info@ernw.de