

Wie jedes andere Betriebssystem ist auch Windows-NT nicht vor Systemabstürzen gefeit. Es arbeitet zwar in der Regel wesentlich stabiler, als die in die Jahre gekommene DOS-Windows-Kombination, jedoch können fehlerhafte Programmteile auch unter NT zu fatalen Folgen führen, die den gesamten Rechner zum Stillstand bringen. Anders als früher bietet Windows-NT aber vergleichsweise weit reichende Möglichkeiten, der fehlerhaften Komponente auf die Spur zu kommen.

DIRK PELZER

Bevor jedoch auf diese Möglichkeiten eingegangen wird, ist ein kleiner Exkurs in die NT-Architektur notwendig. Windows-NT unterscheidet grundsätzlich zwei Modi. Das sind zum einen der Benutzermodus (User Mode) und zum anderen der Kernel-Modus (Kernel Mode). Der Benutzermodus beinhaltet die verschiedenen Subsysteme von Windows-NT, wie zum Beispiel das Win32- oder das POSIX-Subsystem. Hier laufen auch alle Anwendungen, wie zum Beispiel ein SQL-Server oder Textverarbeitungs-Programme. Im Gegensatz dazu laufen im Kernel-Modus alle Systemdienste, wie beispielsweise der I/O-Manager, der unter anderem für Dateisysteme, Netzwerktreiber und weitere Komponenten verantwortlich ist. Der Bereich, in dem der Kernel liegt, ist für Anwendungen des Benutzermodus nur indirekt über eine definierte Programmierschnittstelle (API) ansprechbar. Auf diese Weise wird zumindest theoretisch verhindert, daß Anwendungen, die im Benutzermodus laufen, Komponenten im Kernel-Modus zu Absturz bringen können. Dadurch wird erreicht, daß NT auch dann stabil weiter laufen kann, wenn eine oder mehrere Anwendungen im Benutzermodus Probleme bereiten. Im Benutzermodus sind fehlerhafte Komponenten vergleichsweise einfach zu ermitteln, da man als Anwender sofort mit bekommt, wenn sich eine

Anwendung, die man gestartet hat, mit einem Anwendungsfehler verabschiedet. Das Betriebssystem selbst bleibt jedoch davon unberührt und läuft weiter. Im Gegensatz dazu können fehlerhafte Komponenten, die im Kernel-Modus arbeiten, wie zum Beispiel SCSI- oder Netzwerk-Treiber, dazu führen, daß ein NT-System komplett abstürzt und nur durch einen Kaltstart wieder belebt werden kann. Auch erschließt sich dem Anwender hier nicht sofort, welche Komponente für den Absturz verantwortlich war.

Wenn ein solcher Kernel-Modus-Fehler auftritt, äußert sich dieser mit einem sogenannten »Bluescreen«, kurz auch BSOD (»Bluescreen of Death«) genannt. Dieser tritt beispielsweise dann auf, wenn ein Treiber versucht, den Speicherbereich einer anderen Systemkomponente zu schreiben. Das Problem ist, daß sich solche Systemabstürze in der Regel nicht ankündigen, sondern einfach passieren. Die gängige Methode im Eventlog von NT nach Fehlerursachen zu forschen, verläuft in solchen Fällen meist ergebnislos. Der erste und zugleich am schwierigsten zu entziffernde Hinweis auf ein solches Problem ist der Bluescreen. Er enthält jedoch für den Eingeweihten sehr viele wertvolle Hinweise darauf, welcher Fehler aufgetreten ist und von welcher Komponente er verursacht wurde.

```

*** STOP: 0x0000000A (0x00000000, 0x0000001a, 0x00000000, 0xfe5000df)
IRQL NOT LESS OR EQUAL
p4-0300 irq1:lf  SYSVER:0xf000030e

dll Base DateStamp - Name
80100000 2e93fe95 - ntoskrnl.exe
80010000 2e41804b - bhad154x.sys
8001b000 2e4e7b6b - Scsidisk.sys
fe420000 2e406607 - Floppy.SYS
fe440000 2e406659 - Es Rec.SYS
fe460000 2e406654 - Bcap.SYS
fe480000 2e42a4a4 - i8042prt.SYS
fe4a0000 2e40660c - Kbdclass.SYS
fe4b0000 2e53d42d - atapi.SYS
fe4e0000 2e406655 - Msfs.SYS
fe510000 2e53f222 - NDIS.SYS
fe550000 2e406697 - TDI.SYS
fe580000 2e527949 - nlnkspk.sys
fe5b0000 2e494973 - tcpip.sys
fe5d0000 2e5279d3 - netbt.sys
fe5e0000 2e4066b3 - mup.sys
fe630000 2e53f24a - srv.sys

dll Base DateStamp - Name
80400000 2e93b8a6 - hal.dll
80013000 2e4bc29a - SCSIPORT.SYS
80220000 2e83f238 - Ntfs.sys
fe430000 2e406618 - ScsiCdm.SYS
fe450000 2e40660f - Null.SYS
fe470000 2e406634 - Smmouse.SYS
fe490000 2e40660d - Mouclass.SYS
fe4c0000 2e4066e2 - VIDEOPRT.SYS
fe4d0000 2e4066e8 - vga.sys
fe4f0000 2e414f30 - Ndis.SYS
fe500000 2e40719b - eLnkii.sys
fe530000 2e47c740 - nbfs.sys
fe570000 2e83a89e - nlnkmb.sys
fe5a0000 2e5256b8 - afd.sys
fe5d0000 2e4167f7 - netbios.sys
fe5f0000 2e4f9f51 - rdr.sys
fe600000 2e4166e2 - nlnkspk.sys

Address dump Build [1057]
FF941E4c fe5105df fe5105df 00000001 f6640128 fe4a8228 000002fe - Name
ff941e60 fe501360 fe501360 00000246 00004a02 00000000 00000000 - NDIS.SYS
ff941eb4 fe481509 fe481509 f6688c83 f6682888 00000000 f668138 - eLnkii.sys
ff941ee0 fe481ea8 fe481ea8 fe482078 00000000 ff541f04 8013c58a - i8042prt.SYS
ff941ef0 fe482078 fe482078 00000000 ff941f04 8013c58a f6688c8 - i8042prt.sys
ff941ef0 8013c58a 8013c58a f6688c83 f6680940 80405900 00000031 - ntoskrnl.exe
ff941efc 80405900 80405900 00000031 06060606 06060606 06060606 - hal.dll

Restart and set the recovery options in the system control panel
or the /CRASHDEBUG system start option if this message reappears,
contact your system administrator or technical support group.
  
```

So sieht ein typischer »Bluescreen« aus

Auf den ersten Blick sieht der Bluescreen sehr verwirrend aus, da er alle möglichen hexadezimalen Speicheradressen von zuletzt geladenen Treibern und Systemkomponenten darstellt. Man kann den BSOD grob in drei Bereiche unterteilen, die unterschiedliche Informationen beinhalten:

1. Der Fehlercode samt zugehöriger Parameter befindet sich obersten Teil des Bluescreen
2. Im Mittelteil ist eine Liste aller geladenen Treiber zu sehen.
3. Der unterste Abschnitt enthält eine Liste der Komponenten, die ich zuletzt im Stack befanden.

Sehr viele Problemursachen lassen sich mit den beiden erst genannten Informationen ermitteln. Für einige wird jedoch ein Kernel-Debugger benötigt, mit dem sich innerhalb eines Speicherdumps und einem Stack-Trace genauere Aussagen über die Absturzursache machen lassen. Da der Umgang mit dem Kernel-Debugger nicht gerade unkompliziert ist und zudem einiges an Hintergrundwissen über Betriebssysteminternia erfordert, wird sich in der Regel nur das Support-Personal von Microsoft oder einem entsprechenden Partner mit dieser Thematik beschäftigen müssen.

Um auf die Bluescreen-Analyse näher einzugehen, soll im folgenden ein konkretes Beispiel betrachtet werden. Grundlage ist der Bluescreen, wie im Bild auf Seite 1 dargestellt.

Zunächst ermittelt man aus dem Bluescreen den Stop-Code, der sich im Beispiel folgendermaßen darstellt:

```
*** STOP: 0x0000000A (0x00000000, 0x0000001a,
0x00000000, 0xfe5000df)
IRQL_NOT_LESS_OR_EQUAL
```

Der Stop-Code selbst ist die erste hexadezimale Zahl, die nach dem Schlüsselwort STOP steht, also 0x0000000A, welcher mit der Meldung IRQL_NOT_LESS_OR_EQUAL verknüpft ist und über vier Parameter verfügt, die in Klammern angegeben sind. Dieser Stop-Code wird in der Regel dann ausgegeben, wenn ein Treiber versucht auf eine ungültige Speicheradresse zuzugreifen. Die in

Klammern angegebenen Parameter geben (von links nach rechts betrachtet) Aufschluß über:

1. Die Adresse (0x00000000), die fälschlicherweise referenziert wurde,
2. den Internal Request Level (IRQL), der nötig war, um auf die Speicheradresse zuzugreifen, im Beispiel 0x0000001a,
3. die Zugriffsart, wobei 0x00000001 für Lese- und 0x00000000 für Schreibzugriff steht,
4. die Speicheradresse (0xfe5000df), von der aus versucht wurde, auf die im ersten Parameter angegebene Adresse zuzugreifen.

Aus dem letzten Parameter, der Speicheradresse 0xfe5000df kann man mit Hilfe der im zweiten Abschnitt dargestellten Liste von Speicherbereichen der verschiedenen geladenen Module erkennen, welcher Treiber den Fehler verursacht hat. Im Beispiel war der Treiber elnkii.sys der Verursacher, da er den Speicherplatz von 0xfe500000 bis 0xfe52ffff belegt und 0xfe5000df genau in diesem Bereich liegt. Wenn man nun weiß, welche Komponente der mutmaßliche Verursacher ist, kann man versuchen, durch Ersetzen des betreffenden Treibers das Problem zu beseitigen.

Unter Windows-NT gibt es sehr viele verschiedene Stop-Codes, die leider nicht alle dokumentiert sind. Zudem haben die beim Stop-Code ausgegebenen Parameter unterschiedliche Bedeutung, so daß man nicht davon ausgehen kann, daß man alle Stop-Codes interpretieren kann. Microsoft hat im Internet eine Liste der gängigsten Stop-Codes und deren Bedeutung veröffentlicht. Man findet sie in der Microsoft Knowledge Base unter www.microsoft.com/kb/articles/q103/0/59.htm, oder der Technet-CD unter der Artikel-Nummer Q103059. Zudem kann man in der Knowledge Base auch gezielt nach Einträgen zu bestimmten Stop-Codes suchen lassen. Man muß lediglich den Stop-Code eingeben und wenn die Datenbank entsprechende Einträge enthält, werden alle relevanten aufgelistet. Aufpassen muß man bei der Eingabe des Stop-Codes, denn es kommt häufig vor, daß dieser von den Verfassern der Artikel nicht vollständig ausgeschrieben wird. So wird zum Beispiel ein Stop

0x0000000A oft einfach mit 0xA abgekürzt.

Problemanalyse mit einem Kernel-Dump

Nicht immer sind auftretende Probleme so einfach zu diagnostizieren, wie im gezeigten Beispiel. Es kommt zuweilen vor, daß hartnäckige und schwer zu diagnostizierende Probleme auftauchen, bei denen eine eingehendere Analyse notwendig ist. In solchen Fällen ist es ratsam, beim Systemabsturz von NT einen »Kernel-Dump« erstellen zu lassen, den man dann entweder selbst auswerten kann, oder an den Microsoft-Support sendet. Unter einem Kernel-Dump versteht man dabei eine Datei, in der ein Abbild des gesamten Systemspeichers zum Zeitpunkt eines Stops abgelegt wird. Nützlich ist so Dump zum Beispiel auch, wenn ein Server nur sporadisch ausfällt und man nicht unnötig viel Zeit für die direkte Auswertung des Bluescreens verschwenden möchte, sondern die Maschine entweder automatisch oder manuell neu gestartet werden soll.

Der Kernel-Dump enthält alle Informationen des Bluescreens plus zusätzlich vieler weiterer, mit denen aber nur ausgebildete Spezialisten etwas anfangen können. Um im Falle eines Falles einen Dump zu erhalten, muß man zunächst einmal in der Systemsteuerung über das Icon »System« einstellen, daß ein solcher erzeugt wird. Dazu wählt man im Fenster »Systemeigenschaften« das Register »Starten/Herunterfahren« aus und kreuzt dort unter »Wiederherstellung« den Eintrag »Debug-Info festhalten in:« an. In der Zeile darunter stehen Pfad und Name der Datei, in die der Dump geschrieben werden soll. Standardmäßig ist das die Datei MEMORY.DMP, die im Windows-NT Systemverzeichnis, also zum Beispiel C:\WINNT abgelegt wird.

Wenn man nicht möchte, daß der Server im Zustand des Bluescreens stehen bleibt, kann man noch den Eintrag »Automatisch neu starten« ankreuzen. Dann fährt der Server nach dem Schreiben des Dumps automatisch wieder hoch, sofern dies technisch noch möglich ist.

Gängige Stop-Codes:

Stop-	Beschreibung	Anmerkung
0x0000000A	IRQL_NOT_LESS_OR_EQUAL	Ein Prozess hat versucht, mit einem zu hohen Internal Request Level (IRQL) auf auslagerbaren Speicher zuzugreifen. Ein Prozess darf nur auf Objekte zugreifen deren IRQL kleiner oder gleich dem eigenen ist. In der Regel wird dieses Problem durch einen Systemtreiber verursacht, der auf eine falsche Speicheradresse zugreift.
0x0000003A	MULTIPROCESSOR_CONFIGURATION_NOT_SUPPORTED	Windows-NT hat in einem System mehrere Prozessoren entdeckt, die sich jedoch nicht symmetrisch zueinander verhalten. Das bedeutet, daß die verwendeten Prozessoren nicht vom selben Typ (Hersteller, Baureihe, Taktfrequenz, etc.) sind. So wäre zum Beispiel nicht möglich, auf einem Multiprozessor-Board ein Intel- und einen AMD-Prozessor gemeinsam unter NT zu betreiben.
0x0000001E	KMODE_EXCEPTION_NOT_HANDLED	Auch dieser Stop-Code wird häufig von fehlerhaften Treibern verursacht. Er kann aber auch bei Unverträglichkeiten mit bestimmten BIOS-Versionen auftreten. Eine weitere mögliche Ursache ist das Fehlen von Speicherplatz bei einer Installation von Windows-NT.

Stop-	Beschreibung	Anmerkung
0x00000073	CONFIG_LIST_FAILED	Dieser Stop-Code wird ausgegeben, wenn NT bei Starten Probleme hat, auf einen der Registry-Hives SAM, SOFTWARE oder SECURITY zuzugreifen. Man kann versuchen, diese Dateien über die Notfalldiskette wieder herzustellen.
0x0000007B	INACCESSIBLE_BOOT_DEVICE	Dieser Stop-Code weist auf ein Problem beim Booten von Windows-NT hin. Entweder besteht ein Problem mit dem Platten-Controller, oder es liegt an einem fehlenden bzw. korrupten Treiber für SCSI- oder (E)IDE-Laufwerke. Auch bei einem mit einem Bootsektorvirus verseuchten Rechner kann diese Meldung erscheinen.
0x0000008B	MBR_CHECKSUM_MISMATCH	Dieser Stop-Code weist auf einen Boot-sektorvirus hin und tritt beim Booten von Windows-NT auf.
0x0000002E	DATA_BUS_ERROR	Dieser Fehler wird häufig durch einen Parity-Error im System-Speicher hervorgerufen. Möglicherweise befinden sich beschädigte oder nicht korrekt eingebaute Speichermodule im System, die ausgetauscht oder richtig eingesetzt werden müssen.

Damit ein Dump erzeugt werden kann, müssen noch verschiedene Voraussetzungen erfüllt sein:

1. Es muß eine Auslagerungsdatei auf dem Laufwerk vorhanden sein, auf dem sich das Windows-NT-System-Verzeichnis befindet.
2. Die Auslagerungsdatei muß mindestens ein Megabyte größer sein, als der komplette im System vorhandene physikalische Speicher. Das ist notwendig, da bei einem Absturz der gesamte Inhalt des physikalischen Speichers in der Auslagerungsdatei zwischen gepuffert wird.
3. Auf dem Systemlaufwerk muß soviel freier Speicher vorhanden sein, daß der Inhalt der Auslagerungsdatei darin Platz findet. Das hat damit zu tun, daß beim Systemneustart der Dump aus der Auslagerungsdatei in eine Datei (MEMORY.DMP) kopiert wird, die dann zur weiteren Verarbeitung zur Verfügung steht.

Wenn nun so eine Dump vorliegt, kann man ihn wie gesagt an Microsoft zur Analyse schicken oder aber selbst daran gehen, die enthaltenen Informationen auszuwerten. Da die Datei zunächst in einem nicht lesbaren Binärformat vorliegt, benötigt man ein Werkzeug, mit dem man sich Zugang zu den relevanten Daten verschaffen kann. Microsoft liefert ein solches Werkzeug standardmäßig mit aus. Es hat

den Namen DUMPEXAM.EXE und befindet sich für die Intel-Version im Verzeichnis SUPPORT\DEBUG\I386 der Windows-NT-CD. Neben DUMPEXAM werden noch die beiden Dateien IMAGEHLP.DLL und KDEXTX86.EXE benötigt, die sich im selben Verzeichnis der CD befinden. Durch den Aufruf von DUMPEXAM, wird eine Textdatei mit dem Namen MEMORY.TXT erzeugt, die in der Regel zum einen sehr viel kleiner ist, als der Original-Dump und zum anderen alle Informationen beinhaltet, die schon aus dem Bluescreen bekannt sind. Um möglichst viele Informationen zu erhalten, sollte man beim Aufruf von DUMPEXAM eine Pfadanangabe für die NT-Debug-Symboldateien mit angeben. Die Symboldateien für die Intel-Version befinden sich auf der Windows-NT-Installations-CD im Pfad SUPPORT/DEBUG /I386/ SYMBOLS. Allerdings liegen sie dort, zumindest bei NT 4.0, nur in gepackter Form vor und müssen vor Verwendung entpackt werden. Dazu gibt es ebenfalls auf der CD ein Skript mit dem Namen EXPNDSYM.CMD, welches alle notwendigen Dateien kopiert und entpackt. Wenn auf dem NT-Rechner Service Packs installiert, müssen die entsprechenden Symboldateien ebenfalls im Pfad vorhanden, beziehungsweise alte Dateien durch die neuen ersetzt worden sein.

Beispiel:

Der Kernel-Dump befindet sich im Verzeichnis C:\WINNT und hat den Namen MEMORY.DMP, die Symboldateien im Verzeichnis C:\WINNT\SYSTEM32\SYMBOLS.

Um nun eine ASCII-Datei mit verwertbaren Informationen zu erhalten, gibt man folgendes ein:

```
DUMPEXAM y C:\WINNT\SYSTEM32\SYMBOLS
C:\WINNT\MEMORY.DMP.
```

Nach der Ausführung befindet sich im Verzeichnis C:\WINNT die Datei MEMORY.TXT, die zur weiteren Auswertung verwendet werden kann.

Ausschnitt aus der Datei MEMORY.TXT, die den Stop-Code samt Parametern zeigt

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
**
** Windows NT Crash Dump Analysis
**
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*
Filename . . . . . C:\WINNTUS\MEMORY.DMP
Signature . . . . . PAGE
ValidDump . . . . . DUMP
MajorVersion . . . . . free system
MinorVersion . . . . . 1381
DirectoryTableBase . .0x00030000
PfnDataBase . . . . . 0x827c3000
PsLoadedModuleList . .0x8014eab0
PsActiveProcessHead . .0x8014e9a8
MachineImageType . . .i386
NumberProcessors . . . 1
BugCheckCode . . . . . 0x0000001e
BugCheckParameter1 . .0xc0000005
BugCheckParameter2 . .0x00000000
BugCheckParameter3 . .0x00000000
BugCheckParameter4 . .0x00000000
ExceptionCode . . . . . 0x80000003
ExceptionFlags . . . . 0x00000001
ExceptionAddress . . . 0x80119276
  
```

In der Abbildung auf Seite 3/4 ist ein Beispiel für einen Stop-Code samt zugehörigen Parametern zu sehen, der aus einem Kernel-Dump mit Hilfe von DUMPEXAM ermittelt wurde. Es handelt sich hier um einen STOP 0x0000001E (0xc0000005, 0x00000000, 0x00000000, 0x00000000) KMODE_EXCEPTION_NOT_HANDLED.

Fazit

Man kann mit Hilfe der von Microsoft zur Verfügung gestellten Werkzeuge, wie DUMPEXAM und den Knowledgebase-Artikeln im Internet sehr viele Probleme selbst beheben. Häufig sind fehlerhafte oder beschädigte Treiber die Ursache. Mit dem Wissen, um welchen Treiber es sich handelt, kann man bereits selbst im Eigenversuch probieren, durch den Einsatz von neueren oder gegebenenfalls auch älteren Versionen das Problem in den Griff zu bekommen und sich die teilweise kostspielige Unterstützung durch die einschlägigen Support-Abteilungen sparen. Neue Treiberversionen sind im Internet zuhauf auf den Servern der jeweiligen Hardwareanbieter oder aber auf dem Microsoft-FTP-Server unter ftp.microsoft.com zu finden.

Zur Person

DIPL. ING. DIRK PELZER arbeitet als freier Consultant und Journalist in München. Er betreibt ein Storage Labor für verschiedene namhafte Fachzeitschriften. Zudem beschäftigt er sich mit Speichernetzen und Hochverfügbarkeit.