ulm university universität ulm

Angst vor dem Datengau?
Tipps und Tricks für einen besseren Schlaf

Thomas Nau, kiz (Thomas.Nau@uni-ulm.de)

Timeline

- über uns und mich
- Ransomware und andere Probleme
- Gründe für ZFS (und Solaris)
- Datenmanagement und andere Mythen
- Die Grenzen von herkömmlichen Backuplösungen
- wrap-up
- Q & A

Über mich

- eigentlich Physiker
- stlv. Direktor des kiz und Leiter der Abteilung „Infrastruktur“
  - Balanceakt zwischen Technik und Management
- erste IT Berührung mit einer PDP-11 vor (sehr) langer Zeit
- Schwerpunkte UNIX und Storage Lösungen

Safe Harbor Statement

Meine Aussagen geben meine persönliche Meinung und Erfahrung wieder

One team to serve them all

- Die „Abteilung Infrastruktur“ des kiz erbringt Dienstleistungen für ~15.000 Personen
  - Netzwerke (LAN, MAN und WLAN)
    - Anbindung an BelWue mit 10Gbit, in wenigen Tagen 100Gbit
    - Telefonie mit 14.000 Anschlüssen
  - Rechnerbetrieb
    - gesamte zentrale Server-IT inklusive der Universitätsverwaltung
    - ca. 600 Desktop PCs und Notebooks mit Windows und Linux
  - Dienstleistungen im Rahmen von Landeskooperationen
    - Backup Service für mehrere Universitäten in Baden-Württemberg
    - HPC Landes-Cluster mit Schwerpunkt „Theoretische Chemie“
  - bedienen Cloud-Hype, Next-Hype, Whatever-else Hype

Die Vorteile einer Abteilung

- spannendes Arbeitsumfeld mit vielen Freiheiten für Leute die bereit sind Verantwortung zu übernehmen
  - „erlaubt ist was funktioniert und wartbar ist“
    bleeding edge ist kein Tabu
  - natürlich gibt es auch Ausnahmen: „Der Chef hat immer Recht“
- ein Team, ein Ziel
  - Server- und Desktop-Betrieb, IT-Security, Netz- und TK-Gruppen sind Bestandteil des Teams
    - kein Elfenbeinturm sondern **„reality exposure“** und **„eat your own dog food“** Philosophie
    - Entscheidungen sind einfacher zu treffen und gemeinsam zu tragen
    - hilft realistische und umsetzbare Ziele zu definieren

Mitwirkung: „Baden-Württemberg (BW)“ Projekte

- Belwue — Landes Hochschulnetz
- bwIDM — föderiertes Identity Management

- bwNet100G — next generation networking; redundante 100G Anbindung der Universitäten
- bwHPC — verteiltes Hochleistungsrechnen
- bwBackup — multi-site Backup
- bwCloud — OpenStack Cloud

Solaris als bewusste strategische Entscheidung

- Stabilität und Datensicherheit sind Schlüsselfaktoren; nahezu alle zentralen Server basieren auf Solaris
  - SAP/Oracle, MySQL, PostgreSQL, Apache/PHP, Typo3, Cyrus IMAP, NFS, CIFS, iSCSI, Bacula Enterprise Backup, LDAP, …
  - Ausnahmen: Windows Active Directory und Citrix XenServer
    - letzterer basiert auf Solaris ZFS Storage
- 100% aller Solaris Systeme verwenden 11.3 (oder neuer)
  - meist x86 aber auch noch wenige SPARC basierte Systeme
- jüngste Oracle Entscheidungen stimmen nachdenklich
  - Zitat meines Vortrags 2014:
    „Umgang damit zeigt das Oracle Solaris sehr ernst nimmt“
  - Festhalten an ZFS als Schlüsseltechnologie

Ransomware und andere „Troublemaker“

Vorwort

- gezielte Angriffe, etwa Spear-Phishing, gegen Personen oder Organisationen sind nahezu unmöglich abzuwehren und nicht Bestandteil der nächsten 40 Minuten
  - ist gleichzusetzen mit freiwilliger Übergabe der Schlüssel
- Fokus liegt für heute auf „Schadensbegrenzung“
- must see: „Hacking your mind and emotions“

  https://www.usenix.org/conference/lisa13/hacking-your-mind-emotions

Wikipedia Definition

Ransomware (von englisch ransom für „Lösegeld“), auch Erpressungstrojaner, Erpressungssoftware, Kryptotrojaner oder Verschlüsselungstrojaner, sind Schadprogramme, mit deren Hilfe ein Eindringling den Zugriff des Computerinhabers auf Daten, deren Nutzung oder auf das ganze Computersystem verhindern kann. Dabei werden private Daten auf dem fremden Computer verschlüsselt oder der Zugriff auf sie verhindert, um für die Entschlüsselung oder Freigabe ein Lösegeld zu fordern.

Ransomware, Trojaner, Würmer, ...

- existieren seit über drei Jahrzehnten
- Ransomware wurde in den letzten Jahren zum Massenphänomen
- Opfer sind meist zufällig, gezielte Angriffe auf Unternehmen sind (waren) eher die Ausnahme
- Einfallstore sind vielfältig
  - zero-day exploits → wurmartige Verbreitung
  - drive-by download
  - E-Mail Attachments oder „klick-mich Seiten“
  - ungeschultes Personal

Die Layer-8 Seite des Problems

- Problem nach unserer Erfahrung unabhängig von Ausbildung, Alter, Geschlecht, Position, ...

Universität Ulm

Ihr Webmail-Kontingent überschritten hat, die Quote ist 20 GB. Sie laufen zur Zeit auf 20.9GB.

Zur Reaktivierung und erhöhen Sie Ihre Webmail-Quote unter dem Link klicken.
https://uniulmde.seamlessdocs.com/f/uniulmde
Geschieht dies nicht, kann die Kündigung Ihres Webmail führen

Vielen Dank und sorry für die Unannehmlichkeiten
localhost
© 2017 Universität Ulm | Ulm University

Was tun?

- propagierte Schutzmaßnahmen, etwa Virenscanner ..., begrenzen allenfalls das Ausmaß und Einfallstore
  - warum konnten sich Petya, WannaCry, ... so schnell verbreiten?
- essentiell für die Minimierung des Problems und der Folgen sind
  - zeitnahe Erkennung von Angriffen
    - keine Patentlösung; Zeitfenster beeinflusst Schadenspotential und Backupstrategie
  - Vorhalten **unabhängiger schreibgeschützter** Daten Kopien
  - Wissen um Zugriffsrechte
    - Gruppenverzeichnisse, Netzlaufwerke, ...
  - Organisatorische Entscheidungen: „easy of use“ vs. Sicherheit

Solaris to your rescue

Solaris to your rescue

- Darren Moffat beschreibt bereits 2008(!) in seinem Blog „*Making files on ZFS Immutable (even by root!)*“
  - quick poll: wer nutzt dieses oder vergleichbare Möglichkeiten?
- Tipp: die ZFS basierten Maßnahmen lassen sich auch weitestgehend mit FreeBSD und Linux umsetzen

Solaris to your rescue: ZFS snapshots

- automatisierte ZFS snapshots auf Basis der SMF-Services

  svc:/application/time-slider:default
  *svc:/system/filesystem/zfs/auto-snapshot:**

- diese übernehmen das Management hinsichtlich der Erzeugung und Löschung von snapshots nach vordefinierten Regeln
- im nachfolgenden Beispiel
  - 23 stündliche snapshots
  - 28 tägliche snapshots
- Achtung: snapshots sind kein Backup, lediglich dessen Ergänzung

Solaris to your rescue: ZFS snapshots

```
INSTANCE=nfs
IH=${INSTANCE}-hourly
ID=${INSTANCE}-daily

svccfg -s svc:/system/filesystem/zfs/auto-snapshot add ${IH}
svccfg -s auto-snapshot:${IH} addpg zfs application
svccfg -s auto-snapshot:${IH} setprop zfs/interval = hours
svccfg -s auto-snapshot:${IH} setprop zfs/keep = 23
svccfg -s auto-snapshot:${IH} setprop zfs/period = astring: 1
svccfg -s auto-snapshot:${IH} addpg general framework
svccfg -s auto-snapshot:${IH} setprop general/complete = astring: \"\"
svcadm refresh svc:/system/filesystem/zfs/auto-snapshot:${IH}

svccfg -s svc:/system/filesystem/zfs/auto-snapshot add ${ID}
svccfg -s auto-snapshot:${ID} addpg zfs application
svccfg -s auto-snapshot:${ID} setprop zfs/interval = days
svccfg -s auto-snapshot:${ID} setprop zfs/keep = 28
svccfg -s auto-snapshot:${ID} setprop zfs/period = astring: 1
svccfg -s auto-snapshot:${ID} addpg general framework
svccfg -s auto-snapshot:${ID} setprop general/complete = astring: \"\"
svcadm refresh svc:/system/filesystem/zfs/auto-snapshot:${ID}
```

Solaris to your rescue: ZFS snapshots

- bei den ersten Schritten ist es sinnvoll den Service im debugging Modus zu betreiben

  svccfg -s time-slider setprop daemon/verbose = true

- dann können alle Services aktiviert werden

  svcadm refresh svc:/application/time-slider:default
  svcadm enable svc:/application/time-slider:default
  svcadm enable svc:/system/filesystem/zfs/auto-snapshot:${IH}
  svcadm enable svc:/system/filesystem/zfs/auto-snapshot:${ID}

- die Benennung der snapshots ist einfach und eindeutig

  pool1/home/kiz@zfs-auto-snap_nfs-hourly-2017-09-24-16h09
  pool1/home/kiz@zfs-auto-snap_nfs-daily-2017-09-24-16h58
  pool1/home/kiz@zfs-auto-snap_nfs-hourly-2017-09-24-17h09
  pool1/home/kiz@zfs-auto-snap_nfs-hourly-2017-09-24-18h09

Wo ist der Zusammenhang zu Ransomware, ...?

Solaris to your rescue: ZFS snapshots

- typische Schritte von Ransomware nach erfolgreichen Zugriff auf das System
  - Öffnen der Quelldatei *S*
  - Anlegen und Öffnen einer Zieldatei *T* deren Namen aus dem verschlüsselten Dateinamen von *S* besteht
    - im Allgemeinen ist ein encoding notwendig um spezielle Zeichen wie „:“ zu eliminieren
  - Verschlüsselung der Daten aus *S* nach *T*
  - Löschen der Quelldatei *S*
- gezielte Angriffe laufen subtiler und langsamer ab

Solaris to your rescue: ZFS snapshots

```
# ls -l
total 49
-rw-r--r--  1 root  root 15205 Sep 24 19:38 etc.txt
-rw-r--r--  1 root  root  8694 Sep 24 19:37 var.tmp.txt

# for s in *.txt; do
    t=$(echo $s |
        encrypt -a aes -k /tmp/MyKey |
        base64 -w 0 | tr '/' '@')
    encrypt -a aes -k /tmp/MyKey -i "$s" -o "$t"
    rm "$s"
done

# ls -l
total 49
-rw-r--r--  1 root  root  8744 Sep 24 19:40 AAAAAQAAA+h07s8rfI…
-rw-r--r--  1 root  root 15256 Sep 24 19:40 AAAAAQAAA+jUjV4IEU…
```

Solaris to your rescue: ZFS snapshots

- wenig bekannt und genutzt:

```
# zfs diff -o user -o oldname -o name rpool/test@snap
M       root    -       /test/
-       root    -       /test/var.tmp.txt
-       root    -       /test/etc.txt
+       root    -       /test/AAAAAQAAA+jUjV4IEU…
+       root    -       /test/AAAAAQAAA+h07s8rfI…
```

- bietet noch einige weiter gehende Informationen
  - Umbenennung von Dateien, ...
  - mit den zusätzlich einzublendenden Zeitstempeln lässt sich der Ablauf besser rekonstruieren

Solaris to your rescue: ZFS snapshots

- Schritte zur Rekonstruktion
  - System abschotten
  - wann hat der Angriff begonnen?
  - Identifikation aller seither geänderten Dateien (*zfs diff*)
  - Unterscheidung verdächtiger vs. normaler Änderung
  - Suche nach intakter Kopie in zurückliegenden snapshots
    - falls erfolglos als Kandidat für „*in Backup suchen*“ Liste notieren
  - Datei wieder herstellen
  - Einfallstor finden und schließen
- Geschwindigkeit bzw. „Qualität“ des Angriffs und Intervalle der Snapshots bestimmen den potentiellen Datenverlust

Solaris to your rescue: ZFS snapshots

- regelmäßige snapshots sind nicht teuer, mit ZFS sehr schnell erzeugt und bieten erheblichen Mehrwert
- Beispiele
  - Mail-Server: 23 stündliche und 90 täglich snapshots; jede Mail (auch SPAM) wird 24h verzögert gelöscht und findet sich damit in einem snapshot
  - CIFS-Server für 200 Arbeitsplätze: 23 stündliche und 28 täglich snapshots

```
# zfs list -o space -r data2/mail
NAME         AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
data2/mail   50.0T  10.9T      5.35T   5.57T              0          0

# zfs list -o space -r smb/cifs
NAME         AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
smb/cifs     27.2T  6.25T      1019G   5.26T              0          0
```

Erwiesenermaßen funktionieren diese Schritte in der Praxis!

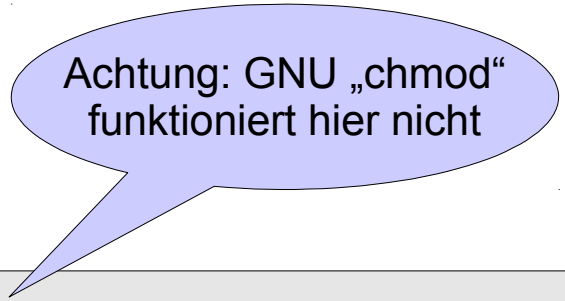
Empfehlung: ZnapZend

- ZnapZend (vom RRDtool Entwickler Tobias Oetiker)
  - ersetzt *time-slider* vollständig
  - verschlüsselte Replikation mittels *zfs send/receive* und *ssh*
  - feingranulare Definition von Intervallen auf Quell- **und** Ziel-System
  - http://www.znapzend.org/

Solaris to your rescue: ZFS Attribute

- *Immutable (Global) Zones* wären ideal…
  - leider eignet sich nicht jede Anwendung dafür
- Verwendung von ZFS Attributen kann die tägliche Arbeit einschränken
  - Haupteinsatzgebiet ist der Schutz von Installationen und Konfigurationen gegen Manipulation
    - Bsp.: PHP-Installationen und Web-Server Log-Dateien
- die Verwendung einiger Attribute erfordert besondere Privilegien eines accounts oder *root*-Rechte
  - appendonly, nounlink, immutable
- nach dem Setzen dieser Attribute: testen, testen, testen
  - log-rotation, backup, restore, ...

Solaris to your rescue: ZFS Attribute




```
root@alderaan:/test# chmod S+vappendonly,vnounlink logfile

root@alderaan:/test# ls -/v logfile
-rw-r-----   1 nau      kizinfra    1808 Sep 25 10:45 logfile
    {archive,nohidden,noreadonly,nosystem,appendonly,
     nonodump,noimmutable,av_modified,noav_quarantined,
     nounlink,nooffline,nosparse,nosensitive}
```

Solaris to your rescue: ZFS Attribute

```
nau@alderaan:/test> > logfile
bash: logfile: Not owner

nau@alderaan:/test> rm logfile
rm: logfile not removed: Not owner

nau@alderaan:/test> mv logfile tmp123456
mv: cannot rename logfile to tmp123456: Not owner

nau@alderaan:/test> echo A_NEW_LINE_APPENDED >> logfile


root@alderaan:/test# rm logfile
rm: remove logfile (yes/no)? y
rm: logfile not removed: Not owner
```

Solaris to your rescue: ZFS Attribute

```
root@alderaan:/test> chmod S+vimmutable httpd.conf

root@alderaan:/test> > httpd.conf
-bash: httpd.conf: Not owner

root@alderaan:/test> rm httpd.conf
rm: httpd.conf: override protection 644 (yes/no)? y
rm: httpd.conf not removed: Not owner


# unlocking / editing / locking

root@alderaan:/test> chmod S-vimmutable httpd.conf

root@alderaan:/test> vi httpd.conf

root@alderaan:/test> chmod S+vimmutable httpd.conf
```

Organisatorische Fragen / Aufgaben etwa zum Datenmanagement

Datenmanagement, meist ungeliebt ...

- das Wissen um die eigenen Daten, ihre Struktur, … ist nicht nur für die zuvor beschriebenen Szenarien essentiell
- es existieren viele, oft weit zurückreichende, Mythen
- technische und organisatorisch Entscheidungen müssen regelmäßig auf den Prüfstand gestellt und mit der Realität abgeglichen werden
  - im Bereich der Datensicherung kommen oft die selben grundlegenden Techniken zum Einsatz wie bereits vor 15 Jahren
- nachfolgend einige Mythen (nicht nur) aus meinem „*Backup Umfeld*“

Mythos #1

- *„Nutzer kennen ihre Daten, die sich daraus ergebenden Anforderungen und folgen einem Datenmanagementkonzept.“*
- grundlegende Fragen sind u.a.
  - Welche Daten sind heiß/wichtig und wie oft wird diese Frage neu beantwortet?
    → Setzen von Prioritäten bei Wiederherstellung
  - Liegen Daten strukturiert/chaotisch vor (S*potlight* Problem)?
    → einfache Auswahl und Zeitgewinn bei Wiederherstellung
  - Wie lange kann eine Gruppe, ein Institut oder eine Einrichtung ohne Daten produktiv arbeiten?
    → entscheidet über Technologie

Mythos #2

- *„Die Netzwerkbandbreite ist der begrenzende Faktor für Backups und andere Sicherungen. Clients brauchen 10GE!“*
- kann in Einzelfällen zutreffen
- im Allgemeinen limitiert das Durchsuchen des zu Grunde liegende Filesystems die Geschwindigkeit

```
Elapsed time:              3 hours 39 mins 57 secs
Priority:                  10
SD Files Written:          552,535
SD Bytes Written:          310,878,063,026 (310.8 GB)
Rate:                      23540.0 KB/s
```

→ für das Zurückspielen muss **mindestens** die selbe Zeit angesetzt werden

Mythos #3

- *„Archiv ist doch nur Backup mit langen Lagerzeiten.“*
- Bibliothekare haben im Vergleich zur IT-Welt ein grundlegend anders Verständnis von Archiven
- wesentliche Unterschiede sind
  - Archivdaten sind mit Metadaten angereichert
    → vereinfacht die Suche erheblich
  - Archivdaten werden vom Quellsystem gelöscht
    → damit werden sie nach einiger Zeit auch im Backup gelöscht
- Anwender sagen *„Archivdaten“* meinen aber *„kalte Daten“*
  → zurück zu Mythos #1

Mythos #4

- *„Es existiert ein Plan für den Fall der Fälle.“*
- jeder Plan bedarf regelmäßiger realitätsnaher Tests
  - System (beschaffen und) wieder herstellen
  - Daten wieder herstellen
  - Anwendung mit restaurierten Daten „bekannt machen“
    - transaction logs, ...
- kritische Punkte
  - Schlüsselmanagement im Falle verschlüsselter Daten
  - Priorisierung der Daten für die Wiederherstellung
  - die Zeiten, die für die Wiederherstellung benötigt werden
    - Ersatzhardware usw. ggf. mit einrechnen

Lessons learned

- herkömmliche Backup Methoden sind nicht in alle Fällen ausreichend
- für Fileserver, SAP Systeme, … werden sie ergänzt durch
  - ZFS snapshots
  - asynchrone Replikation für wichtige Systeme
- für Mailserver werden sie ersetzt durch
  - ZFS Snapshots
  - delayed expunge
  - semi-synchrone application-layer Replikation für Mail Systeme
- Sensibilisierung der Mitarbeiter ist ein Schlüsselfaktor

Danke für's Zuhören
und
Danke an Harald Däubler