Wenn der GAU kommt

Datensicherungsstrategie erarbeiten und umsetzen

Wenn alle Unternehmensrechner ihre Daten verlieren, kann nur ein kleiner, unbedeutender Systemadministrator mit einem Schrank voll Sicherungsbänder das Schicksal wenden. Wehe ihm, wenn dann nicht alle Werkzeuge bereitliegen. Dieser Artikel zeigt, welche Maßnahmen er ergreift und welche Fallen er umgeht, um eine gute Datensicherungsstrategie zu besitzen.

von **Stefan Schumacher**

Sicher ist,
dass nichts sicher ist.
Selbst das nicht.

Joachim Ringelnatz

Komplette Kinofilme auf einem Handy herumzutragen, ist heute kein Problem mehr. Mit den steigenden Datenmengen wachsen die Anforderungen an Datensicherungsmethoden. Die Datenmengen selbst sind nicht das Problem – umständlich ist nur, in so genannten organisch gewachsenen Netzen die Daten zu bestimmen, die zu sichern sind. Sobald sie in den Untiefen des Sicherungssystems verschwinden, sollten sie dann auch wieder herauszubekommen sein. Eine alte Administratorenweisheit besagt: Niemand will Backup. Aber alle wollen Restore.

Dieser Artikel gibt Tipps für eine Datensicherungsstrategie und ihre Umsetzung. Es geht nicht um die Technik, sondern ums Verwalten (lateinisch: administrare). Einige dieser Tipps hat der Autor mit Blut, Schweiß und Tränen erkauft.

Disaster Recovery Planning

Ein Notfallplan, neudeutsch *Disaster Recovery Plan*, dient zur Betriebswiederherstellung im Notfall. Er ist das Fundament der Sicherungsstrategie. Für die Organisation einer Sicherungsstrategie ist eine Anforderungsanalyse vorzunehmen und die gewonnenen Erkenntnisse umzusetzen. Schritte dahin sind:

- Inventur machen
- Bedrohungsszenarien analysieren
- Wichtige Daten identifizieren
- Systeme sichern
- Dokumentieren
- Testen

Bei der **Inventur** katalogisiert der Admin, welche Hardware, Betriebssysteme, Anwendungsprogramme und Dienste auf seinen Maschinen laufen. Das ermöglicht, Anforderungen an die Sicherungssoftware festzulegen. Existiert schon ein Inventar, zum Beispiel in einer Datenbank oder einem speziellen Inventurprogramm, ist dieses um die Sicherungsinformationen zu erweitern.

Bei den **Bedrohungsszenarien** geht es um die Fragen, wer oder was die Daten bedroht, und wie der Schutz davor aussehen kann. In der Regel gibt es bestimmte Fehlertypen:

Benutzerfehler: Der Benutzer löscht oder verfälscht Dateien. Hier muss die letzte vorhandene Version zurückgespielt werden. Dieser häufige Fehler erfordert eine Archivierung der Daten oder Sicherungsbänder.

Administratorenfehler: Wenn er vorkommt, hat er es in sich. Hierfür sollte das gesamte System zum Beispiel mit Snapshots gesichert werden.

Systemfehler: Bei Festplattendefekt schützen Sicherungen auf Wechselmedien, anderen Rechnern oder RAID-Systemen. Gegen Dateisystemkorruption helfen nur archivierte Sicherungen, da sich ein Fehler auf die Spiegel fortpflanzt.

Einbruch: Bei elektronischem Einbruch ist die Frage zu stellen, ob die Daten noch integer sind, sofern sie nicht gelöscht wurden. Datenvernichtung erfordert die Rücksicherung. Aber im Falle manipulierter Daten sind Archive notwendig. Gegen herkömmlichen Einruch, also Vandalismus oder Diebstahl, hilft die letzte Datensicherung nur, wenn sie der Einbrecher nicht finden konnte.

Naturkatastrophen: Bis zum Sommer 2002 haben einige Unternehmen nicht im Traum daran gedacht, das die Elbe mal in ihren Kellern vorbeischaut. In Deutschland sind also Fluten an manchen Standorten eine Gefahr. Weiterhin sind Flugzeugabstürze, Erdrutsche, Großbrände oder Unglücke in benachbarten Unternehmungen (Chemiewerk, E-Werk, Raffinierie) möglich. Was passiert also, wenn unser Gebäude oder Stadtteil ausradiert wird, unser Unternehmen aber die Daten noch benötigt? Existieren Sicherungen an anderen Orten? Sicherungsbänder sollten entsprechend dem Gefahrenpotential ausgelagert werden, zum Beispiel in andere Filialen. Ebenso möglich ist eine Spiegelung oder Replikation der Datenbestände auf entfernte Systeme.

Wie **identifiziert** man nun die wirklich wichtigen Daten – also die, deren Verlust nicht akzeptabel ist? Drei Kriterien helfen:

1. Die Daten sind bei Verlust nicht wiederherstellbar, da sie an einen bestimmten Zeitpunkt gebunden sind – wie Patientendaten, Logdateien, Versuchsprotokolle – oder an bestimmte Programme und Bearbeiter.
2. Die Daten sind zwar rekonstruierbar, aber der Aufwand dafür übersteigt den Aufwand der Sicherung. Oder die Nichtverfügbarkeit der Daten verursachte gleich einen wirtschaftlichen Totalschaden.
3. Es handelt sich um Betriebssysteme, Anwendungen oder Dienste. Kann oder möchte man diese nicht sichern, dann zumindest deren Konfigurationsdateien.

Herzstück: Datensicherung

Zur **Datensicherung** gehören neben den Home-Verzeichnissen der Benutzer im Allgemeinen die Datenbestände aus Anwendungen, etwa Datenbanken oder CAD-Systeme. Auch Konfigurationsdateien oder modifizierte Programme sollten nach jeder Veränderung gesichert werden. Systematische Dateiablage vereinfacht die Sicherung. Das ist ein Knackpunkt, wenn Heim-Betriebssysteme eingesetzt werden, die es dem Benutzer erlauben, seine Dateien wahllos im System zu verstreuen – es sei denn, man kopiert sowieso die komplette Platte. Verzeichnisstrukturen vereinfachen auch die Sicherung von Daten, die nicht oft verändert werden, etwa Bilder, Fotos, Audiodateien oder Filme.

Einige Sicherungssysteme verwenden einen Index, insbesondere solche für Netzwerksysteme wie *Amanda* oder *Bacula*. Amanda legt Logdateien an, Bacula verwendet eine Datenbank. In diesen Index wandern alle Metadaten zur Sicherung nach dem Schema: Wann wurde welche Datei von welchem Client in welcher Version auf welches Band gesichert. Dieser Index ist also wichtig, wenn man die Rücksicherung einer bestimmten Version durchführen muss. Ohne den Index kann man zwar in der Regel noch die Bänder von Hand zurückspielen, dies ist aber bei einer größen Anzahl mühselig. Daher sollte zwingend auch das Sicherungssystem selbst gegen Ausfälle und Verlust gesichert werden. Verwendet man eigene Skripte, die einen Katalog der gesicherten Dateien erstellen, sollten diese ebenfalls gesichert werden.

In Netzwerken, in denen nur Teile der Clients gesichert werden, ist es wichtig, die Benutzer der Systeme über die Strategie zu unterrichten. Denn die Anwender sollen ihre Daten nur in Verzeichnissen speichern, die auch gesichert werden. Dürfen die Benutzer auf den Clients selbständig Programme installieren, muss bereits im Vorfeld dafür gesorgt werden, dass deren Anwendungsdaten ebenfalls mitgesichert werden. Homogenität vereinfacht natürlich die Datensicherung. Meistens ist es aber nicht möglich, alle Systeme homogen aufzusetzen. Es ist hingegen möglich, die Systemdienste anzugleichen, Benutzerdaten zentral zu verwalten und gleiche Hardware einzusetzen.

Ebenso erleichtert Systematisierung im Tagesgeschäft die Sicherung. Viele erledigen bestimmte Handgriffe manuell, etwa neue Benutzer anlegen oder den Datenbankserver warten. Sie beherrschen die Befehle aus dem Effeff. Weitsichtiger ist es aber, diese Befehle in ein Shellskript zu gießen. Das hat den Vorteil, dass die Skripte mitgesichert werden, also nach einem Systemausfall wiederhergestellt werden können. Außerdem kann man ein Skript leicht kommentieren und somit dokumentieren. Im Zweifelsfall wissen also auch die Kollegen, was auf den Maschinen abläuft.

A propos Skript: Wer einzelne Systeme nachts per Cronjob erledigt, sollte seine Skripte robust halten, indem er zum Beispiel Toleranzen einplant. Kommt es nämlich dazu, das sich die Laufzeit der Skripte verschiebt, kann unter Umständen der gesamte Sicherungsalgorithmus fehlschlagen, wie das Beispiel im Kasten *Cronjob trifft auf Batchjob* zeigt.

Praxisbeispiel: Cronjob trifft auf Batchjob

Ein Unternehmen wollte vier kleine Arbeitsgruppenserver mit verschiedenen Unix-Varianten sichern. Dazu stand ein älterer Rechner mit Spoolingplatte und Streamer zur Verfügung. Nächtlich sollten die vier Server ihre Daten per `dump` und SSH-Tunnel auf den Spoolserver schieben, der diese danach auf ein Streamerband schreibt. Da alle Rechner im selben Raum standen, verkabelte sie ein einfacher Switch extra für die Sicherung mit TBase100. Ein Cronjob auf den vier Servern startete mit einer Stunde Abstand den Dump-Lauf und auf dem Spoolingserver die endgültige Sicherung auf Band.

Da das Backup-Netzwerk nur für das Backup genutzt wurde, bemerkte niemand, dass ein Port am Switch massive Hardwareprobleme entwickelte und daher die Sicherung über das Netzwerk extrem langsam wurde. Als der Spoolingserver per Cron begann, die Daten auf Band zu schreiben, lieferte der dritte Server seine Sicherung immer noch an die Spoolingplatte aus. Der Spoolingserver verwendete zwar `dump`, um das Band zu schreiben, sodass die anderen Sicherungen fehlerfrei waren – das Archiv vom dritten Server war aber defekt. Das Ganze flog erst auf, als ein Admin früher als gewöhnlich in der Firma auftauchte und bemerkte, dass der Sicherungsprozess noch läuft.

Nachdem der Netzwerkfehler behoben war, wurde das Skript abgeändert. Und zwar so, dass nächtlich per Cron auf dem Spoolingserver ein Shellskript anlief, das nacheinander die Sicherungsläufe auf den Servern anwarf und erst nach erfolgreichem Abschluss dieser Sicherungen die Archive auf Band schrieb. Abschließend wurden die Logdateien aller Dump-Läufe per Mail an die Administratoren weitergeleitet – und von diesen auch gelesen.

Der Admin ist der wichtigste Faktor

Die meisten Sicherungsprogramme erstellen Logdateien. Dies sollte man aktivieren – und die erzeugten Protokolle unbedingt lesen. Denn nur so ist sichergestellt, dass alle Sicherungsläufe auch wie erwartet funktionieren. Wann immer Daten übertragen oder bearbeitet werden, kann dabei etwas schief gehen. Daher sollte jedes Programm, das Daten verifizieren kann, dies auch tun. Bei den Kompressionsprogrammen Bzip2 und Gzip zum Beispiel dient dazu die Option `-t`. Ansonsten erledigen das Prüfsummen wie MD5 oder SHA1. Komplette Dateisysteme lassen sich mit *mtree(8)* überprüfen und vergleichen.

Wichtig ist, die Konfiguration aller Rechner und deren Software zu kennen, und zwar nicht nur als Datei, sondern auch als Ausdruck. Als Letztes benötigt man eine Live-CD oder eine separate Festplatte. Die Live-CD lässt sich mit `pkgsrcsysutilsmklivecd` selbst erstellen. Zu denken ist an Unterstützung für alle benötigten Geräte (Laufwerke, Netzwerkkarten, CGD, RAID, LVM) und Pakete. Alternativ kann man auch eine kleinere Festplatte mit NetBSD einrichten und alle benötigten Programme einbinden.

Zum Sicherungssystem gehört auch der Admin. Es darf nicht nur eine einzelne Person wissen, wie das Sicherungssystem funktioniert. Schließlich ist die auch auch mal im Urlaub. Oder im Krankenhaus. Es muss sichergestellt sein, dass auch andere Administratoren als der implementierende die Systemsicherung beherrschen. Daher ist es lebenswichtig, eine funktionierende Dokumentation des gesamten Systems zu erstellen. Idealerweise sollte hier der gesamte Entwurf und die gesamte Implementierung des Systems formal dargelegt werden – zusätzlich aber auch eine einfache und leicht verständliche Schritt-für-Schritt-Anleitung zur Rücksicherung von Daten vorhanden sein.

Nützlich ist es, die Einweisung in das Sicherungssystem mit der Dokumentation als Testlauf anzusetzen. Hierbei sollen die beteiligten Admins anhand der Dokumentation eine Rücksicherung auf ein Testsystem durchführen. Das stellt die Qualität sowohl der Dokumentation als auch des Sicherungssystems bequem auf die Probe: Versteht jeder, was beschrieben wurde? Fehlen Teile? Gelingt die Rücksicherung? Nach Abschluss der Rücksicherung dient eine gemeinsame Manöverkritik dazu, Schwächen und Fehler des Systems und der Dokumentation durchzuprechen und abzustellen.

Nicht nur der Administrator sollte in das Sicherungssystem eingewiesen werden. Insbesondere in kleinen bis mittleren Unternehmungen ist das ein Problem, weil die oft keine EDV-Abteilung besitzen, sondern nur einen Teilzeit-Admin beschäftigen. Ist der weg, weiß keiner bescheid. Lösungen lassen sich über einen externen Berater finden, der im System eingewiesen und vertraglich entsprechend in den Ablauf eingebunden wird.

Damit sich die ganze Mühe lohnt: Eine Inventur der Medienbestände ist regelmäßig angebracht, bei der neben der Vollständigkeit auch die Funktionsfähigkeit getestet wird. Bänder sollten mindestens einmal pro Jahr einmal durchgespult werden.

Testen, testen, testen

Ein Sicherungssystem, das nicht unter realen Bedingungen getestet wurde, kann niemand als funktionierend bezeichnen. Das hat mehrere Gründe. Zunächst müssen die Administratoren tatsächlich in der Lage sein, die Rücksicherung durchzuführen. In der Regel erzeugt ein Systemausfall mit Datenverlust Stress, denn der Chef will das Problem am besten gestern gelöst sehen. Daher ist es unerlässlich, derartige Situationen und Verfahren in Ruhe durchzuspielen, um Sicherheit im Ablauf zu bekommen. Das verringert den Stress und somit Fehlerquellen.

Außerdem ist ohne Tests nicht sichergestellt, dass die Sicherungsstrategie auch wirklich funktioniert. Man kann zwar versuchen, im Entwurf der Strategie Fehlerquellen zu minimieren, kann diese aber nicht ausschließen, wie in jedem Entwurfsprozess. Nur Testläufe stellen sicher, das die Prozedur funktioniert und auch wirklich die gewünschten Daten sichert. Folgende Testläufe spielen die gängigsten Situationen durch:

- Rücksicherung einzelner Dateien
- Sichern von Versionen einzelner Dateien
- Rückspielen eines Client-Dateisystems
- Neuaufsetzen eines Komplettsystems, also Betriebssystem mit Konfiguration, Anwendungssoftware und Daten
- Rücksicherung einer Archivversion zu einem bestimmten Zeitpunkt in der Vergangenheit (*Point-in-Time-Recovery*)
- Rücksicherung eines Systems, obwohl der Backup-Server ausgefallen ist

Woran man alles denken muss...

Aus Datenschutz-, rechtlichen oder technischen Gründen kann es notwenig sein, das Netzwerk in verschiedene Sicherungskreise zu unterteilen. So sollten Forschungsdaten anders behandelt werden, als das Datenverzeichnis des Webservers. Erstere sollten verschlüsselt im Tresor liegen, zweitere sind diesen Aufwand in der Regel nicht wert. Und so, wie der Gesetzgeber für einige Daten eine Mindestarchivierungszeit vorschreibt, legt er auch Obergrenzen fest. Beispielsweise sind Eintragungen in der Personalakte über Disziplinarmaßnahmen nach Eintritt des Verwertungsverbots meistens nach zwei Jahren von Amts wegen zu entfernen. Das heißt, dass binnen dieser Frist angelegte Sicherungskopien der Akte zu vernichten sind. Daher ist es sinnvoll, die Personalabteilung in einen eigenen Sicherungskreis zu gliedern und auch entsprechend zu archivieren, da man sonst juristisch belangt werden kann.

Unternehmen sichern oft nur an Werktagen. Wenn doch einmal jemand an einem Feiertag oder am Wochenende arbeitet, dann will es Murphy's Law, dass es dann garantiert zu einem Datenverlust kommt. Sind die entsprechenden Rechner also außerhalb der Werkszeiten an, können sie über das Netzwerk gesichert werden. Falls nicht, sollte die Netzwerkrichtlinien und Benutzerordnungen auf die fehlende Sicherung hinweisen. Bestimmte Maßnahmen ermöglichen auch eine Sicherung der Daten am Wochenende, zum Beispiel ein Fileserver, auf den die Mitarbeiter Kopien der zu sichernden Dateien ablegen können.

Tipps zum Komprimieren und Verschlüsseln

Wer Archive komprimiert, um Platz zu sparen, sollte bedenken, dass Archivfehler die Dekomprimierung oft unmöglich machen. Daher sollte man nach entsprechender Kosten-Nutzen-Abwägung die gesicherten Archive verifizieren. Ebenso ist es sinnvoll, nur gängige Komprimierungsprogramme zu verwenden, um auch später wieder an die Daten heranzukommen. Aktuelle Bandlaufwerke unterstützen auch Hardwarekomprimierung. Das heißt, dass das Laufwerk vor dem eigentlichen Schreiben der Daten diese in einem eigenen Chip komprimiert. Dieses Verfahren hat den Vorteil, die sichernde Maschine nicht weiter zu belasten.

Ob man Komprimierungsverfahren einsetzt, hängt von der entstehenden Last (Netzwerk, CPU, Bandbedarf) und Zusammensetzung der Daten ab. Besteht die Mehrheit aus natürlichsprachigen Texten wie Logdateien, Textdokumenten oder Text-Datenbanken, ist eine recht große Kompressionsrate gegeben. Bereits komprimierte Datentypen wie JPEG, MP3, AVI oder auch verschlüsselte Daten hingegen lassen sich nicht weiter komprimieren. Zur Komprimierung eignen sich die Standardprogramme *Gzip* und *Bzip2*, da sie weit verbreitet sind und bisher zuverlässig arbeiten. Bzip2 erzeugt zwar im Allgemeinen eine bessere Komprimierungsrate, erzeugt dabei aber auch höhere Systemlast und Laufzeiten.

Es ist sinnlos, ein verschlüsselndes Dateisystem einzusetzen und unverschlüsselte Backups im Schrank liegen zu haben. Daher sollten auch Backups verschlüsselt werden. Dies gilt insbesondere dann, wenn die Sicherungsbänder extern gelagert werden. Verschlüsseln kann man Archive sicher mit *OpenSSL*, *GnuPG* oder dem symmetrischen Verschlüsselungprogramm *mcrypt*, das verschiedene Algorithmen wie *Rijndael*, *3DES* oder

Panama unterstützt. Im Fall von GnuPG empfiehlt es sich, symmetrische Verschlüsselung einzusetzen. Denn für die Entschlüsselung benötigt man bei der asymmetrischen Verschlüsselung sonst wieder den GnuPG-Schlüssel – was das Archiv unbrauchbar macht, wenn der nicht mehr existiert.

Werden die Archive nicht mittels einer Pipe direkt verschlüsselt auf Datenträger geschrieben, müssen die unverschlüsselten Dateien sicher gelöscht (gewipet) werden. Dazu gibt es zwar die Option `-P` des *rm*-Kommandos, die Dateien vor dem Löschen dreimal überschreibt. Diese Option genügt aber nicht den allgemein anerkannten Regeln der Technik. Daher sollte dringend ein Programm verwendet werden, das zumindest den US-Sicherheitsstandard *5220.22-M(ECE)* oder Peter Gutmanns Standard implementiert. Im Verzeichnis *pkgsrc* befinden sich dazu die Tools `pkgsrcsecuritydestroy` oder `pkgsrc/sysutils/wipe`. Empfehlenswert ist im Fall von NetBSD, die unverschlüsselten Archive in eine mit CGD verschlüsselte Partition zu schreiben und danach an der Quelle zu wipen. Hierzu ist auf einzelnen Rechnern eine hinreichend große, mit CGD verschlüsselte *tmp*-Partition nützlich.

Folgendes Listing erzeugt einen Dump, der sofort per Bzip2 komprimiert wird. Anschließend wird der Dump mit GnuPG symmetrisch verschlüsselt und mit Destroy sicher gelöscht:

```
01 # dump -0a -f - /etc/ | \
            bzip2 > dump.bz2
02 # gpg -a -c dump.bz2 && \
            destroy -f -s 7 t dump.bz2
```

Das nächste Beispiel erzeugt einen Dump, der sofort per Pipe an Bzip2 zum Komprimieren und an OpenSSL zum Verschlüsseln geschickt wird:

```
01 # dump -0a -f - /etc/ | bzip2 | \
       openssl aes-256-ecb -out \
       dump.bz2.enc -e -salt
02 # openssl aes-256-ecb -in \
        dump.bz2.enc -d -salt | \
        bunzip2 - | restore -r -f -
```

Die zweite Zeile ist des Gegenstück zur ersten, denn hier wird der verschlüsselte und komprimierte Dump entschlüsselt, dekomprimiert und an Restore geschickt.

Eine Frage der Medien

Als Medium für die Datensicherung eignet sich prinzipiell alles, was am Markt erhältlich ist, jedoch sollte man einige Vorüberlegungen treffen.

Verlässlichkeit: Die Medien müssen manchmal jahrelange Archivierungsperioden überstehen können. Außerdem müssen Lesegeräte noch verfügbar sein. es bietet sich an, zum Beispiel zusammen mit der Inventur regelmäßig die ältesten Medien auf Lesbarkeit zu überprüfen, und gegebenenfalls das Medium zu wechseln. Alle acht bis zehn Jahre ist eine komplette Migration der Medien nötig, da sie und ihre Lesegeräte veralten.

Geschwindigkeit: Langsam sollte das Sicherungssystem nicht sein. Doch die Kapazitäten des Netzwerks und des Sicherungsservers sind zu beachten, denn ein 50-MBit-Streamer verliert in einem 10-MBit-Netz seinen Vorteil. Auch senkt ein Bandwechsel den Durchsatz des Systems, wenn sich Sicherungslauf auf mehrere Bänder erstreckt.

Dauer der Rücksicherung oder *Time to Data*: Die Zugriffszeit der Medien spielt genauso eine Rolle wie die Organisation der Archivierung selbst. Mehr als 90 Prozent aller Rücksicherungen betreffen nur einzelne Dateien, die in älteren Versionen restauriert werden müssen. Der gesamte Zeitaufwand der Operation besteht darin, die richtigen Medien mit der gewünschten Version zu finden, einzulegen und die Dateien zurückzuspielen. Verwendet man hierzu ein automatisiertes System (Index, Bandwechsler), geht das schneller, als wenn man die Daten von Hand zurückspielt. Ist man auf besonders schnelle Zugriffszeiten angewiesen, verwendet man hierarchische Speichersysteme, die verschiedene Medientypen mit unterschiedlichen Zugriffzeiten einsetzen. Häufig benötigte Dateien landen dann auf schnellen Medien wie Festplatten oder magneto-optischen Medien, während selten benötigte Daten auf entsprechend langsameren, aber größeren Medien wie Magnetbändern ruhen. Diese Methode war früher, als Festplatten noch in Megabyte vermessen wurden, sehr beliebt, und datenintensive Umgebungen wie Grafik- oder Videobearbeitung setzen sie noch heute ein.

Kapazität: Die anfallenden Datenmengen müssen auf möglichst wenige Medien passen. Verwendet man einen einfachen Streamer, ist besonders wichtig, alle Daten auf einem einzigen Band zu haben – denn niemand möchte nachts um drei neben dem Streamer Wache schieben. Beachten sollte man auch, dass steigende Kapazität die Zeit zur Rücksicherung anhebt, denn ein 10-GByte-Band lässt sich schneller durchspulen als ein 100-GByte-Band. Aus betriebswirtschaftlicher Sicht ist es zwar unnötig, einen 100-GByte-Streamer mit entsprechend großen und teuren Medien anzuschaffen, wenn täglich nur 5 GByte Daten anfallen. Andererseits sollte man auch nicht zu kurz planen, da die Hardware wenigstens die vier Jahre bis zur Abschreibung im Einsatz bleiben sollte.

Kosten: Das System darf üblicherweise nichts kosten. Wozu braucht man schon ein Backup-System, wenn doch alles prima läuft? ZIP-Medien, CD/DVD und Festplatten sind weit verbreitet, billig und von daher gut zur Rücksicherung geeignet. Nachteilig wirkt sich allerdings deren relativ geringe Integrität und kurze Lebensdauer aus. Kostspielig und langsam, aber dennoch überzeugend durch hohe Kapazitäten, exzellenter Integrität und langer Lebensdauer sind Magnetbänder. Es bietet sich an, verschiedene Medien zu mischen, zum Beispiel nur USB-Sticks oder DVD-RW für tägliche Sicherungen zu verwenden. Festplatten, die Daten sichern, sollten als Wechselplatte ausgeführt sein und offline gelagert werden.

Richtig lagern

Bänder müssen aufrecht stehend gelagert werden, da sich sonst das Magnetband lockern kann – und niemand will Bandsalat bei einer Rücksicherung erleben. Physikalische Sicherung gegen Einbruch und Brandschutzeinrichtungen sind ebenso obligatorisch wie eine restriktive Zugangspolitik.

Wer richtig große Datenmengen zu schultern hat, für den existieren automatische Bandwechseleinheiten, die die Medien und Hüllen mit einem Strichcode etikettieren. Für alle anderen gilt: Da die Anzahl der Medien mit der Zeit wächst und die Archivierung komplexer wird, bietet sich ein datenbankengestützes Inventarisierungssystem an. Die Datenbank erfasst alle Eigenschaften, die helfen, das passende Sicherungsband zu finden. Folgende Eigenschaften sollten darin stehen:

1. Primärschlüssel
2. Name
3. Zweck
4. Medientyp
5. Lagerort
6. Gesicherte Dateien inklusive Datum, Version, Eigentümer, Attribute

Der Primärschlüssel dient der Identifikation des Mediums. Das kann eine laufende Nummer sein, sinnvoller ist aber, in ihm Informationen zu kodieren, etwa Datum, gesicherte Daten, IP-Adresse oder Level. Der Primärschlüssel steht auf jedem Medium und der entsprechenden Hülle. Die Eigenschaft *Name* ist für jene da, die die Medien nicht über den Primärschlüssel ansprechen möchten. Zweck, Medientyp und der Lagerort der Medien gehören ebenfalls in den Katalog. Das ist notwendig, um im Falle einer Rücksicherung das passende Medium zu finden.

Sicherungsvarianten

Die verschiedenen Arten, Daten zu sichern, sind zum Abschluss hier aufgeführt. Der anschließende Kasten enthält ein Beispiel für eine Sicherungsstrategie. Und für die Spielernaturen zeigt der letzte Kasten, wie man mit Dump-Levels *Türme von Hanoi* spielt. Da bekommt der Begriff **Diskjockey** doch gleich eine neue Bedeutung!

Ein **Komplettbackup** sichert alle Daten. Das ist einfach, ebenso wie das Zurückspielen der Daten. Nachteile sind die Dauer der Sicherung und der benötigte Platz, da auch Daten gesichert werden, die sich seit dem letzten Backup nicht verändert haben. Ein Komplettbackup wird normalerweise montags oder freitags erstellt.

Anders macht das das **differentielle Backup**. Es sichert nur Daten, die seit dem letzten Komplettbackup geändert wurden. Bei der Rücksicherung muss allerdings die Reihenfolge der Bänder beachtet werden. Im allgemeinen benötigt man hierzu die letzte Komplettsicherung und das letzte Tagesband.

Das **inkrementelle Backup** sichert nur die Dateien, die sich seit dem letzten Inkrementalbackup geändert haben. Bei der Rücksicherung muss ebenfalls streng die Reihenfolge beachtet werden, dafür ist bei der Sicherung noch weniger Platz und Zeit als beim Differentialbackup erforderlich.

Sicherungsprogramme unterstützen normalerweise den Einsatz so genannter **Dump-Level**, die mittels einer Zahl die zu sichernden Dateien angeben. Auf dem Level n werden also alle Dateien gesichert, die seit der Sicherung n - *1* verändert wurden. Vorteil ist wieder die Zeit- und Platzersparnis, Nachteil ist die aufwändigere Rücksicherung.

Man kann aber auch die Methoden mischen. So ist es beispielsweise praktikabel, montags eine Komplettsicherung mit Level 0 durchzuführen, und werktags mit Level 1 die Arbeitsdaten der Woche zu sichern. Da es auch am Wochenende zu Arbeitseinsätzen kommen kann, könnten am Sonnabend und Sonntag mit Level 2 die Tagesdaten gesichert werden.

Beispiel für eine Datensicherungsstrategie

Das Beispiel betrachtet einen einfachen Rechner mit Leveln.

- Am Montag existiert 1 GByte an Daten.
- Jeden Tag kommen 100 MByte hinzu.
- Am Sonntag liegen 1,6 GByte Daten vor.
- Sicherungen erfolgen in der Nacht, vor Arbeitsbeginn.
- Sonntagmittag sei eine Rücksicherung vom Band notwendig.

Tabelle 1 zeigt die verwendeten Dump-Level. Die Tabellen 2, 3 und 4 zeigen die anfallende und gesicherte Datenmenge.

	Mo	Di	Mi	Do	Fr	Sa	So
Komplett	0	0	0	0	0	0	0
Differentiell	0	1	1	1	1	1	1
Inkrementell	0	1	2	3	4	5	6

Tabelle 1: Wochensicherung mit Dump-Leveln

	Mo	Di	Mi	Do	Fr	Sa	So
Komplett	0	0	0	0	0	0	0
Tagesration	1 GByte	1,1 GByte	1,2 GByte	1,3 GByte	1,4 GByte	1,5 GByte	1,6 GByte
Gesamt	1 GByte	2,1 GByte	3,3 GByte	4,6 GByte	6,0 GByte	7,5 GByte	**9,1 GByte**

Tabelle 2: Datenmenge einer Komplettsicherung

Die Komplettsicherung sichert jeden Tag alle Daten. Die Rücksicherung erfolgt vom letzten Band, also dem Sonntagsband.

	Mo	Di	Mi	Do	Fr	Sa	So
Differentiell	0	1	2	3	4	5	6
Tagesration	1 GByte	0,1 GByte	0,1 GByte	0,1 GByte	0,1 GByte	0,1 GByte	0,1 GByte
Gesamt	1 GByte	1,1 GByte	1,2 GByte	1,3 GByte	1,4 GByte	1,5 GByte	**1,6 GByte**

Tabelle 3: Datenmenge einer Differentialsicherung

Differentiell sichert man jeden Montag alle Daten und dienstags bis freitags auf Level 1. Die Rücksicherung erfolgt erst mit dem letzten Montagsband und anschließend dem letzten Tagesband.

	Mo	Di	Mi	Do	Fr	Sa	So
Inkrementell	0	1	1	1	1	1	1
Tagesration	1 GByte	0,1 GByte	0,2 GByte	0,3 GByte	0,4 GByte	0,5 GByte	0,6 GByte
Gesamt	1 GByte	1,1 GByte	1,3 GByte	1,6 GByte	2,0 GByte	2,5 GByte	**3,1 GByte**

Tabelle 4: Datenmenge einer Inkrementalsicherung

Die inkrementelle Sicherung greift sich jeden Montag alle Daten und wochentags die inkrementellen, also nur Veränderungen zum Vortag. Die Rücksicherung erfolgt in der Reihenfolge der Bänder von Montag bis Sonntag.

Für Diskjockeys: Türme von Hanoi

Der Nachteil der bisherigen Strategie ist, dass die Dateien in der Regel nur auf einem einzigen Band liegen, was bei Verlust des Bandes dem Verlust aller Daten eines Tages und der darauffolgenden Tage gleichkommt. Es gibt eine weitere, platzsparende Strategie, die an das Knobelspiel *Türme von Hanoi* angelehnt ist. Diese in Tabelle 5 gezeigte Strategie verteilt einzelne Dateien auf mehrere Bänder. Für einen gesamten Monat lässt sich der Algorithmus mit Level-1-Sicherungen an den folgenden Montagen erweitern. Vorteil dieser Strategie ist, dass die Dateien auf mehr als einem Medium liegen, ohne jedesmal alles sichern zu müssen. Die Verteilung der Dateien auf den Bändern illustriert Tabelle 6.

Mo	Di	Mi	Do	Fr	Sa	So
0	3	2	5	4	7	6
1	3	2	5	4	7	6
1	3	2	5	4	7	6
1	3	2	5	4	7	6

Tabelle 5: Türme-von-Hanoi-Algorithmus

Wochentag	Daten auf dem Tagesband
1. Montag	Montag
Dienstag	Montag & Dienstag
Mittwoch	Montag & Dienstag & Mittwoch
Donnerstag	Mittwoch & Donnerstag
Freitag	Mittwoch & Donnerstag & Freitag
Samstag	Freitag & Samstag
Sonntag	Freitag & Sonnabend & Sonntag
2. Montag	1. Montag – 2. Montag
2. Dienstag	2. Montag & 2. Dienstag
[...]	

Tabelle 6: Datenverteilung im Hanoi-Algorithmus

Über Stefan

Stefan Schumacher ist geschäftsführender Direktor des Magdeburger Instituts für Sicherheitsforschung und gibt zusammen mit Jan W. Meine das Magdeburger Journal zur Sicherheitsforschung heraus. Er befasst sich seit knapp 20 Jahren als Hacker mit Fragen der Informations- und Unternehmenssicherheit und erforscht Sicherheitsfragen aus pädagogisch-psychologischer Sicht. Seine Forschungsergebnisse stellt er auf Fachkongressen und in Publikationen vor. Seine Schwerpunkte liegen auf Social Engineering, Security Awareness, Organisationssicherheit, internationale Cyber-Security und Mensch-Maschine-Interaktion.