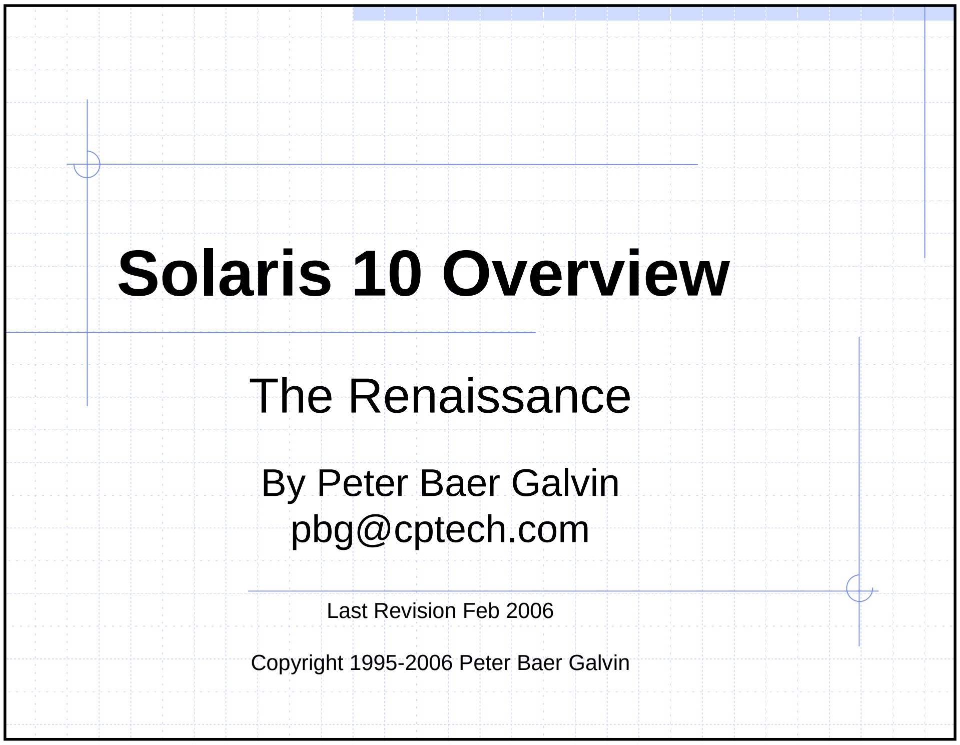
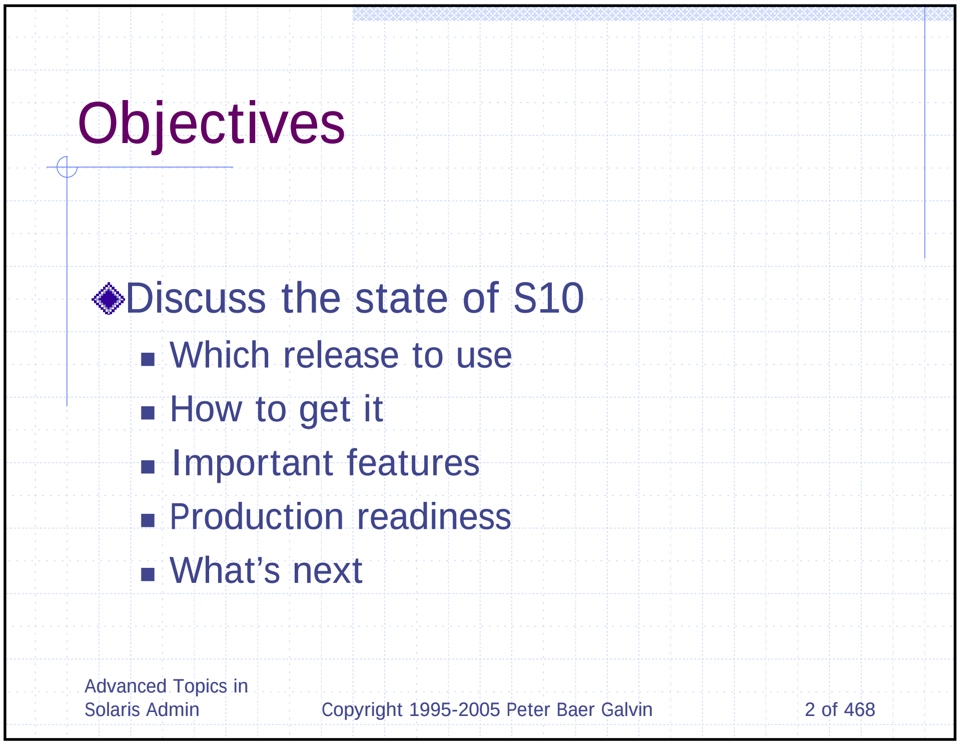


Solaris 10 Overview

The Renaissance

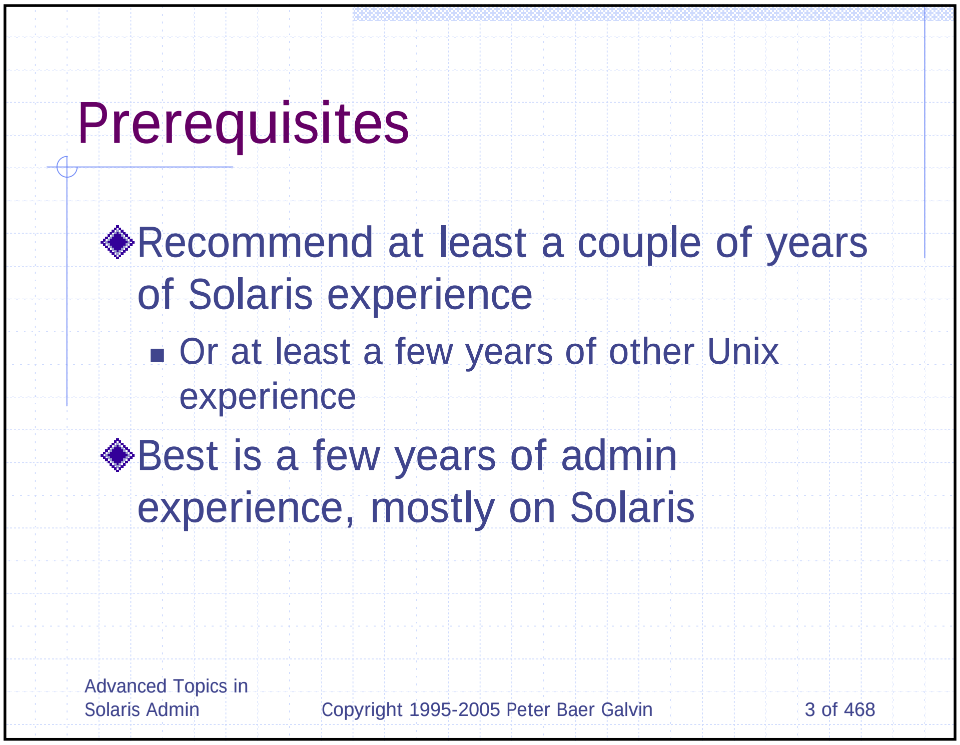
By Peter Baer Galvin
pbg@cptech.com

Last Revision Feb 2006



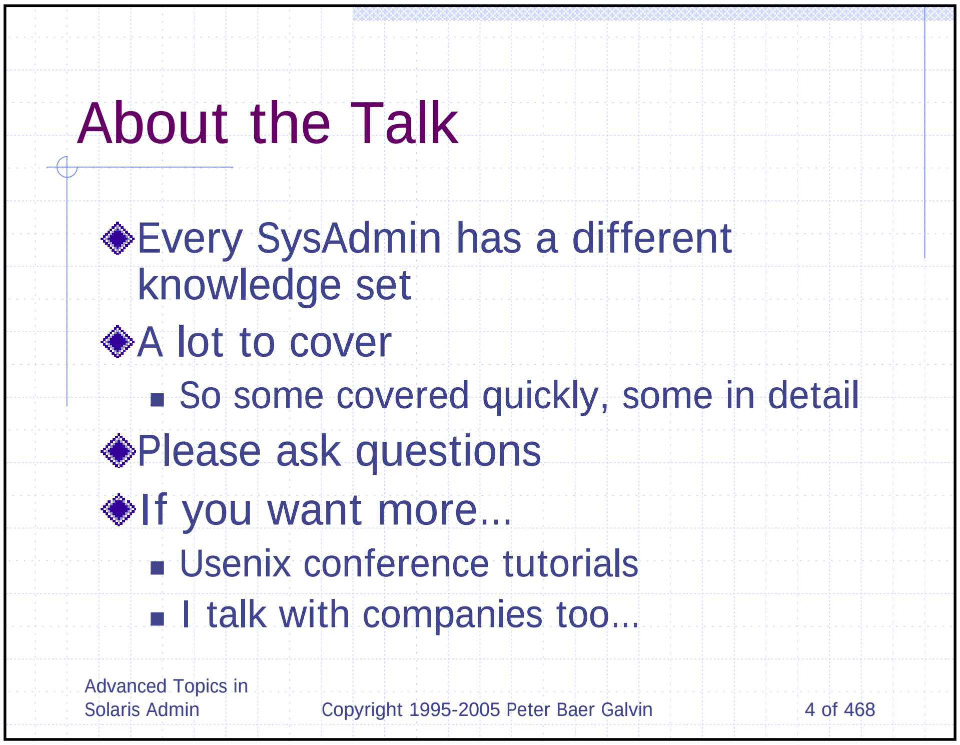
Objectives

- Discuss the state of S10
  - Which release to use
  - How to get it
  - Important features
  - Production readiness
  - What’s next



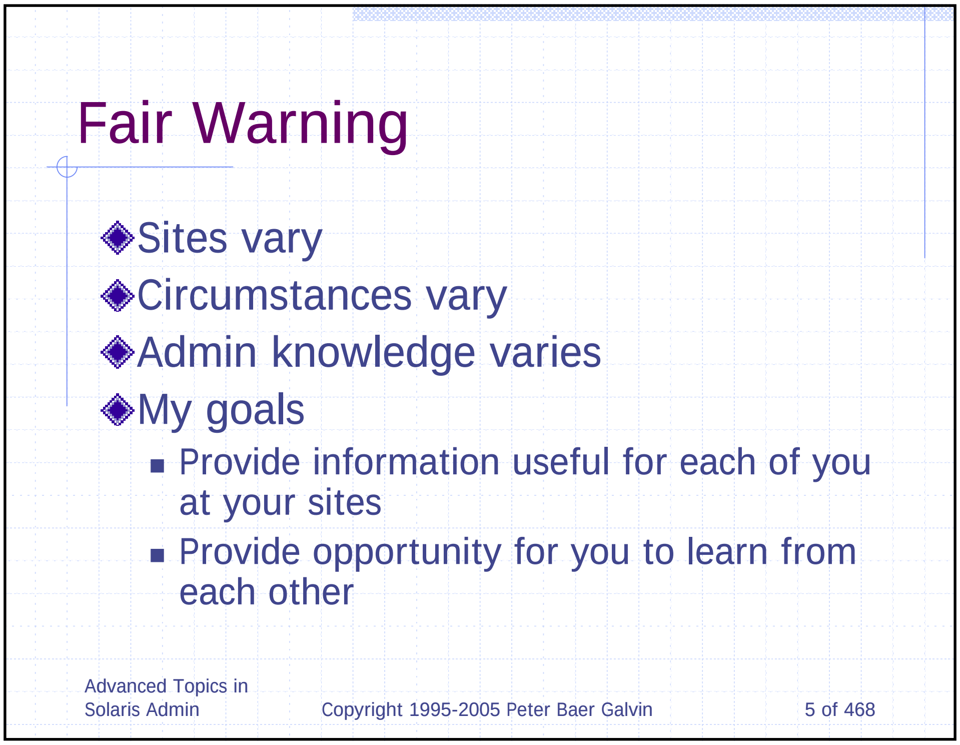
Prerequisites

- Recommend at least a couple of years of Solaris experience
  - Or at least a few years of other Unix experience
- Best is a few years of admin experience, mostly on Solaris



About the Talk

- Every SysAdmin has a different knowledge set
- A lot to cover
  - So some covered quickly, some in detail
- Please ask questions
- If you want more...
  - Usenix conference tutorials
  - I talk with companies too...



Fair Warning

- Sites vary
- Circumstances vary
- Admin knowledge varies
- My goals
  - Provide information useful for each of you at your sites
  - Provide opportunity for you to learn from each other

Why Listen to Me

- 20 Years of Sun experience
- Seen much as a consultant
- Hopefully, you've used:
  - The Solaris Corner @ www.samag.com
  - The Solaris Security FAQ
  - SunWorld “Pete's Wicked World”
  - SunWorld “Pete's Super Systems”
  - Unix Secure Programming FAQ
  - Operating System Concepts (The Dino Book), 7th ed
  - Applied Operating System Concepts

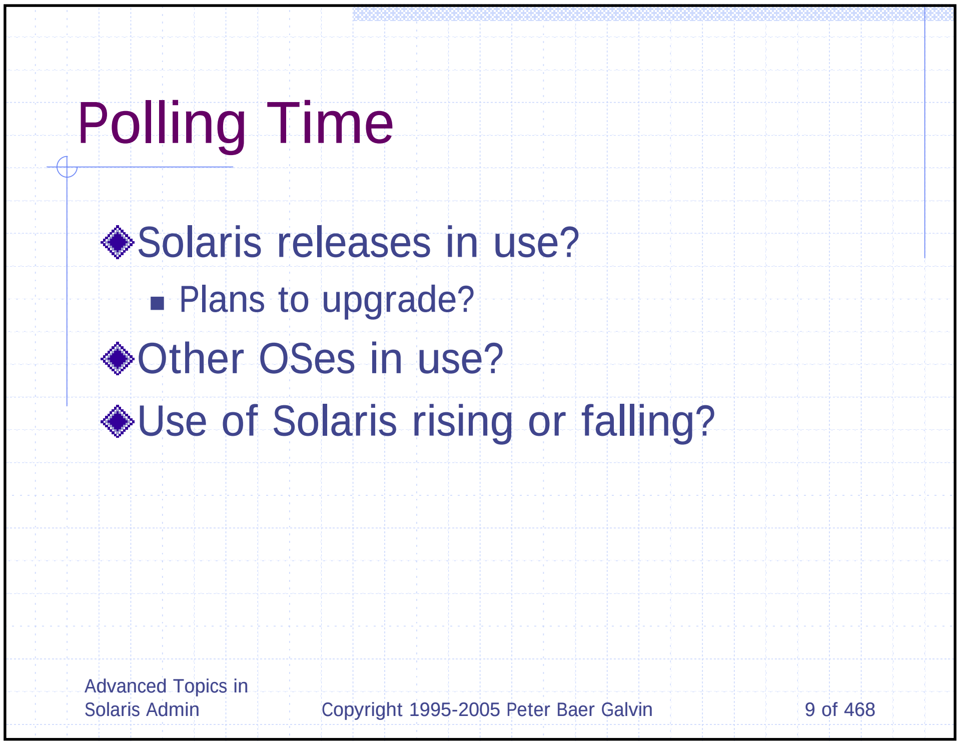
APPLIED OPERATING SYSTEM CONCEPTS
OPERATING SYSTEM CONCEPTS

Overview

Lay of the Land

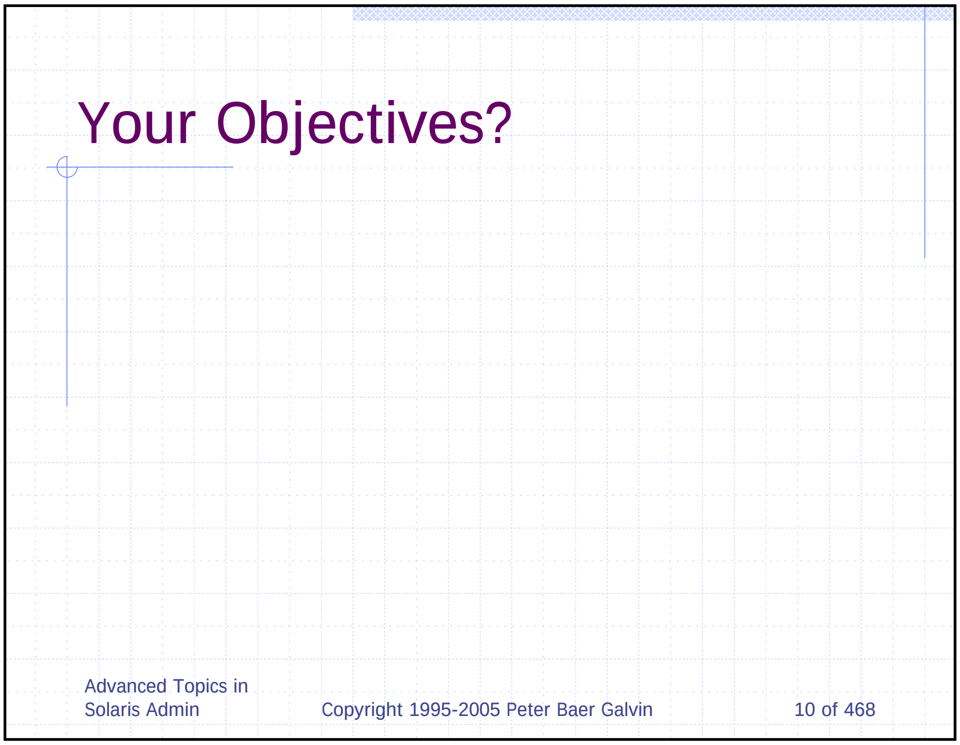
Outline

- Releases
- Reliability
  - SMF / FMA
- Performance
  - FireEngine
  - Dtrace
- Security
  - Zones / containers
  - Least privilege
- Usability
  - ZFS
- Philosophy



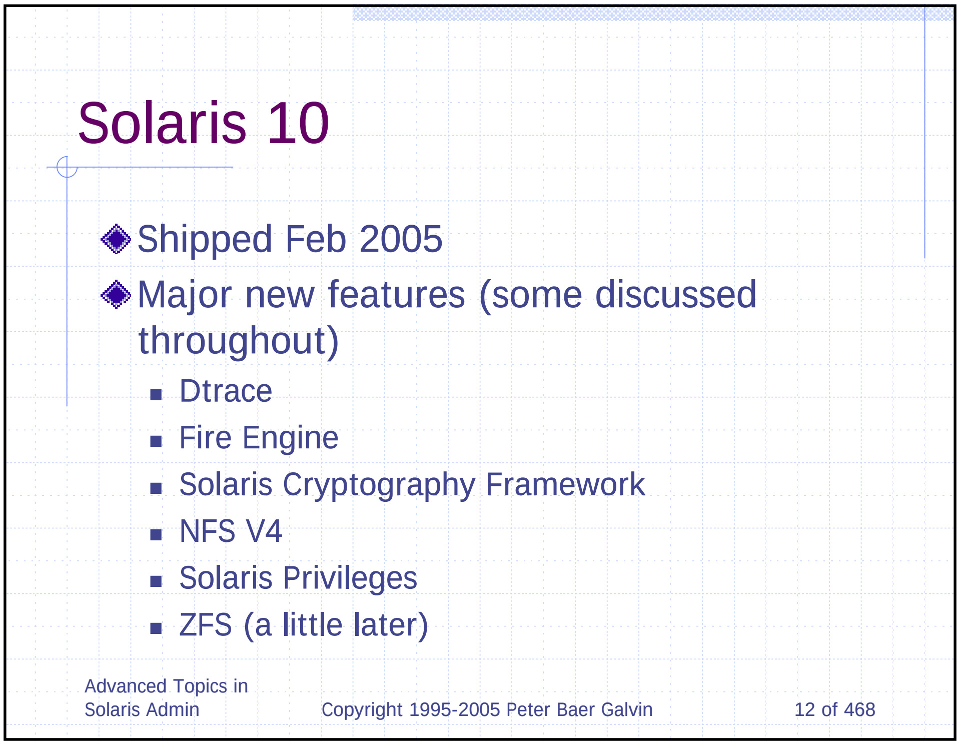
Polling Time

- Solaris releases in use?
  - Plans to upgrade?
- Other OSes in use?
- Use of Solaris rising or falling?



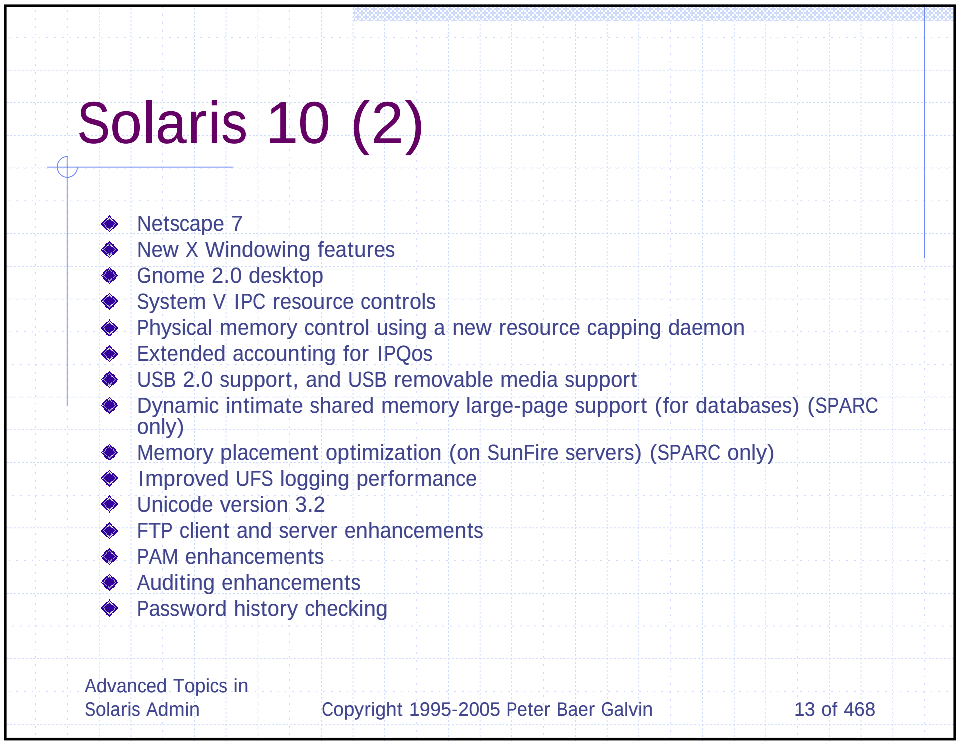
Your Objectives?

Releases



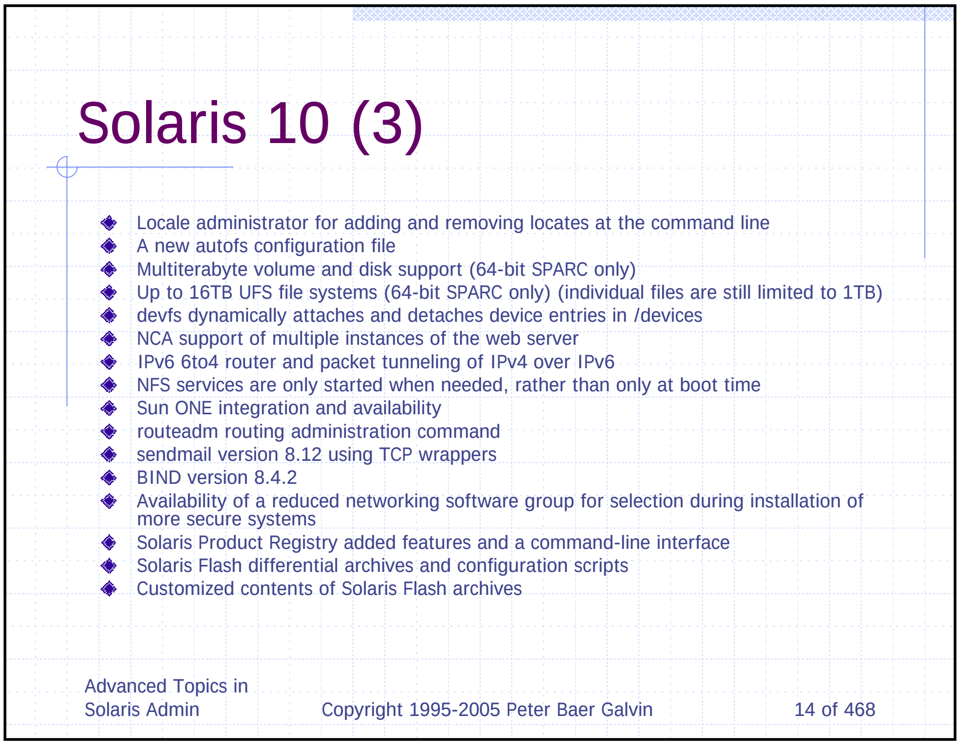
Solaris 10

- Shipped Feb 2005
- Major new features (some discussed throughout)
  - Dtrace
  - Fire Engine
  - Solaris Cryptography Framework
  - NFS V4
  - Solaris Privileges
  - ZFS (a little later)



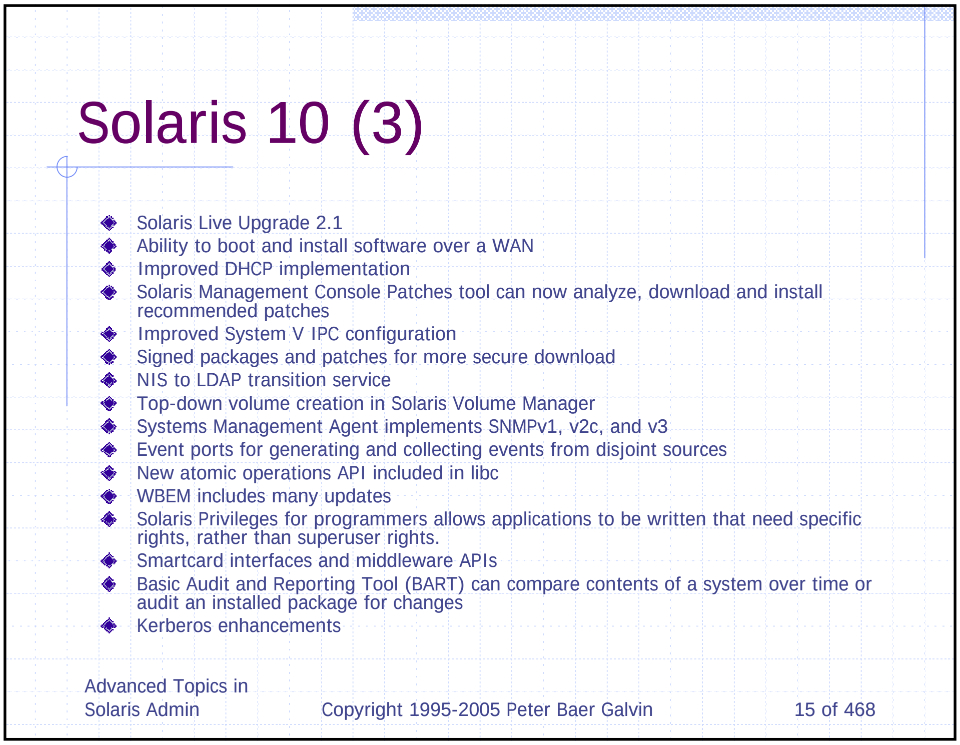
Solaris 10 (2)

- Netscape 7
- New X Windowing features
- Gnome 2.0 desktop
- System V IPC resource controls
- Physical memory control using a new resource capping daemon
- Extended accounting for IPQos
- USB 2.0 support, and USB removable media support
- Dynamic intimate shared memory large-page support (for databases) (SPARC only)
- Memory placement optimization (on SunFire servers) (SPARC only)
- Improved UFS logging performance
- Unicode version 3.2
- FTP client and server enhancements
- PAM enhancements
- Auditing enhancements
- Password history checking



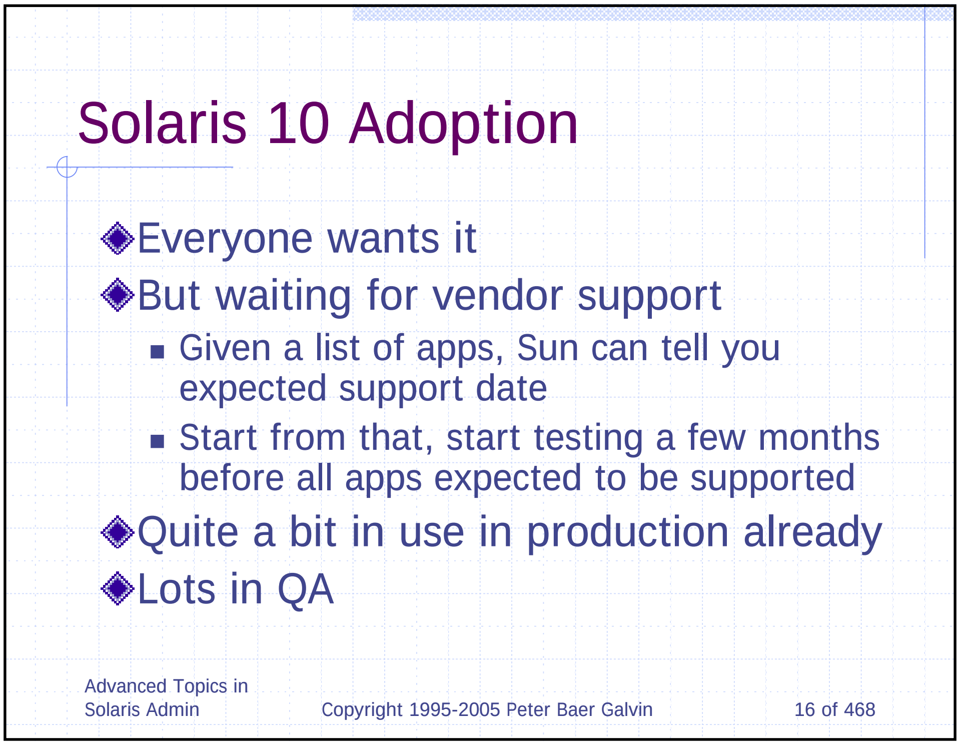
Solaris 10 (3)

- Locale administrator for adding and removing locates at the command line
- A new autofs configuration file
- Multiterabyte volume and disk support (64-bit SPARC only)
- Up to 16TB UFS file systems (64-bit SPARC only) (individual files are still limited to 1TB)
- devfs dynamically attaches and detaches device entries in /devices
- NCA support of multiple instances of the web server
- IPv6 6to4 router and packet tunneling of IPv4 over IPv6
- NFS services are only started when needed, rather than only at boot time
- Sun ONE integration and availability
- routeadm routing administration command
- sendmail version 8.12 using TCP wrappers
- BIND version 8.4.2
- Availability of a reduced networking software group for selection during installation of more secure systems
- Solaris Product Registry added features and a command-line interface
- Solaris Flash differential archives and configuration scripts
- Customized contents of Solaris Flash archives



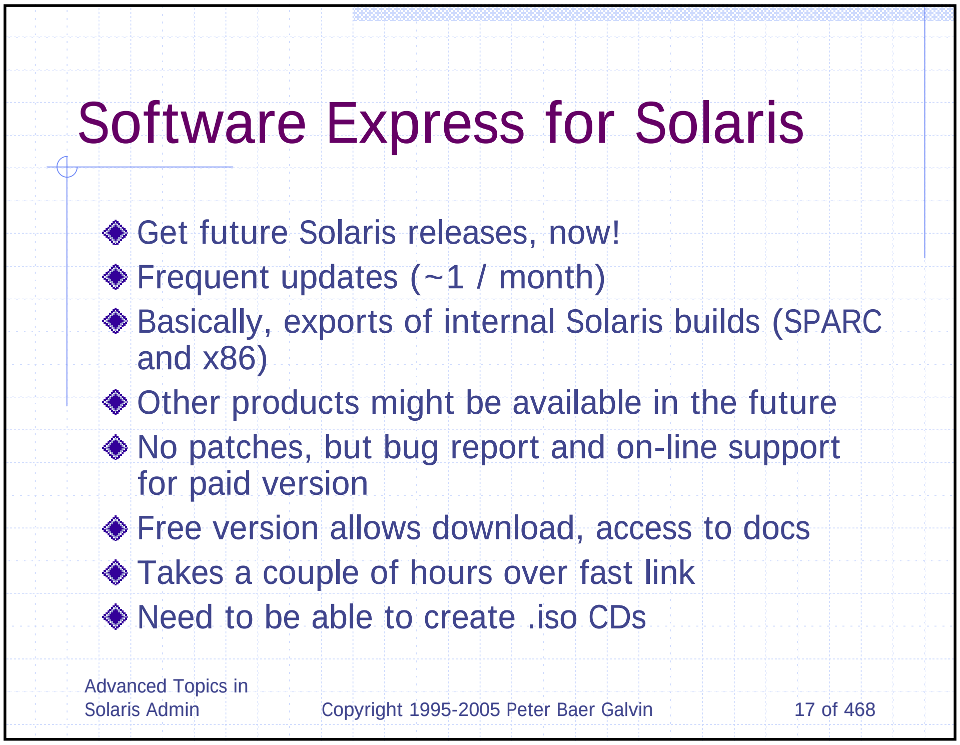
Solaris 10 (3)

- Solaris Live Upgrade 2.1
- Ability to boot and install software over a WAN
- Improved DHCP implementation
- Solaris Management Console Patches tool can now analyze, download and install recommended patches
- Improved System V IPC configuration
- Signed packages and patches for more secure download
- NIS to LDAP transition service
- Top-down volume creation in Solaris Volume Manager
- Systems Management Agent implements SNMPv1, v2c, and v3
- Event ports for generating and collecting events from disjoint sources
- New atomic operations API included in libc
- WBEM includes many updates
- Solaris Privileges for programmers allows applications to be written that need specific rights, rather than superuser rights.
- Smartcard interfaces and middleware APIs
- Basic Audit and Reporting Tool (BART) can compare contents of a system over time or audit an installed package for changes
- Kerberos enhancements



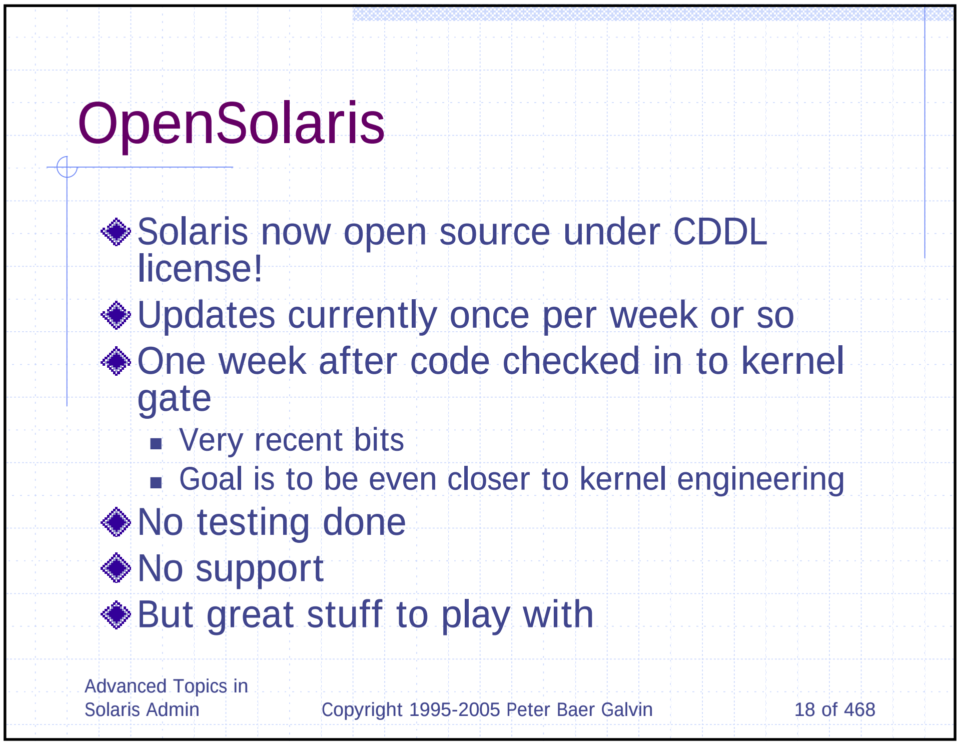
Solaris 10 Adoption

- Everyone wants it
- But waiting for vendor support
  - Given a list of apps, Sun can tell you expected support date
  - Start from that, start testing a few months before all apps expected to be supported
- Quite a bit in use in production already
- Lots in QA



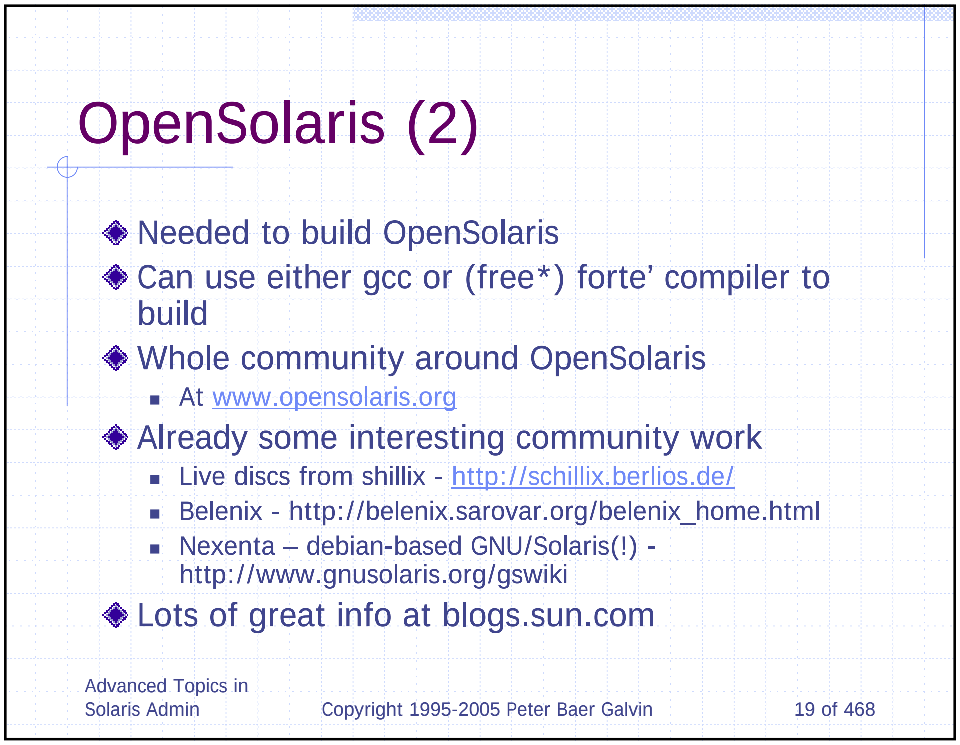
Software Express for Solaris

- Get future Solaris releases, now!
- Frequent updates (~1 / month)
- Basically, exports of internal Solaris builds (SPARC and x86)
- Other products might be available in the future
- No patches, but bug report and on-line support for paid version
- Free version allows download, access to docs
- Takes a couple of hours over fast link
- Need to be able to create .iso CDs



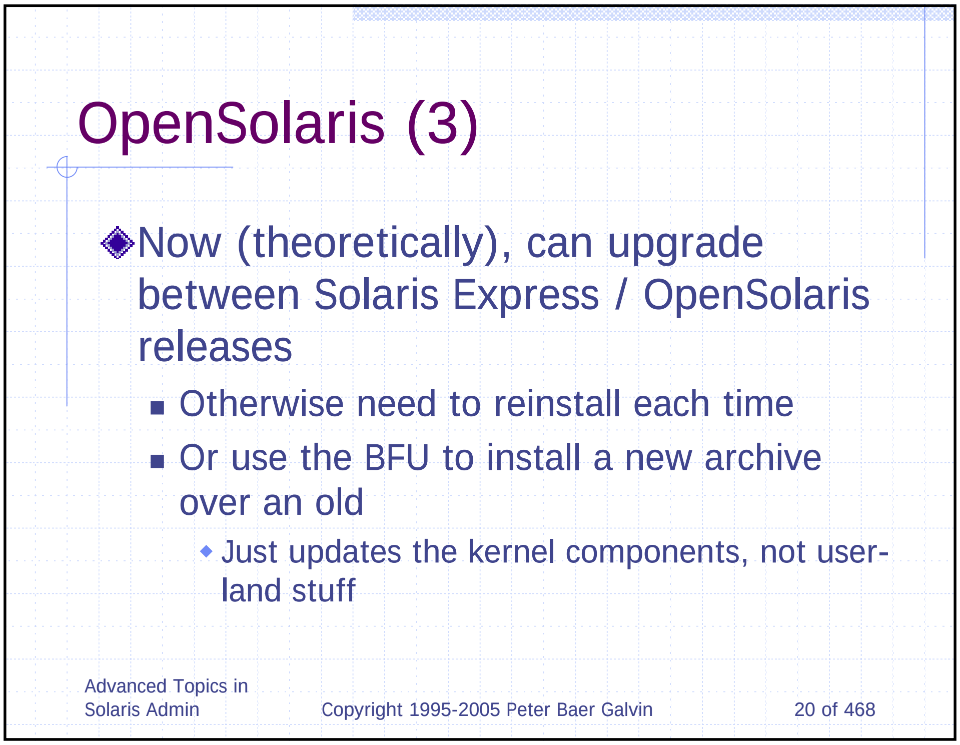
OpenSolaris

- Solaris now open source under CDDL license!
- Updates currently once per week or so
- One week after code checked in to kernel gate
  - Very recent bits
  - Goal is to be even closer to kernel engineering
- No testing done
- No support
- But great stuff to play with



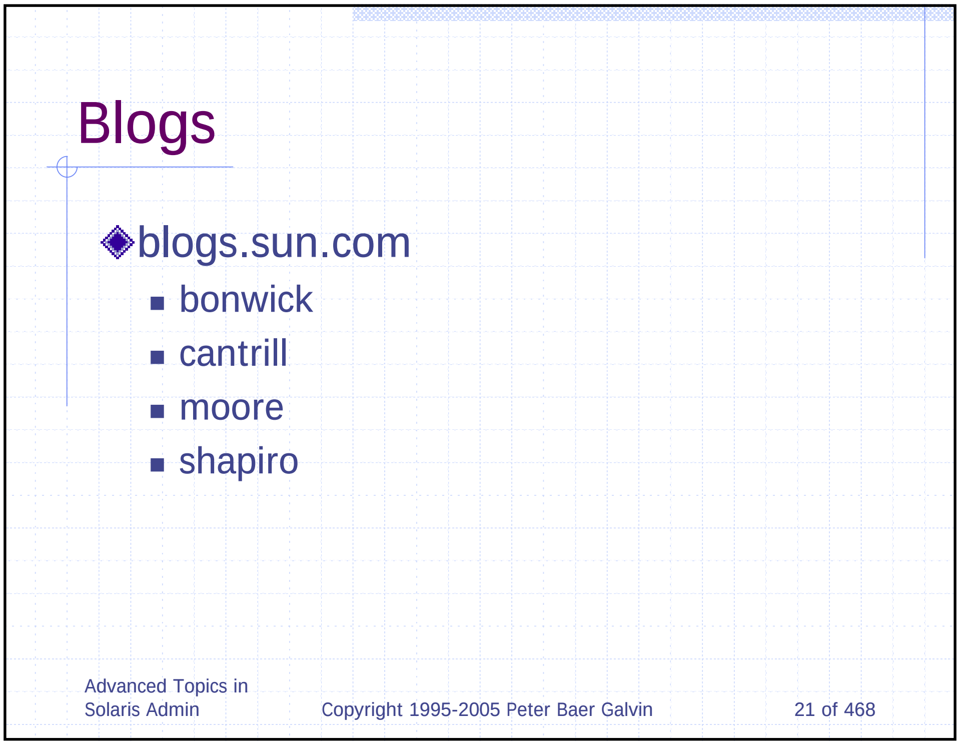
OpenSolaris (2)

- Needed to build OpenSolaris
- Can use either gcc or (free*) forte’ compiler to build
- Whole community around OpenSolaris
  - At www.opensolaris.org
- Already some interesting community work
  - Live discs from shillix - http://schillix.berlios.de/
  - Belenix - http://belenix.sarovar.org/belenix_home.html
  - Nexenta – debian-based GNU/Solaris(!) - http://www.gnusolaris.org/gswiki
- Lots of great info at blogs.sun.com



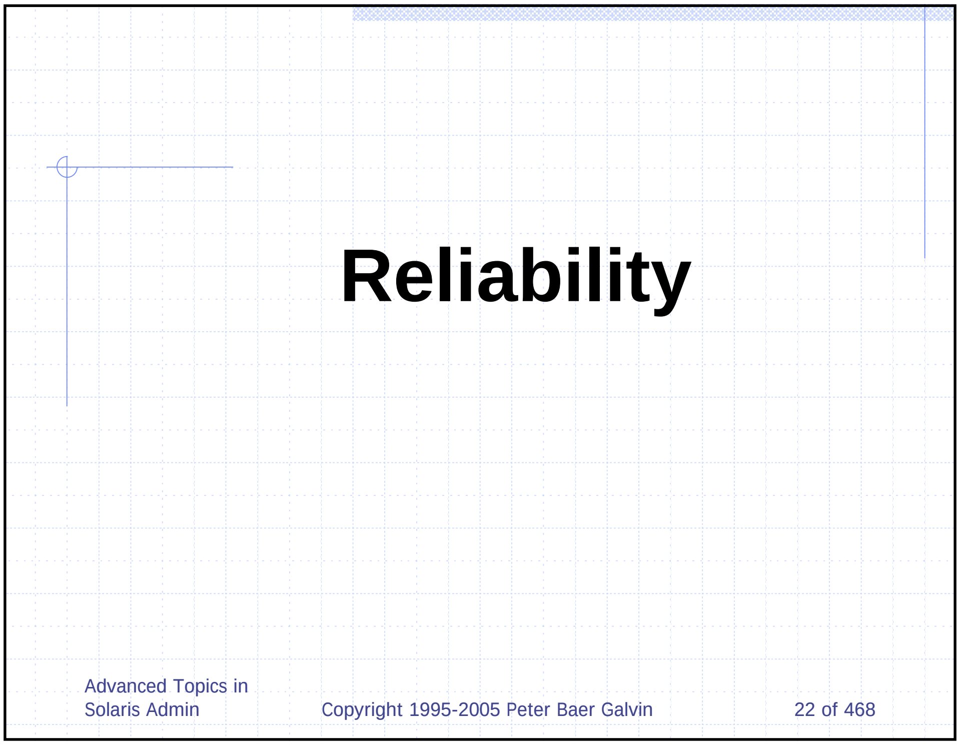
OpenSolaris (3)

- Now (theoretically), can upgrade between Solaris Express / OpenSolaris releases
  - Otherwise need to reinstall each time
  - Or use the BFU to install a new archive over an old
    - Just updates the kernel components, not user-land stuff

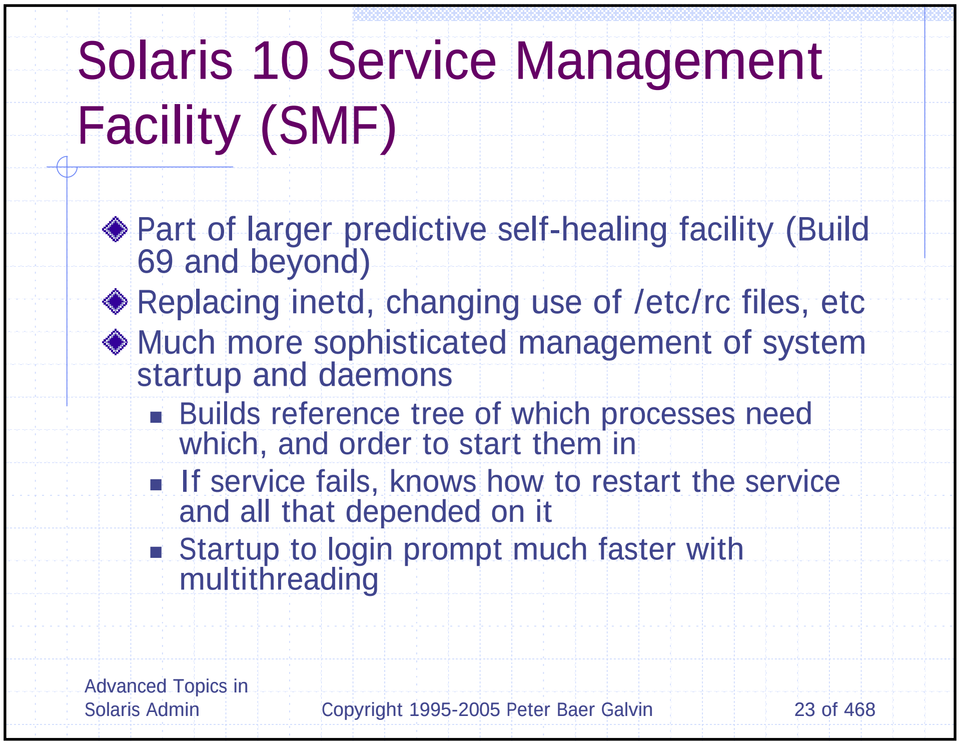


Blogs

- blogs.sun.com
  - bonwick
  - cantrill
  - moore
  - shapiro

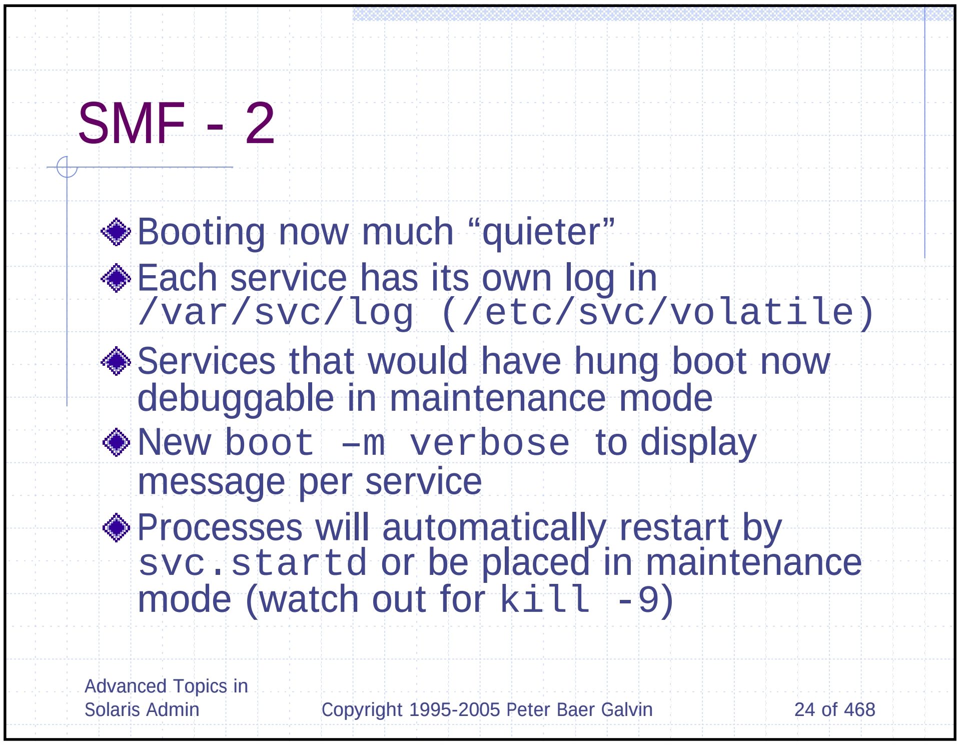


Reliability



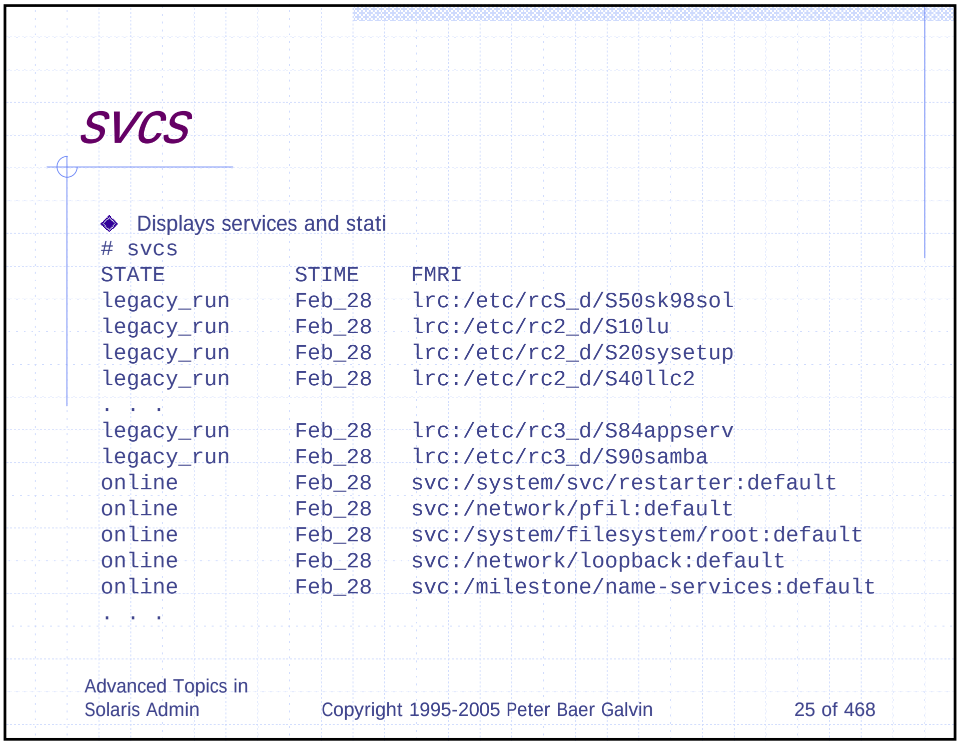
Solaris 10 Service Management Facility (SMF)

- Part of larger predictive self-healing facility (Build 69 and beyond)
- Replacing inetd, changing use of /etc/rc files, etc
- Much more sophisticated management of system startup and daemons
  - Builds reference tree of which processes need which, and order to start them in
  - If service fails, knows how to restart the service and all that depended on it
  - Startup to login prompt much faster with multithreading



SMF - 2

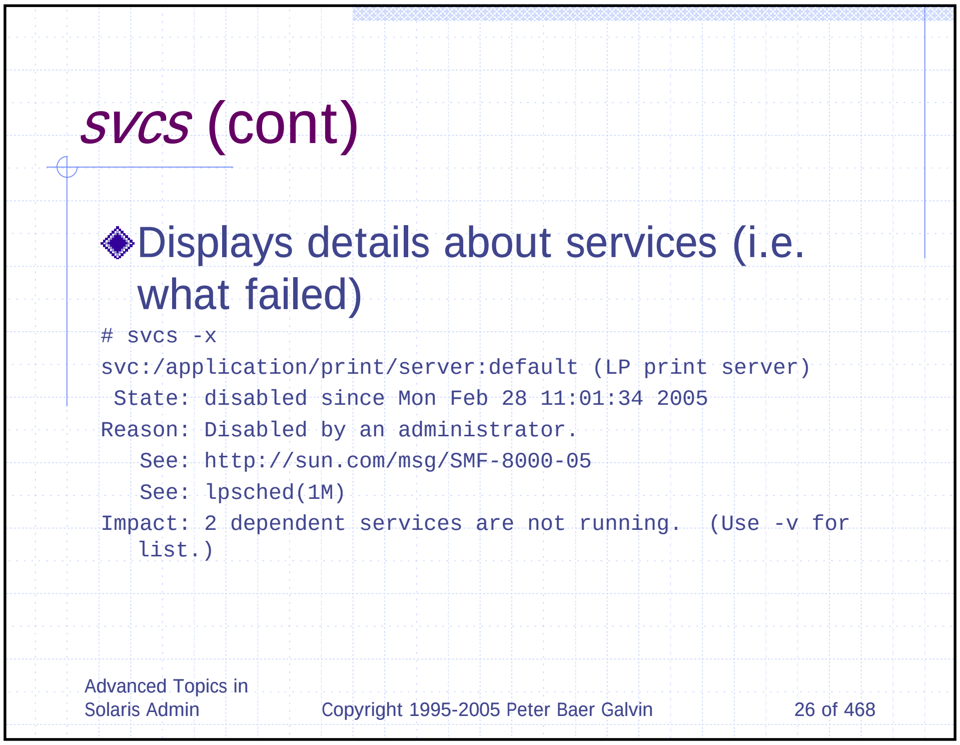
- Booting now much “quieter”
- Each service has its own log in `/var/svc/log (/etc/svc/volatile)`
- Services that would have hung boot now debuggable in maintenance mode
- New `boot -m verbose` to display message per service
- Processes will automatically restart by `svc.startd` or be placed in maintenance mode (watch out for `kill -9`)



svcs

- Displays services and stati

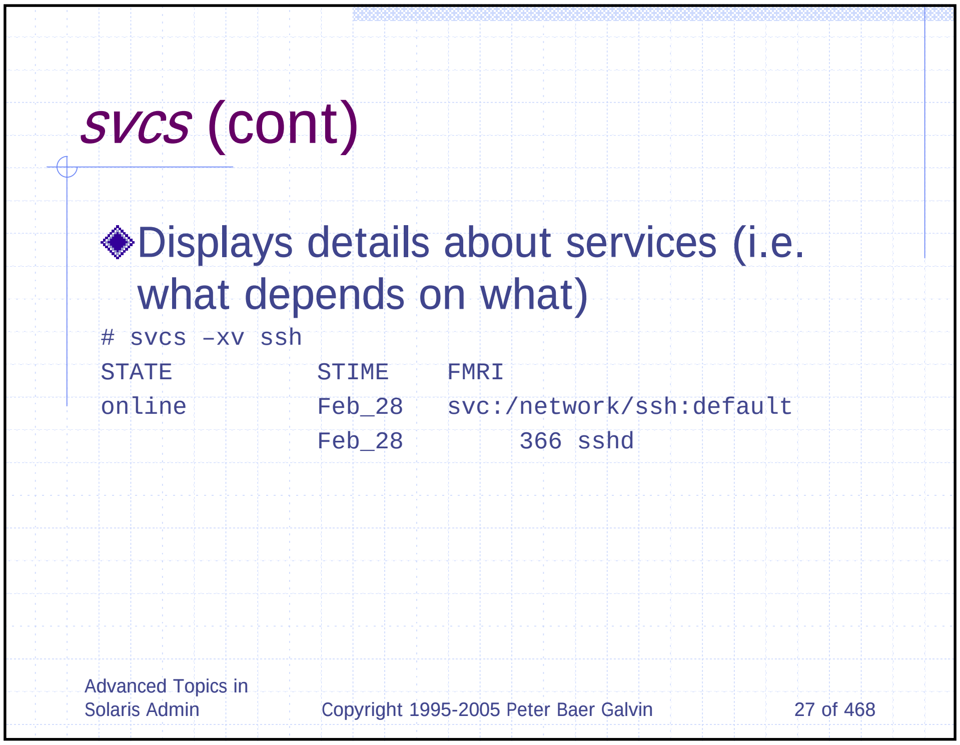
```
# svcs
STATE          STIME    FMRI
legacy_run     Feb_28   lrc:/etc/rcS_d/S50sk98sol
legacy_run     Feb_28   lrc:/etc/rc2_d/S10lu
legacy_run     Feb_28   lrc:/etc/rc2_d/S20sysetup
legacy_run     Feb_28   lrc:/etc/rc2_d/S40llc2
. . .
legacy_run     Feb_28   lrc:/etc/rc3_d/S84appserv
legacy_run     Feb_28   lrc:/etc/rc3_d/S90samba
online         Feb_28   svc:/system/svc/restarter:default
online         Feb_28   svc:/network/pfil:default
online         Feb_28   svc:/system/filesystem/root:default
online         Feb_28   svc:/network/loopback:default
online         Feb_28   svc:/milestone/name-services:default
. . .
```



svcs (cont)

- Displays details about services (i.e. what failed)

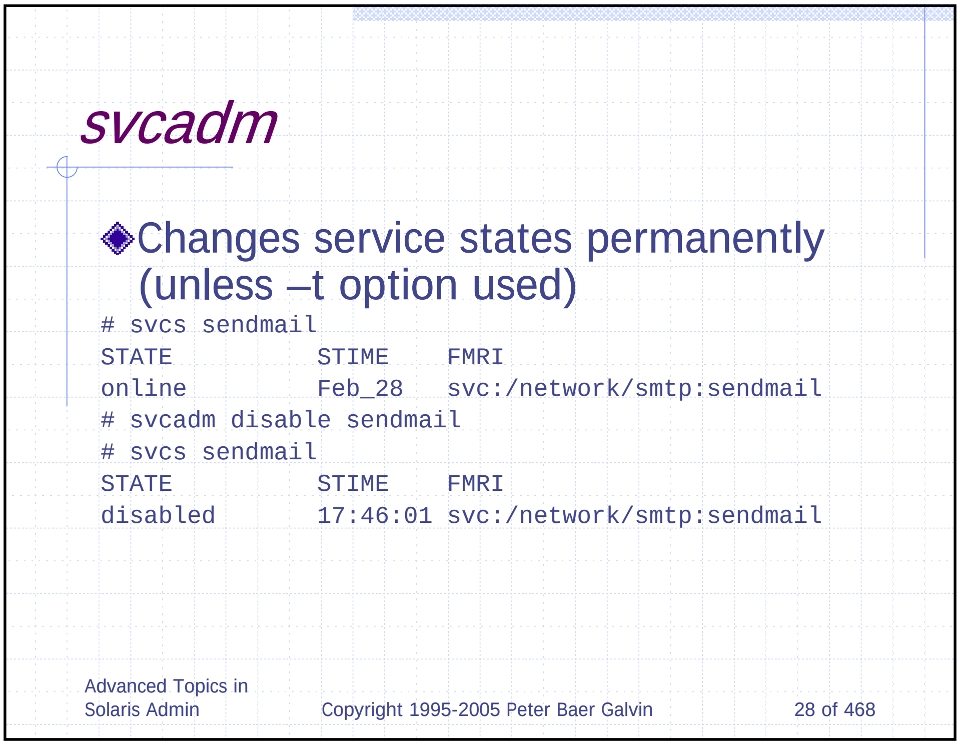
```
# svcs -x
svc:/application/print/server:default (LP print server)
 State: disabled since Mon Feb 28 11:01:34 2005
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: lpsched(1M)
Impact: 2 dependent services are not running.  (Use -v for
   list.)
```



svcs (cont)

- Displays details about services (i.e. what depends on what)

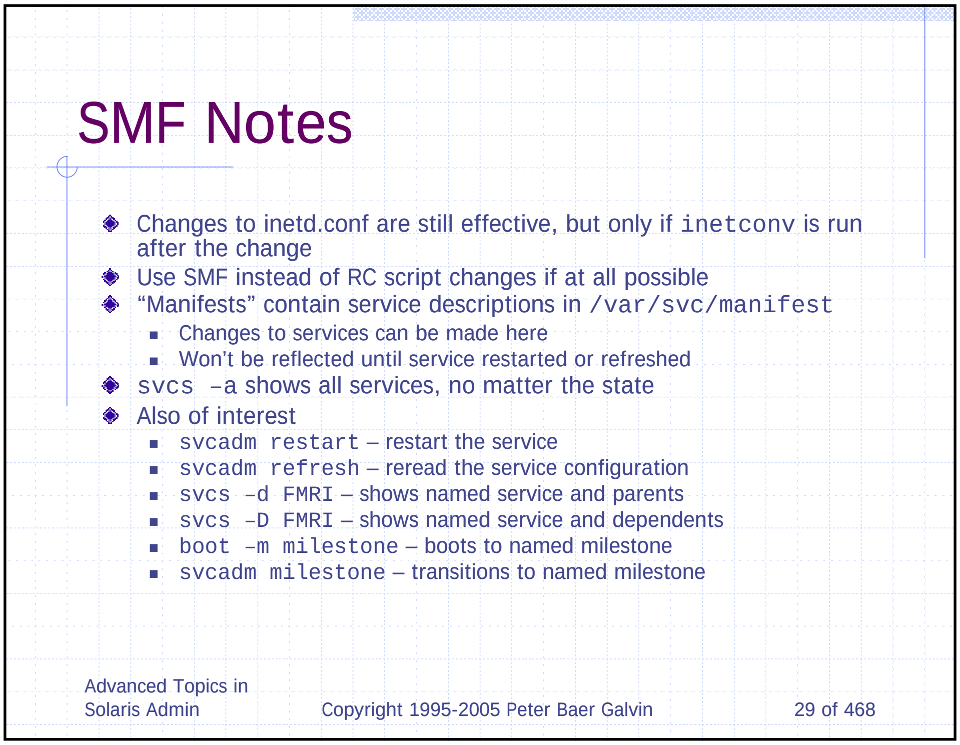
```
# svcs –xv ssh
STATE          STIME    FMRI
online         Feb_28   svc:/network/ssh:default
               Feb_28        366 sshd
```



svcadm

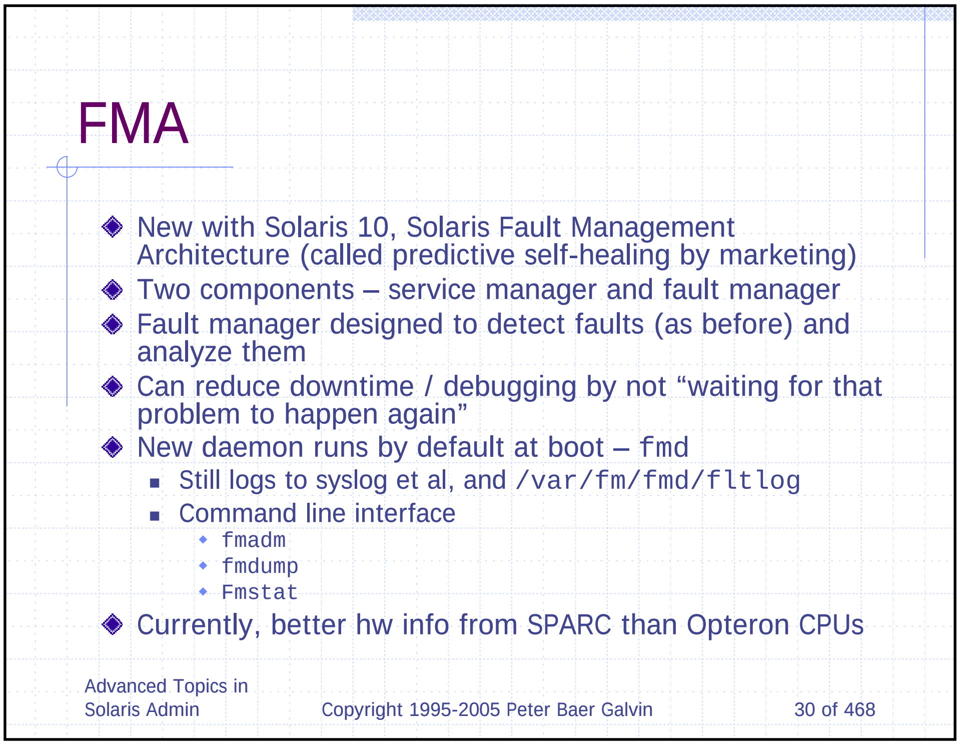
- Changes service states permanently (unless –t option used)

```
# svcs sendmail
STATE          STIME    FMRI
online         Feb_28   svc:/network/smtp:sendmail
# svcadm disable sendmail
# svcs sendmail
STATE          STIME    FMRI
disabled       17:46:01 svc:/network/smtp:sendmail
```



SMF Notes

- Changes to inetd.conf are still effective, but only if `inetconv` is run after the change
- Use SMF instead of RC script changes if at all possible
- “Manifests” contain service descriptions in `/var/svc/manifest`
  - Changes to services can be made here
  - Won’t be reflected until service restarted or refreshed
- `svcs –a` shows all services, no matter the state
- Also of interest
  - `svcadm restart` – restart the service
  - `svcadm refresh` – reread the service configuration
  - `svcs –d FMRI` – shows named service and parents
  - `svcs –D FMRI` – shows named service and dependents
  - `boot –m milestone` – boots to named milestone
  - `svcadm milestone` – transitions to named milestone



FMA

- New with Solaris 10, Solaris Fault Management Architecture (called predictive self-healing by marketing)
- Two components – service manager and fault manager
- Fault manager designed to detect faults (as before) and analyze them
- Can reduce downtime / debugging by not “waiting for that problem to happen again”
- New daemon runs by default at boot – `fmd`
  - Still logs to syslog et al, and `/var/fm/fmd/fltlog`
  - Command line interface
    - `fmadm`
    - `fmdump`
    - `Fmstat`
- Currently, better hw info from SPARC than Opteron CPUs

FMA Fault Management

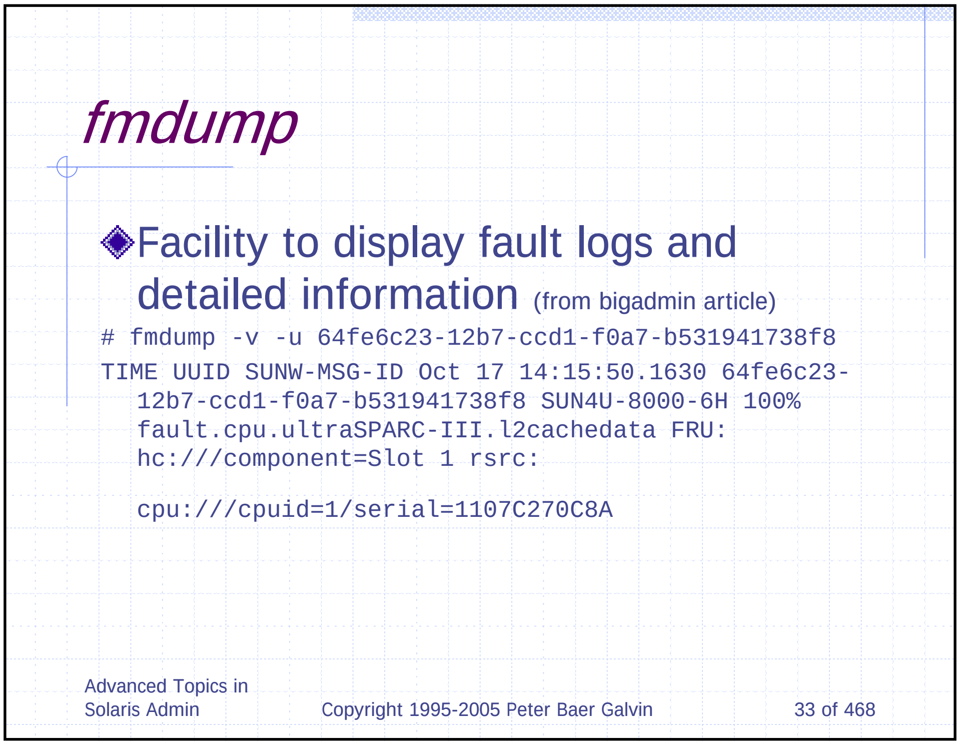
- Should be much more likely to catch and debug intermittent or correctable error and point to a correction: (from bigadmin article)

```
SUNW-MSG-ID: SUN4U-8000-6H, TYPE: Fault, VER: 1,
   SEVERITY: Major EVENT-TIME: Sun Oct 17 14:15:50 PDT
   2004 PLATFORM: SUNW,Sun-Blade-1000, CSN: -,
   HOSTNAME: myhost EVENT-ID: 64fe6c23-12b7-ccd1-f0a7-
   b531941738f8 DESC: The number of errors associated
   with this CPU has exceeded acceptable levels. Refer
   to http://sun.com/msg/SUN4U-8000-6H for more
   information. AUTO-RESPONSE: An attempt will be made
   to remove the affected CPU from service. IMPACT:
   Performance of this system may be affected. REC-
   ACTION: Schedule a repair procedure to replace the
   affected CPU. Use fmdump -v -u <EVENT_ID> to
   identify the CPU.
```

fmadm

- Main administrative interface

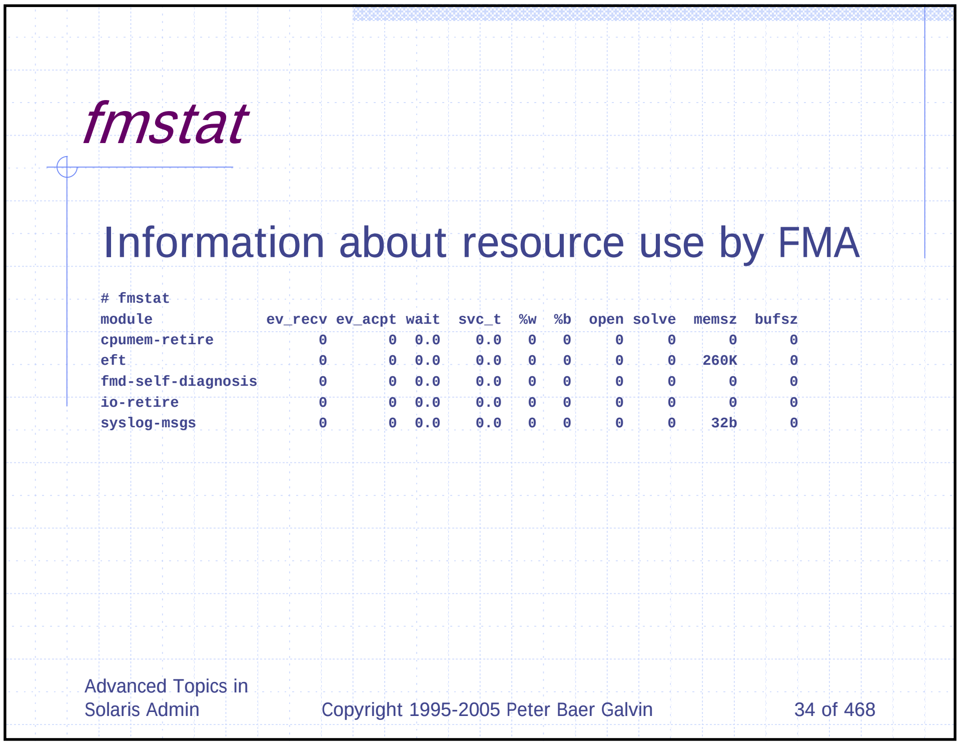
```
# fmadm
Usage: fmadm [-P prog] [-q] [cmd [args ... ]]

    fmadm config                  - display fault manager configuration
    fmadm faulty [-ai]            - display list of faulty resources
    fmadm flush <fmri> ...        - flush cached state for resource
    fmadm load <path>             - load specified fault manager module
    fmadm repair <fmri>|<uuid>    - record repair to resource(s)
    fmadm reset [-s serd] <module> - reset module or sub-component
    fmadm rotate <logname>        - rotate log file
    fmadm unload <module>         - unload specified fault manager module
# fmadm config
MODULE                   VERSION STATUS  DESCRIPTION
cpumem-retire            1.0     active  CPU/Memory Retire Agent
eft                      1.12    active  eft diagnosis engine
fmd-self-diagnosis       1.0     active  Fault Manager Self-Diagnosis
io-retire                1.0     active  I/O Retire Agent
syslog-msgs              1.0     active  Syslog Messaging Agent
```



fmdump

- Facility to display fault logs and detailed information (from bigadmin article)

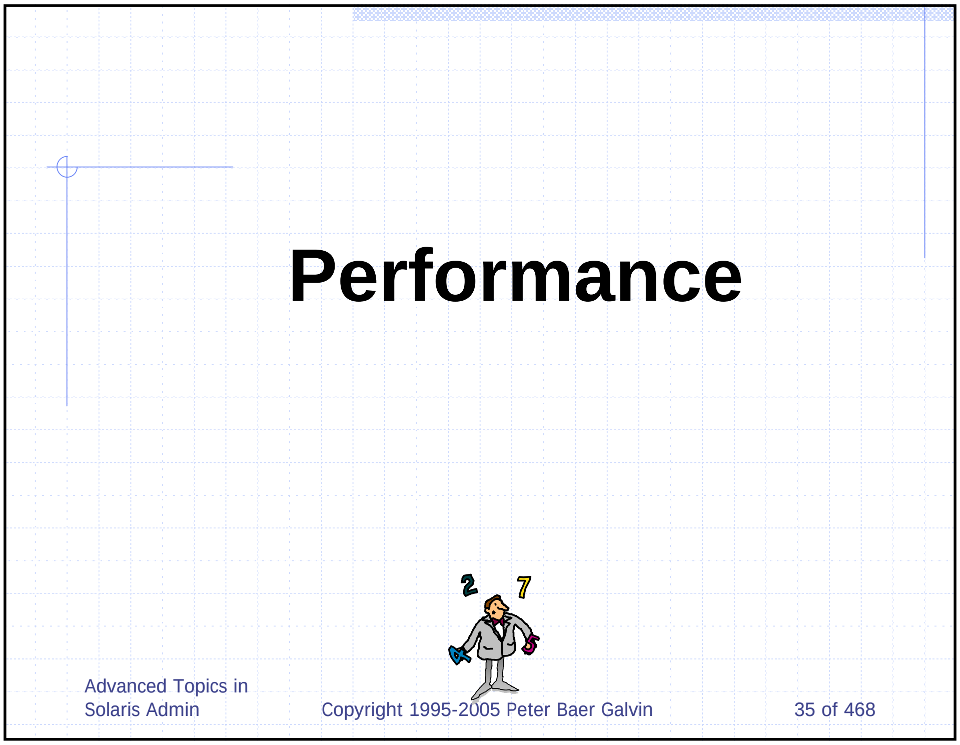
```
# fmdump -v -u 64fe6c23-12b7-ccd1-f0a7-b531941738f8
TIME UUID SUNW-MSG-ID Oct 17 14:15:50.1630 64fe6c23-
   12b7-ccd1-f0a7-b531941738f8 SUN4U-8000-6H 100%
   fault.cpu.ultraSPARC-III.l2cachedata FRU:
   hc:///component=Slot 1 rsrc:

   cpu:///cpuid=1/serial=1107C270C8A
```



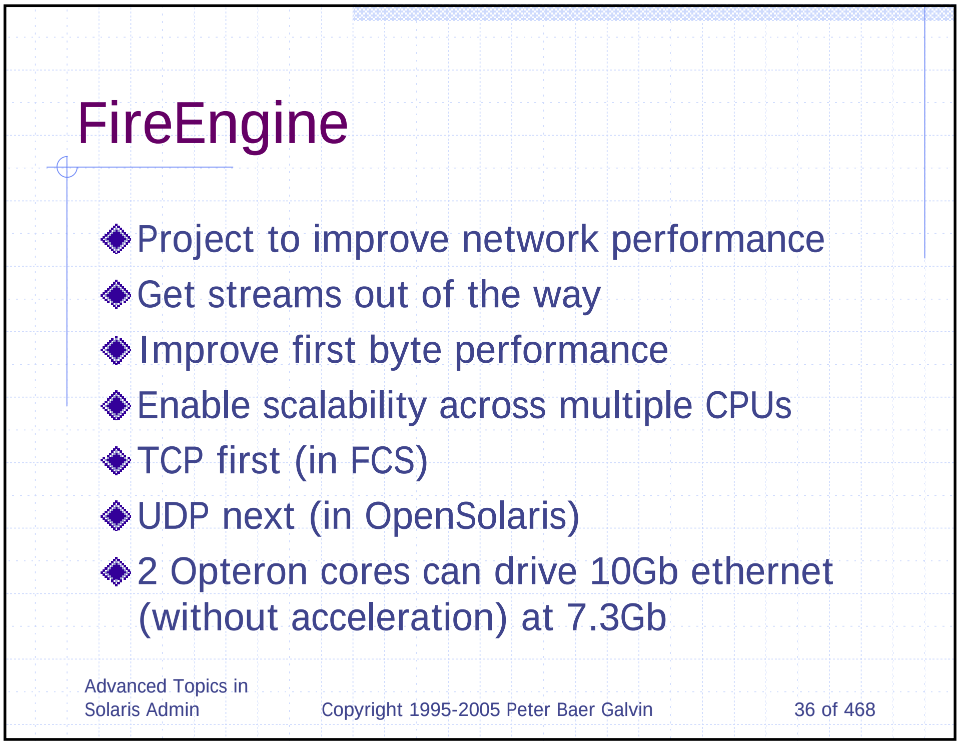
fmstat

Information about resource use by FMA

```
# fmstat
module               ev_recv ev_acpt wait  svc_t  %w  %b  open solve  memsz  bufsz
cpumem-retire              0       0  0.0    0.0   0   0     0     0      0      0
eft                        0       0  0.0    0.0   0   0     0     0   260K      0
fmd-self-diagnosis         0       0  0.0    0.0   0   0     0     0      0      0
io-retire                  0       0  0.0    0.0   0   0     0     0      0      0
syslog-msgs                0       0  0.0    0.0   0   0     0     0    32b      0
```

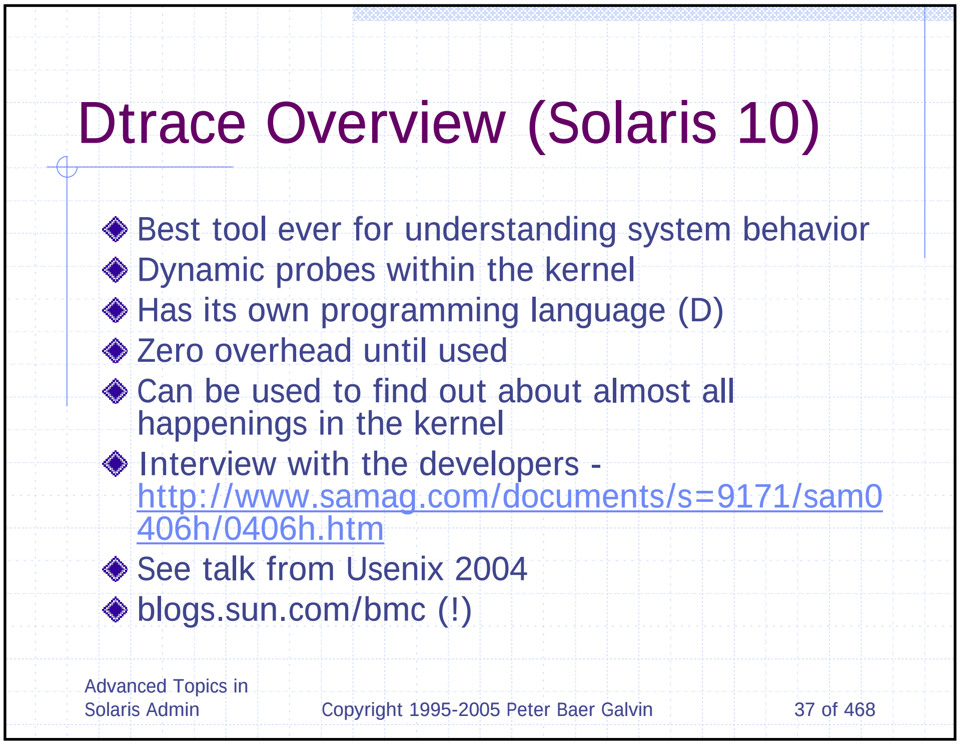


Performance



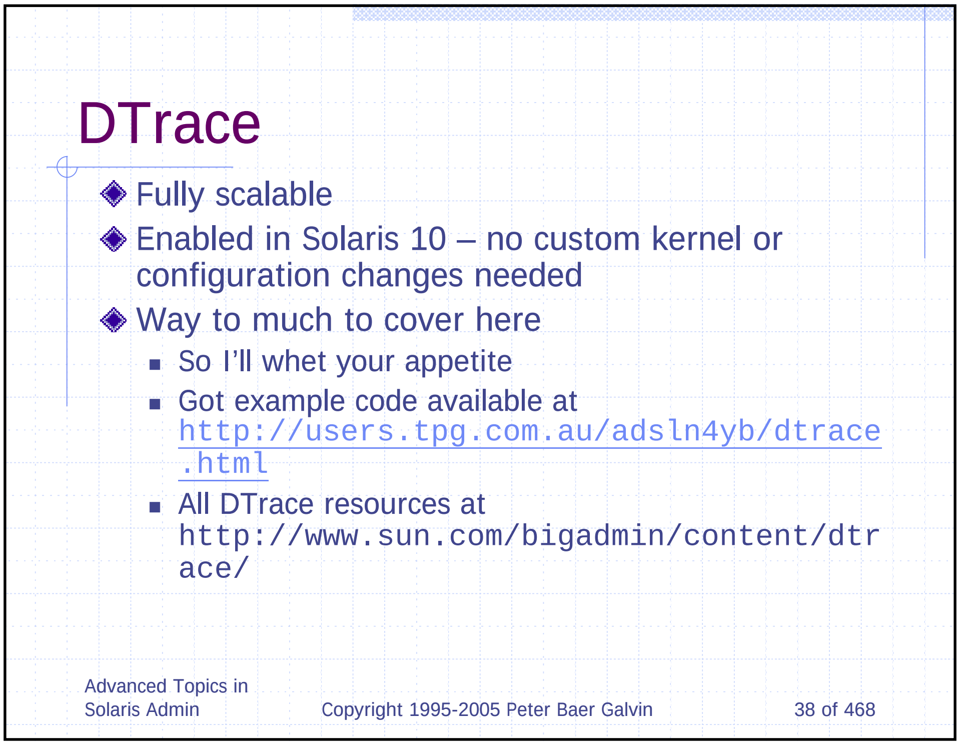
FireEngine

- Project to improve network performance
- Get streams out of the way
- Improve first byte performance
- Enable scalability across multiple CPUs
- TCP first (in FCS)
- UDP next (in OpenSolaris)
- 2 Opteron cores can drive 10Gb ethernet (without acceleration) at 7.3Gb



Dtrace Overview (Solaris 10)

- Best tool ever for understanding system behavior
- Dynamic probes within the kernel
- Has its own programming language (D)
- Zero overhead until used
- Can be used to find out about almost all happenings in the kernel
- Interview with the developers - http://www.samag.com/documents/s=9171/sam0406h/0406h.htm
- See talk from Usenix 2004
- blogs.sun.com/bmc (!)



DTrace

- Fully scalable
- Enabled in Solaris 10 – no custom kernel or configuration changes needed
- Way to much to cover here
  - So I'll whet your appetite
  - Got example code available at `http://users.tpg.com.au/adsln4yb/dtrace.html`
  - All DTrace resources at `http://www.sun.com/bigadmin/content/dtrace/`

DTrace Example - 1

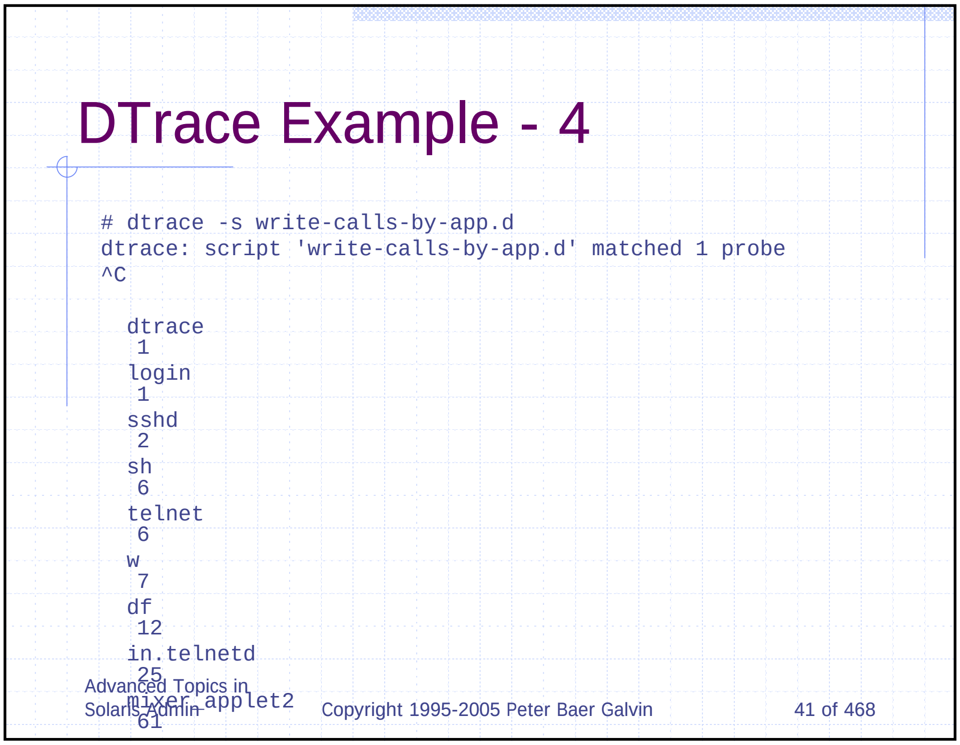
- **connections.d** snoop inbound TCP connections as they are established, displaying the server process that accepted the connection.

```
# ./connections.d
UID PID IP_SOURCE PORT CMD
0 254 192.168.001.001 23 /usr/sbin/inetd -s
0 254 192.168.001.001 23 /usr/sbin/inetd -s
0 254 192.168.001.001 79 /usr/sbin/inetd -s
0 254 192.168.001.001 21 /usr/sbin/inetd -s
0 254 192.168.001.001 79 /usr/sbin/inetd -s
100 2319 192.168.001.001 6000 /usr/openwin/bin/Xsun :0 -
    nobanner
0 254 192.168.001.001 79 /usr/sbin/inetd -s [...]
```

DTrace Example - 2

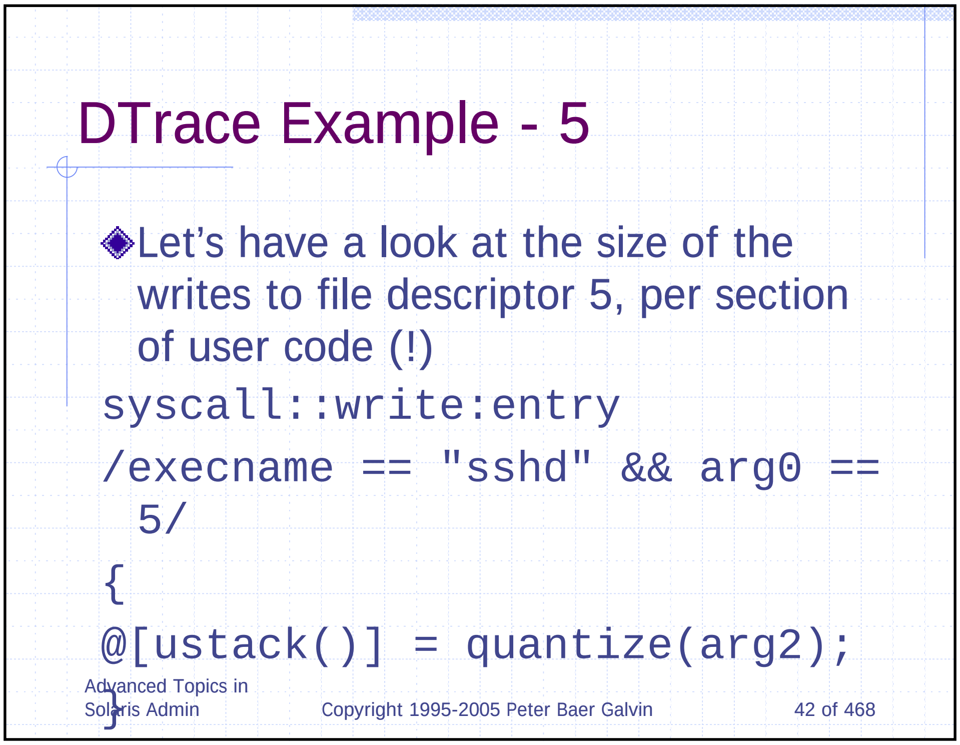
- The following script counts number of write(2) calls by application:

```
syscall::write:entry
{
@counts[execname] = count();
}
```



DTrace Example - 4

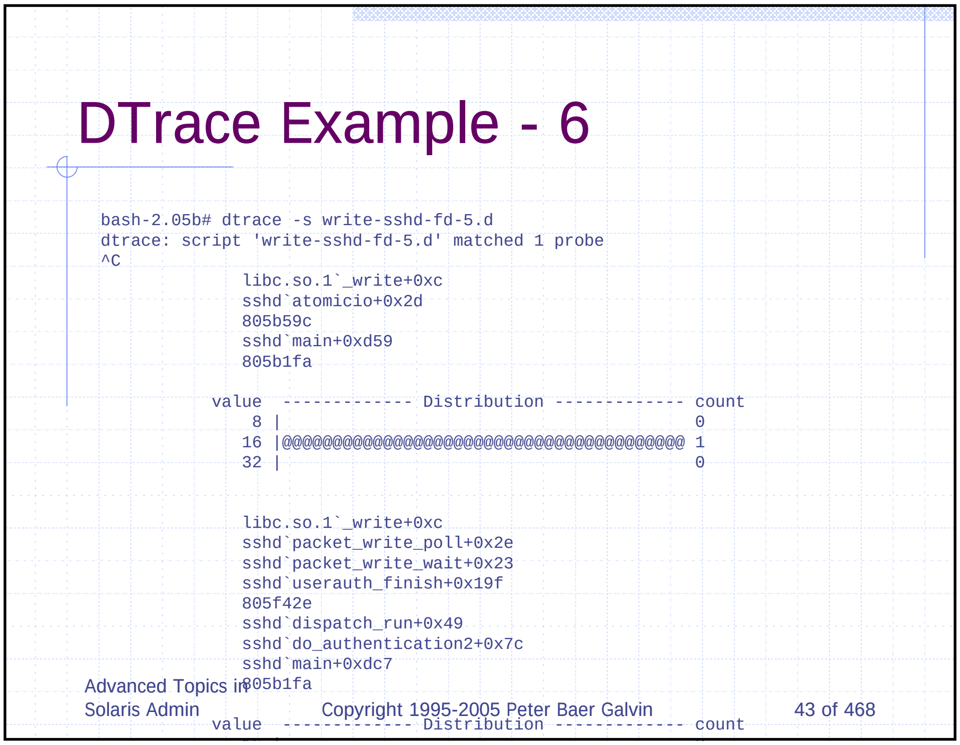
```
# dtrace -s write-calls-by-app.d
dtrace: script 'write-calls-by-app.d' matched 1 probe
^C

  dtrace
   1
  login
   1
  sshd
   2
  sh
   6
  telnet
   6
  w
   7
  df
   12
  in.telnetd
   25
  mixer_applet2
   61
```



DTrace Example - 5

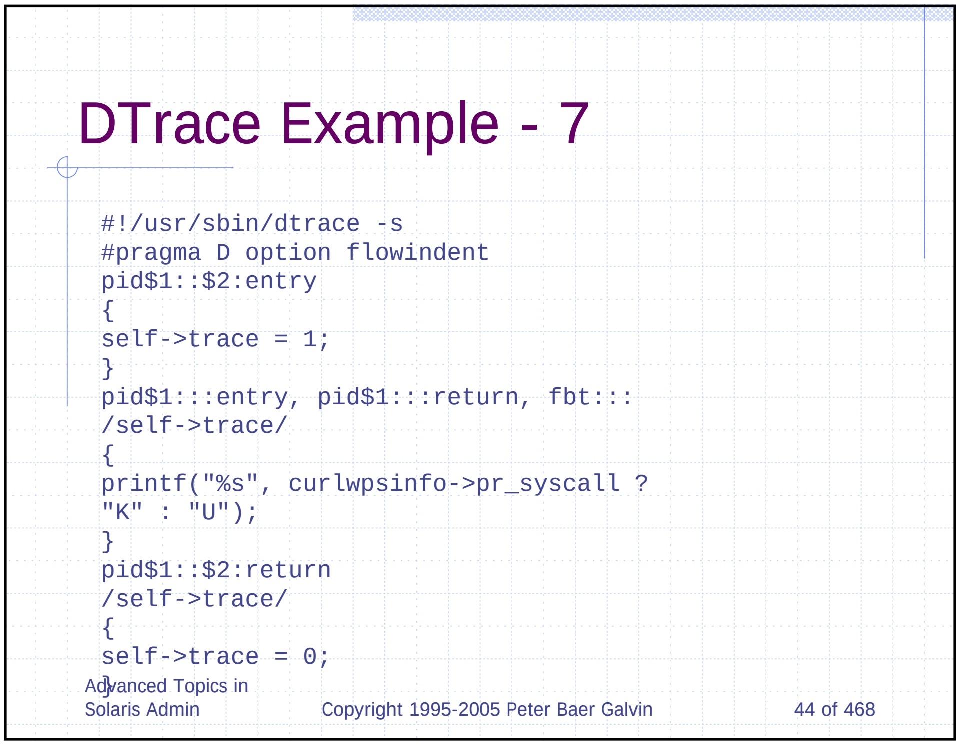
- Let’s have a look at the size of the writes to file descriptor 5, per section of user code (!)

```
syscall::write:entry
/execname == "sshd" && arg0 ==
  5/
{
@[ustack()] = quantize(arg2);
}
```



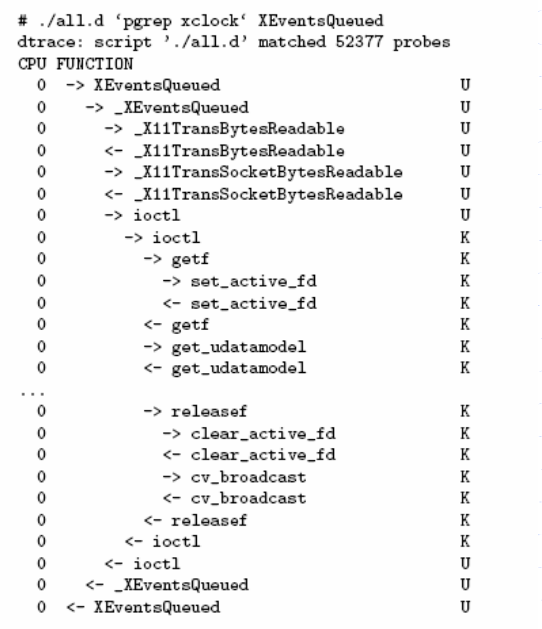
DTrace Example - 6

```
bash-2.05b# dtrace -s write-sshd-fd-5.d
dtrace: script 'write-sshd-fd-5.d' matched 1 probe
^C
              libc.so.1`_write+0xc
              sshd`atomicio+0x2d
              805b59c
              sshd`main+0xd59
              805b1fa

           value  ------------- Distribution ------------- count
               8 |                                         0
              16 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 1
              32 |                                         0


              libc.so.1`_write+0xc
              sshd`packet_write_poll+0x2e
              sshd`packet_write_wait+0x23
              sshd`userauth_finish+0x19f
              805f42e
              sshd`dispatch_run+0x49
              sshd`do_authentication2+0x7c
              sshd`main+0xdc7
              805b1fa

           value  ------------- Distribution ------------- count
```



DTrace Example - 7

```
#!/usr/sbin/dtrace -s
#pragma D option flowindent
pid$1::$2:entry
{
self->trace = 1;
}
pid$1:::entry, pid$1:::return, fbt:::
/self->trace/
{
printf("%s", curlwpsinfo->pr_syscall ?
"K" : "U");
}
pid$1::$2:return
/self->trace/
{
self->trace = 0;
}
```



```
# ./all.d ‘pgrep xclock‘ XEventsQueued
dtrace: script ’./all.d’ matched 52377 probes
CPU FUNCTION
  0  -> XEventsQueued                      U
  0    -> _XEventsQueued                   U
  0      -> _X11TransBytesReadable         U
  0      <- _X11TransBytesReadable         U
  0      -> _X11TransSocketBytesReadable   U
  0      <- _X11TransSocketBytesReadable   U
  0      -> ioctl                          U
  0        -> ioctl                        K
  0          -> getf                       K
  0            -> set_active_fd            K
  0            <- set_active_fd            K
  0          <- getf                       K
  0          -> get_udatamodel             K
  0          <- get_udatamodel             K
...
  0          -> releasef                   K
  0            -> clear_active_fd          K
  0            <- clear_active_fd          K
  0            -> cv_broadcast             K
  0            <- cv_broadcast             K
  0          <- releasef                   K
  0        <- ioctl                        K
  0      <- ioctl                          U
  0    <- _XEventsQueued                   U
  0  <- XEventsQueued                      U
```

DTrace One Liners

- # New processes with arguments,
  dtrace -n 'proc:::exec-success { trace(curpsinfo->pr_psargs); }'
- # Files opened by process,
  dtrace -n 'syscall::open*:entry { printf("%s %s",execname,copyinstr(arg0)); }'
- # Syscall count by program,
  dtrace -n 'syscall:::entry { @num[execname] = count(); }'
- # Syscall count by syscall,
  dtrace -n 'syscall:::entry { @num[probefunc] = count(); }'
- # Syscall count by process,
  dtrace -n 'syscall:::entry { @num[pid,execname] = count(); }'
- # Read bytes by process,
  dtrace -n 'sysinfo:::readch { @bytes[execname] = sum(arg0); }'
- # Write bytes by process,
  dtrace -n 'sysinfo:::writech { @bytes[execname] = sum(arg0); }'
- # Read size distribution by process,
  dtrace -n 'sysinfo:::readch { @dist[execname] = quantize(arg0); }'
- # Write size distribution by process,
  dtrace -n 'sysinfo:::writech { @dist[execname] = quantize(arg0); }'
- # Disk size by process,
  dtrace -n 'io:::start { printf("%d %s %d",pid,execname,args[0]->b_bcount); }'
- # Pages paged in by process,
  dtrace -n 'vminfo:::pgpgin { @pg[execname] = sum(arg0); }'
- # Minor faults by process,
  dtrace -n 'vminfo:::as_fault { @mem[execname] = sum(arg0); }'

Security

Why Me?

Warning about Security Work

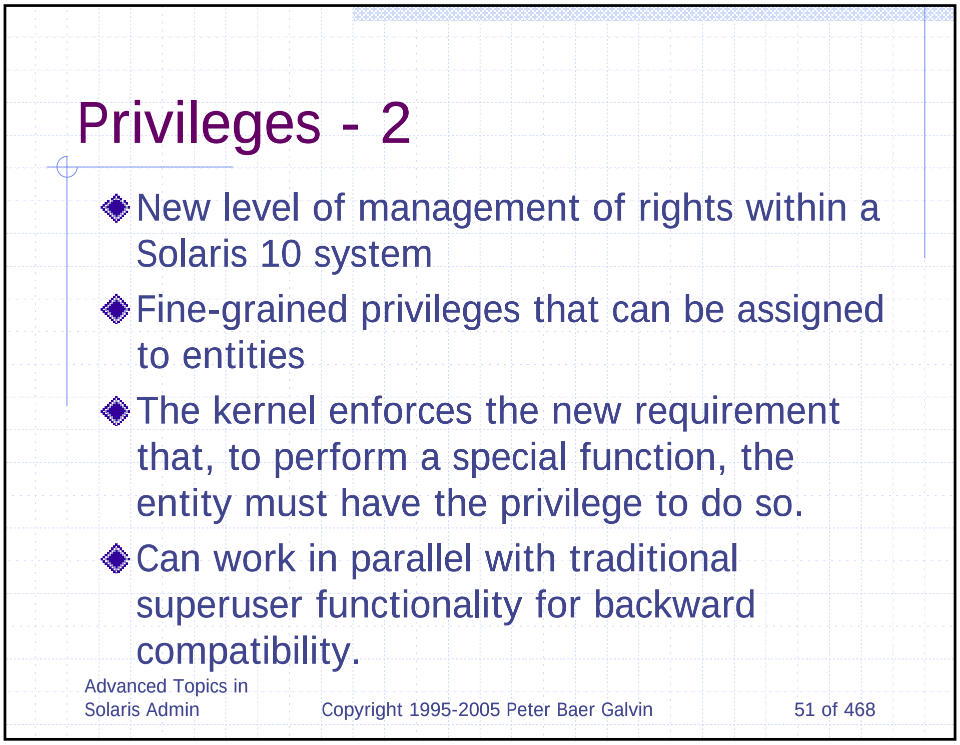
- Be sure to get written permission before performing any security testing
  - Bad things can happen if you don’t
    - **State of Oregon v. Randal Schwartz**
      - http://www.lightlink.com/spacenka/fors

Role-based Administration

- Doles out administrative privs without having to give full root privs
- New to Solaris 8, from Trusted Solaris
- Implemented via psh, pksh, ptcsh
- Like sudo, but built into shells
- Implements rule sets, roles limited to those rule sets
- Logging seems to be limited
- Improvements included in S9, S10
  - To make it actually usable

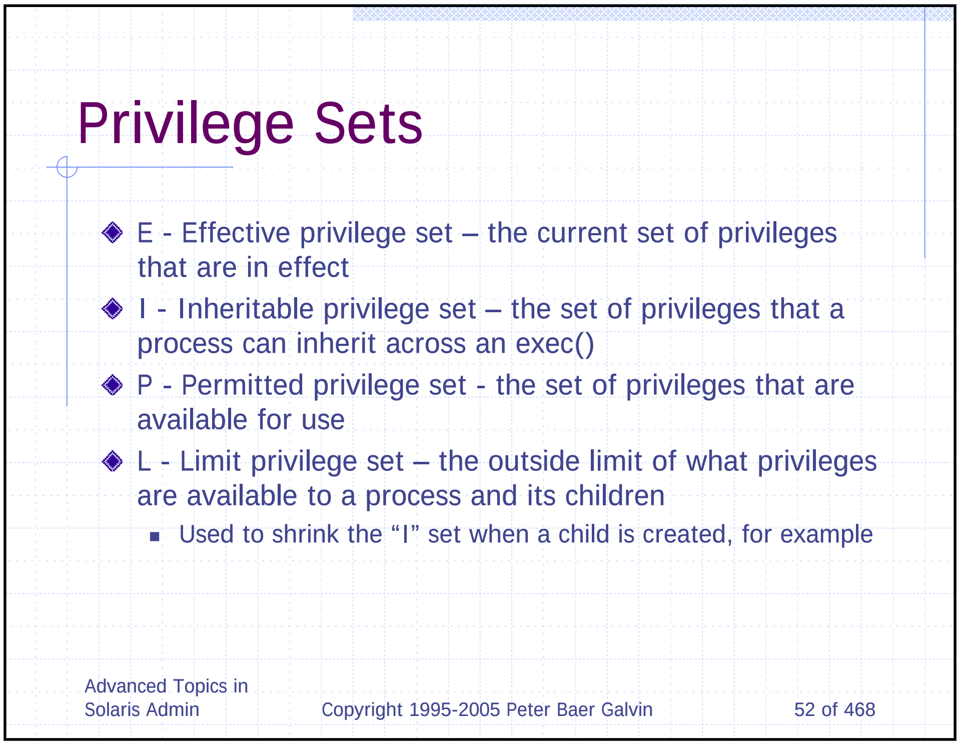
Privileges (s10)

- Really known as “least privilege”
  - Only the minimum privileges to get a job done should be available
- Alternative to being root or no one
- Done at the API level
  - SetUID programs can dictate fine grain access to kernel features
  - Can limit what privs children have
  - Should further help can buffer overflows and other privilege escalation methods
- Done at the user or role level
  - All specific users to perform specific operations regardless of the programs being run



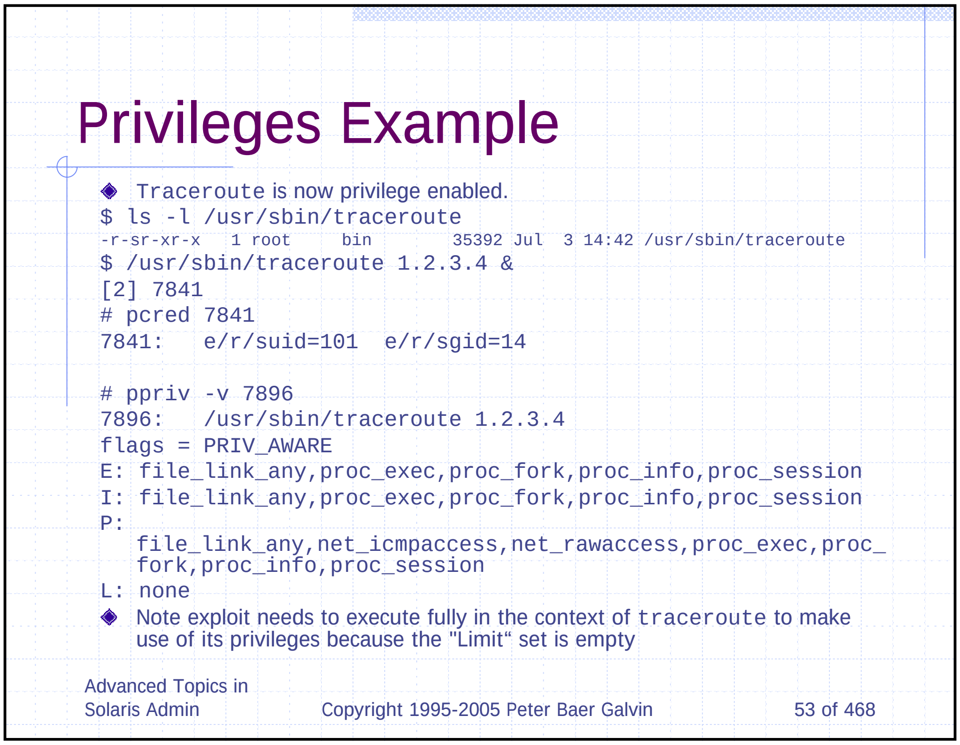
Privileges - 2

- New level of management of rights within a Solaris 10 system
- Fine-grained privileges that can be assigned to entities
- The kernel enforces the new requirement that, to perform a special function, the entity must have the privilege to do so.
- Can work in parallel with traditional superuser functionality for backward compatibility.



Privilege Sets

- E - Effective privilege set – the current set of privileges that are in effect
- I - Inheritable privilege set – the set of privileges that a process can inherit across an exec()
- P - Permitted privilege set - the set of privileges that are available for use
- L - Limit privilege set – the outside limit of what privileges are available to a process and its children
  - Used to shrink the "I" set when a child is created, for example

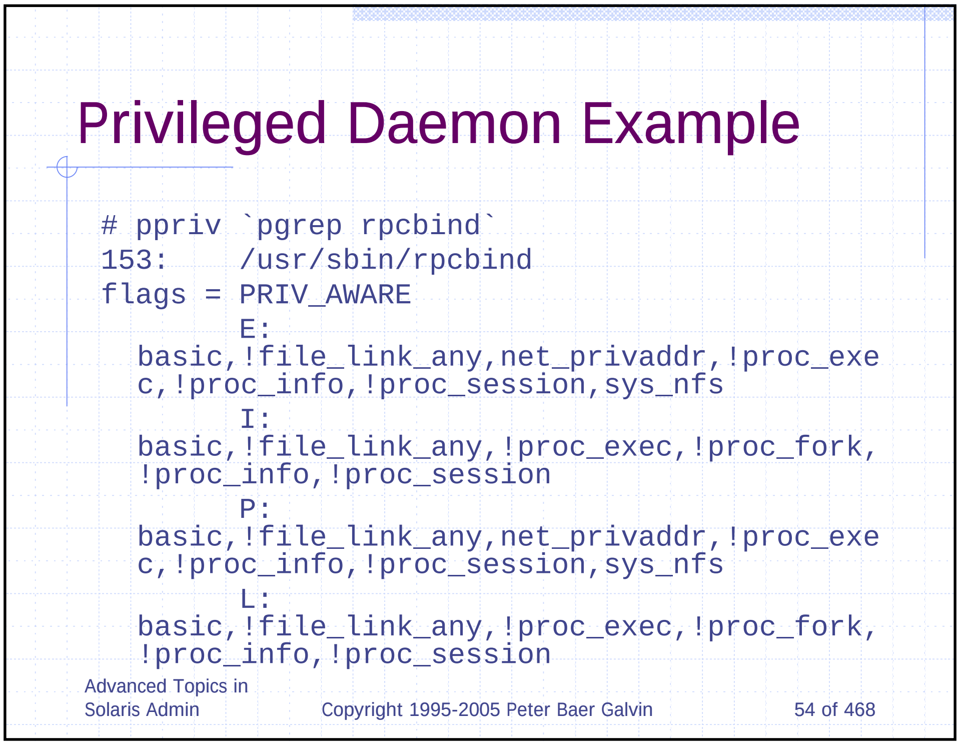


Privileges Example

- Traceroute is now privilege enabled.

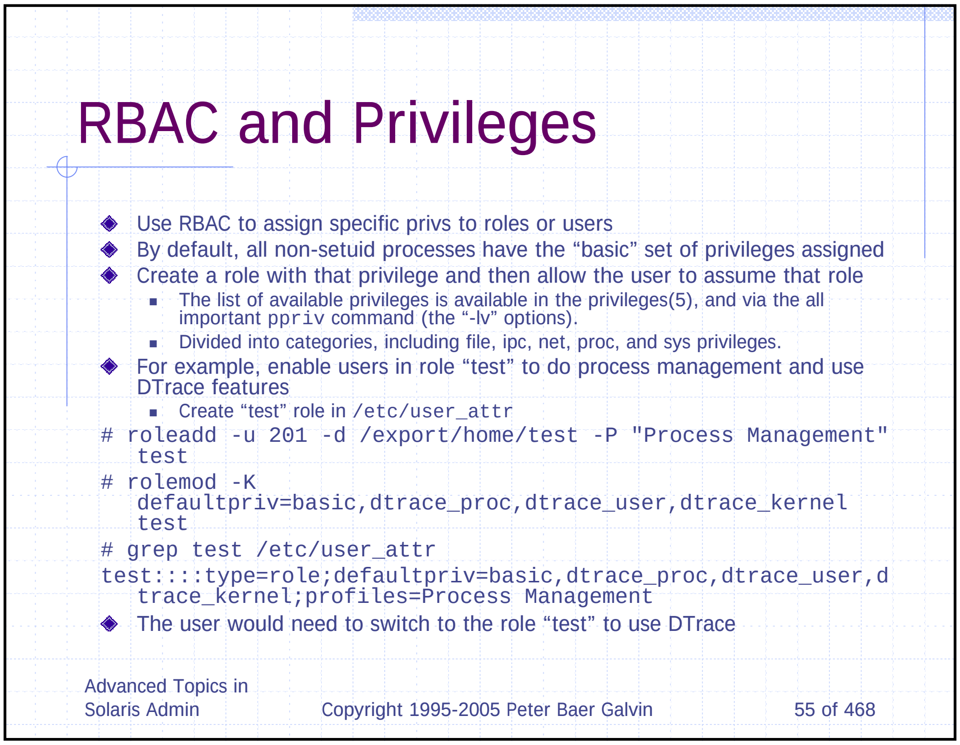
```
$ ls -l /usr/sbin/traceroute
-r-sr-xr-x   1 root     bin        35392 Jul  3 14:42 /usr/sbin/traceroute
$ /usr/sbin/traceroute 1.2.3.4 &
[2] 7841
# pcred 7841
7841:   e/r/suid=101  e/r/sgid=14

# ppriv -v 7896
7896:   /usr/sbin/traceroute 1.2.3.4
flags = PRIV_AWARE
E: file_link_any,proc_exec,proc_fork,proc_info,proc_session
I: file_link_any,proc_exec,proc_fork,proc_info,proc_session
P:
   file_link_any,net_icmpaccess,net_rawaccess,proc_exec,proc_
   fork,proc_info,proc_session
L: none
```

- Note exploit needs to execute fully in the context of `traceroute` to make use of its privileges because the "Limit“ set is empty



Privileged Daemon Example

```
# ppriv `pgrep rpcbind`
153:    /usr/sbin/rpcbind
flags = PRIV_AWARE
        E:
   basic,!file_link_any,net_privaddr,!proc_exe
   c,!proc_info,!proc_session,sys_nfs
        I:
   basic,!file_link_any,!proc_exec,!proc_fork,
   !proc_info,!proc_session
        P:
   basic,!file_link_any,net_privaddr,!proc_exe
   c,!proc_info,!proc_session,sys_nfs
        L:
   basic,!file_link_any,!proc_exec,!proc_fork,
   !proc_info,!proc_session
```

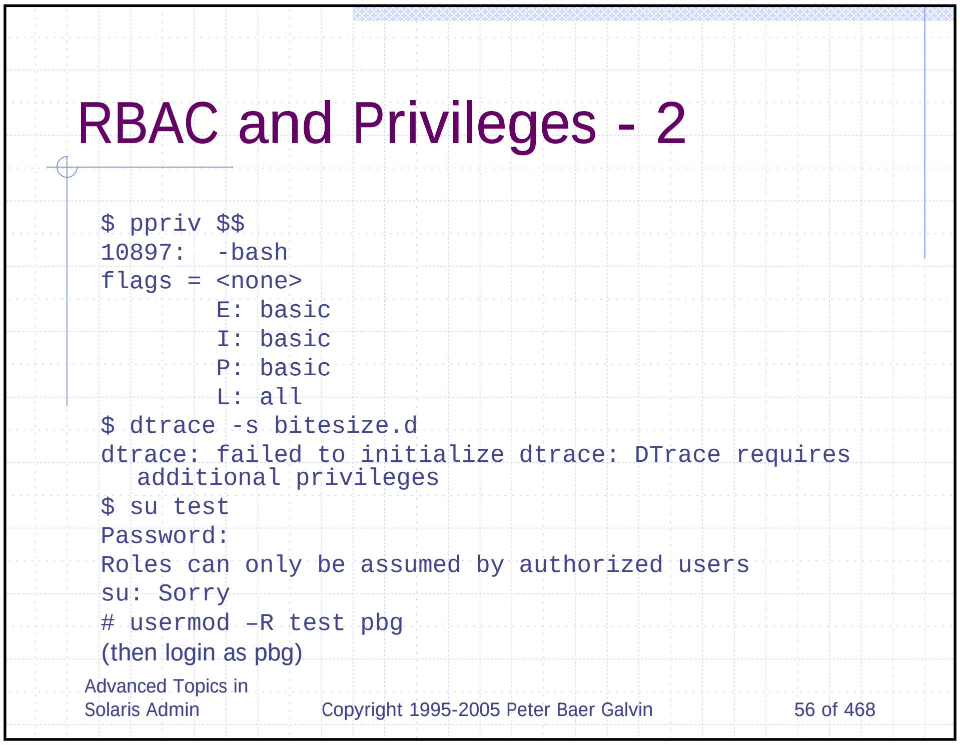


RBAC and Privileges

- Use RBAC to assign specific privs to roles or users
- By default, all non-setuid processes have the “basic” set of privileges assigned
- Create a role with that privilege and then allow the user to assume that role
  - The list of available privileges is available in the privileges(5), and via the all important `ppriv` command (the “-lv” options).
  - Divided into categories, including file, ipc, net, proc, and sys privileges.
- For example, enable users in role “test” to do process management and use DTrace features
  - Create “test” role in `/etc/user_attr`

```
# roleadd -u 201 -d /export/home/test -P "Process Management"
   test
# rolemod -K
   defaultpriv=basic,dtrace_proc,dtrace_user,dtrace_kernel
   test
# grep test /etc/user_attr
test::::type=role;defaultpriv=basic,dtrace_proc,dtrace_user,d
   trace_kernel;profiles=Process Management
```

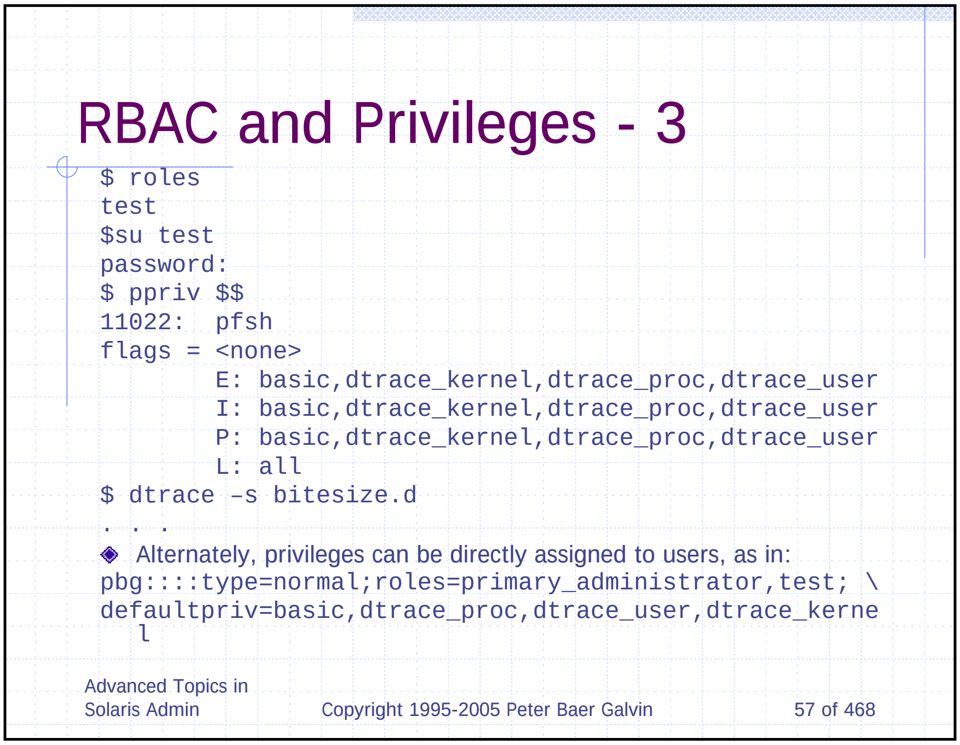
- The user would need to switch to the role “test” to use DTrace



RBAC and Privileges - 2

```
$ ppriv $$
10897:  -bash
flags = <none>
        E: basic
        I: basic
        P: basic
        L: all
$ dtrace -s bitesize.d
dtrace: failed to initialize dtrace: DTrace requires
   additional privileges
$ su test
Password:
Roles can only be assumed by authorized users
su: Sorry
# usermod –R test pbg
```

(then login as pbg)

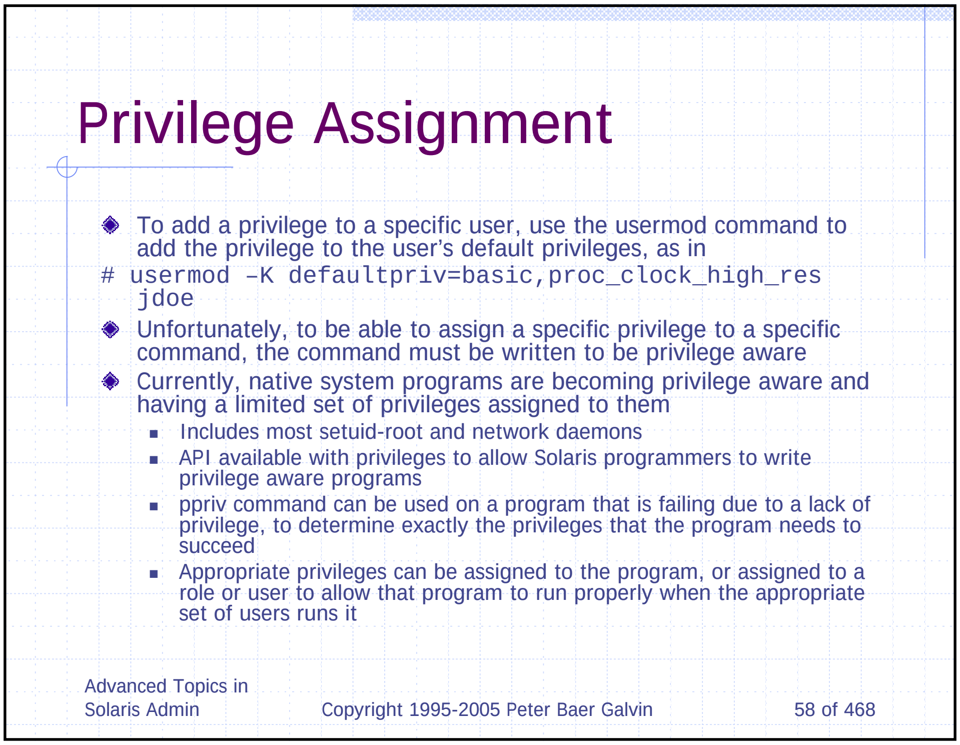


RBAC and Privileges - 3

```
$ roles
test
$su test
password:
$ ppriv $$
11022:  pfsh
flags = <none>
        E: basic,dtrace_kernel,dtrace_proc,dtrace_user
        I: basic,dtrace_kernel,dtrace_proc,dtrace_user
        P: basic,dtrace_kernel,dtrace_proc,dtrace_user
        L: all
$ dtrace –s bitesize.d
. . .
```

- Alternately, privileges can be directly assigned to users, as in:

```
pbg::::type=normal;roles=primary_administrator,test; \
defaultpriv=basic,dtrace_proc,dtrace_user,dtrace_kerne
   l
```



Privilege Assignment

- To add a privilege to a specific user, use the usermod command to add the privilege to the user's default privileges, as in

```
# usermod –K defaultpriv=basic,proc_clock_high_res
  jdoe
```

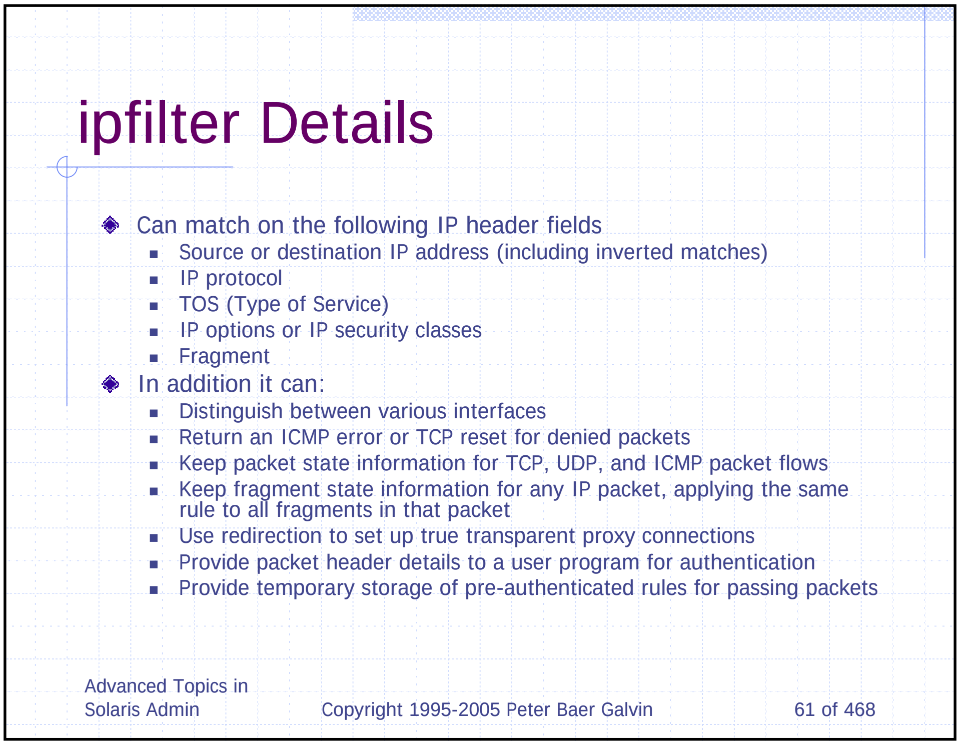
- Unfortunately, to be able to assign a specific privilege to a specific command, the command must be written to be privilege aware
- Currently, native system programs are becoming privilege aware and having a limited set of privileges assigned to them
  - Includes most setuid-root and network daemons
  - API available with privileges to allow Solaris programmers to write privilege aware programs
  - ppriv command can be used on a program that is failing due to a lack of privilege, to determine exactly the privileges that the program needs to succeed
  - Appropriate privileges can be assigned to the program, or assigned to a role or user to allow that program to run properly when the appropriate set of users runs it

Packet Filtering Overview (S10)

- Solaris used to have nothing, then SunScreen was commercial, then SunScreen was included, now ipfilter is standard
- Solaris IP Filter is a host-based firewall that is derived from the open source IP Filter code, developed and maintained by Darren Reed
  - Based on version 4.0.33 of the open source IP Filter
  - Uses the STREAMS module, pfil, to intercept packets
  - By default, pfil is not autopushed onto network interface cards (NICs). Autopush of pfil is disabled for all drivers

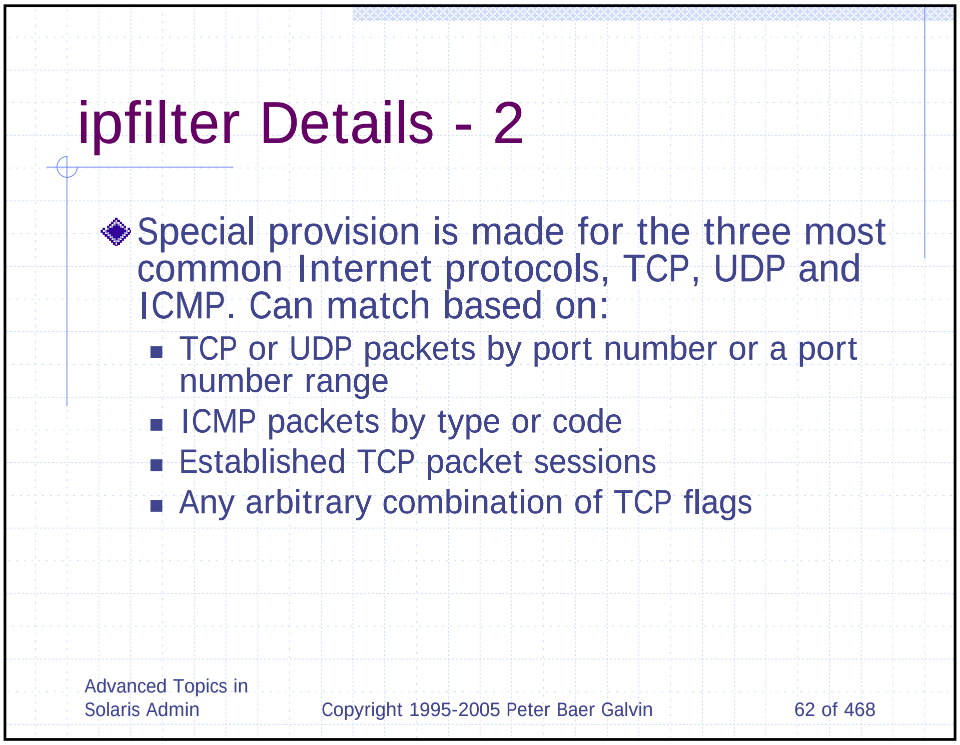
Packet Filtering Overview - 2

- Provides packet filtering and network address translation (NAT), based upon a user-configurable policy
  - Rules are configurable to filter either statefully or statelessly
  - Command line interface only
    - ipf for loading or clearing packet filter rules
    - ipnat for loading or clearing NAT rules
    - ippool for managing address pools associated with IP rules
    - ipfstat for viewing per-interface statistics
    - ipmon for viewing of logged packets
- Good info at http://www.obfuscation.org/ipf/



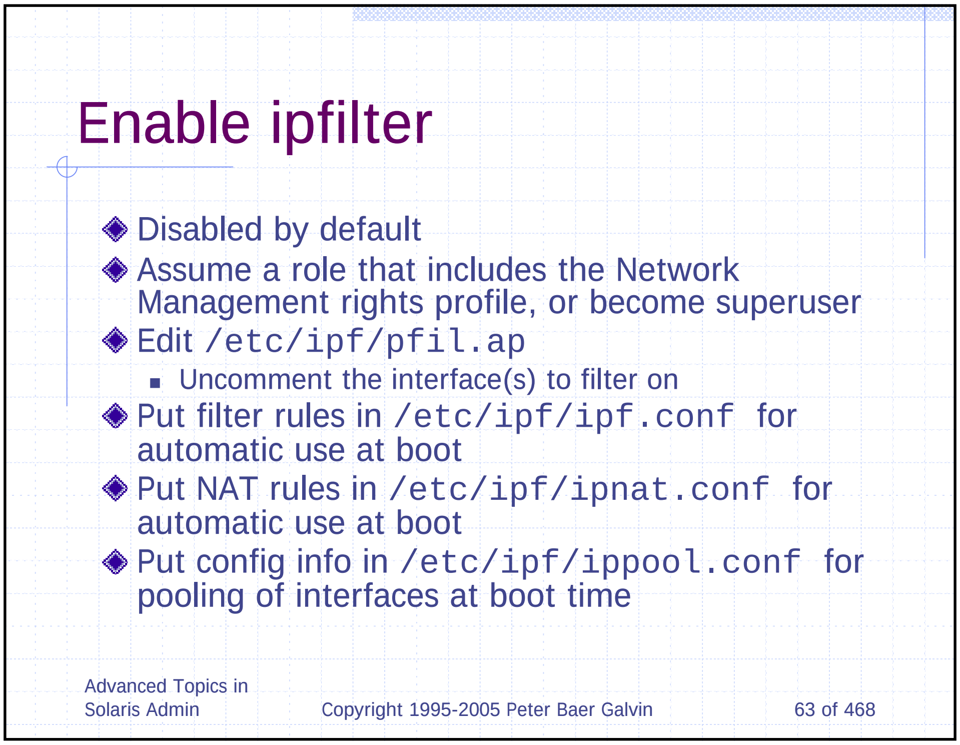
ipfilter Details

- Can match on the following IP header fields
  - Source or destination IP address (including inverted matches)
  - IP protocol
  - TOS (Type of Service)
  - IP options or IP security classes
  - Fragment
- In addition it can:
  - Distinguish between various interfaces
  - Return an ICMP error or TCP reset for denied packets
  - Keep packet state information for TCP, UDP, and ICMP packet flows
  - Keep fragment state information for any IP packet, applying the same rule to all fragments in that packet
  - Use redirection to set up true transparent proxy connections
  - Provide packet header details to a user program for authentication
  - Provide temporary storage of pre-authenticated rules for passing packets



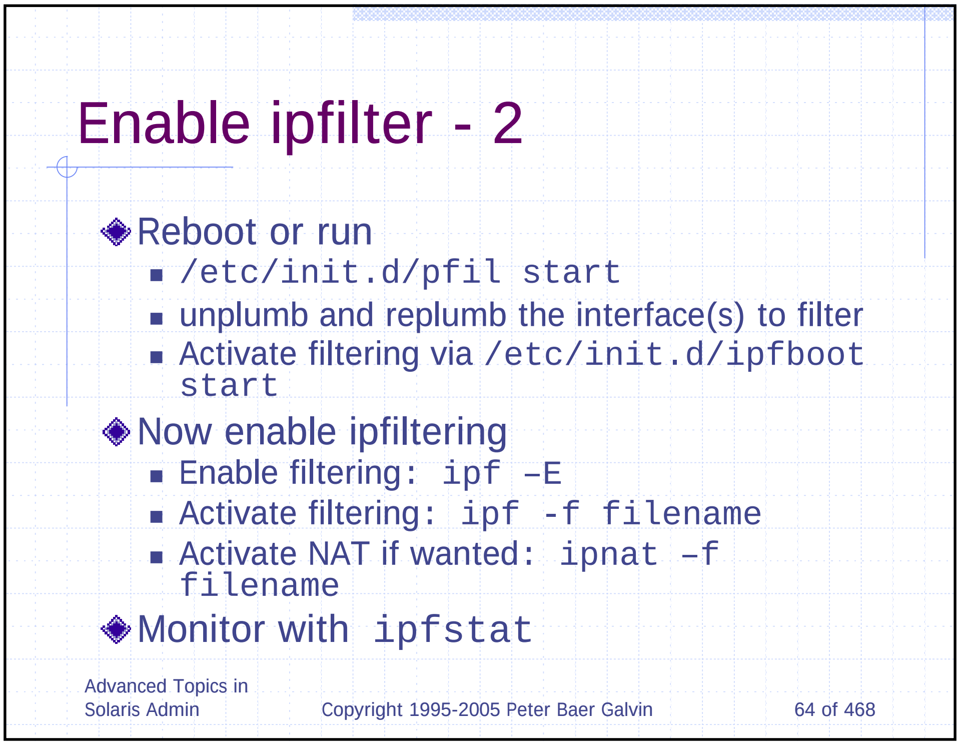
ipfilter Details - 2

- Special provision is made for the three most common Internet protocols, TCP, UDP and ICMP. Can match based on:
  - TCP or UDP packets by port number or a port number range
  - ICMP packets by type or code
  - Established TCP packet sessions
  - Any arbitrary combination of TCP flags



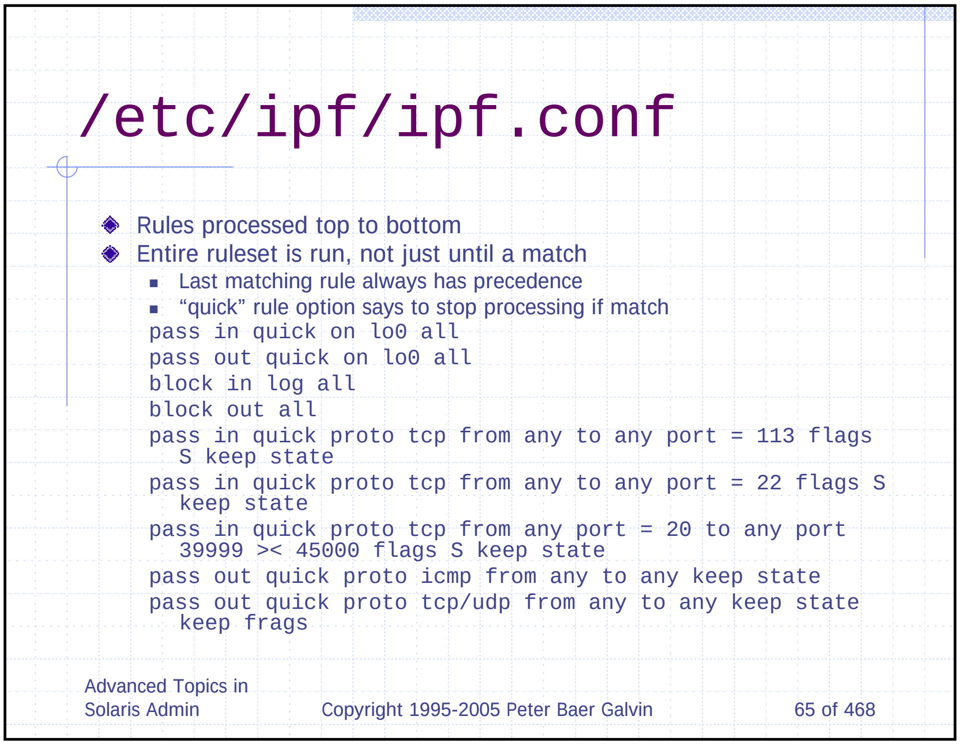
Enable ipfilter

- Disabled by default
- Assume a role that includes the Network Management rights profile, or become superuser
- Edit `/etc/ipf/pfil.ap`
  - Uncomment the interface(s) to filter on
- Put filter rules in `/etc/ipf/ipf.conf` for automatic use at boot
- Put NAT rules in `/etc/ipf/ipnat.conf` for automatic use at boot
- Put config info in `/etc/ipf/ippool.conf` for pooling of interfaces at boot time



Enable ipfilter - 2

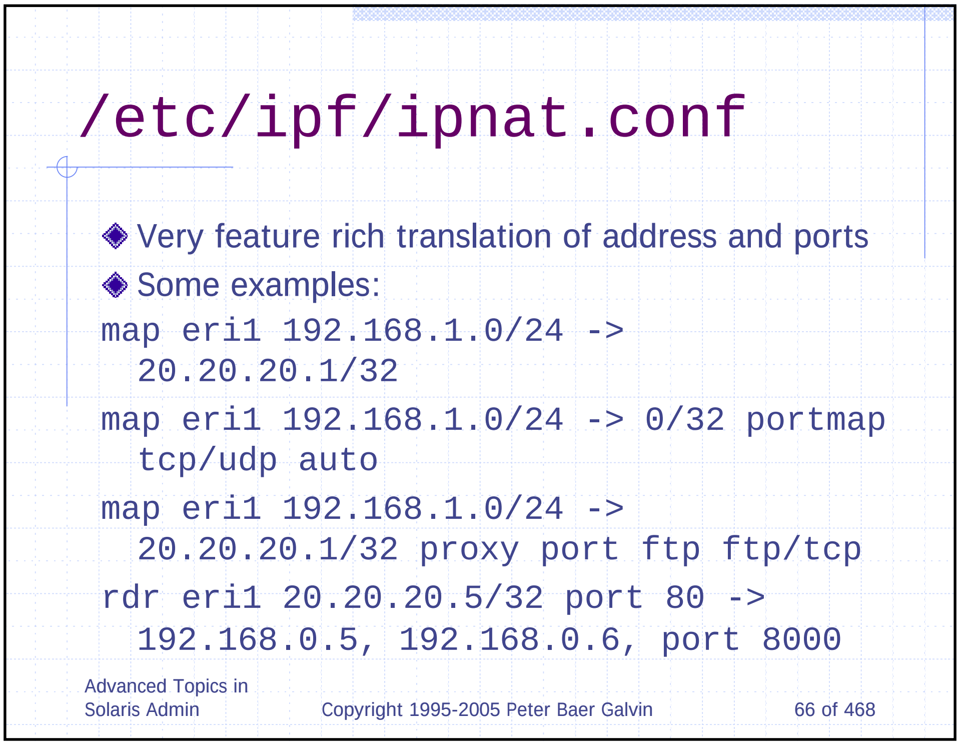
- Reboot or run
  - `/etc/init.d/pfil start`
  - unplumb and replumb the interface(s) to filter
  - Activate filtering via `/etc/init.d/ipfboot start`
- Now enable ipfiltering
  - Enable filtering: `ipf –E`
  - Activate filtering: `ipf -f filename`
  - Activate NAT if wanted: `ipnat –f filename`
- Monitor with `ipfstat`



/etc/ipf/ipf.conf

- Rules processed top to bottom
- Entire ruleset is run, not just until a match
  - Last matching rule always has precedence
  - “quick” rule option says to stop processing if match

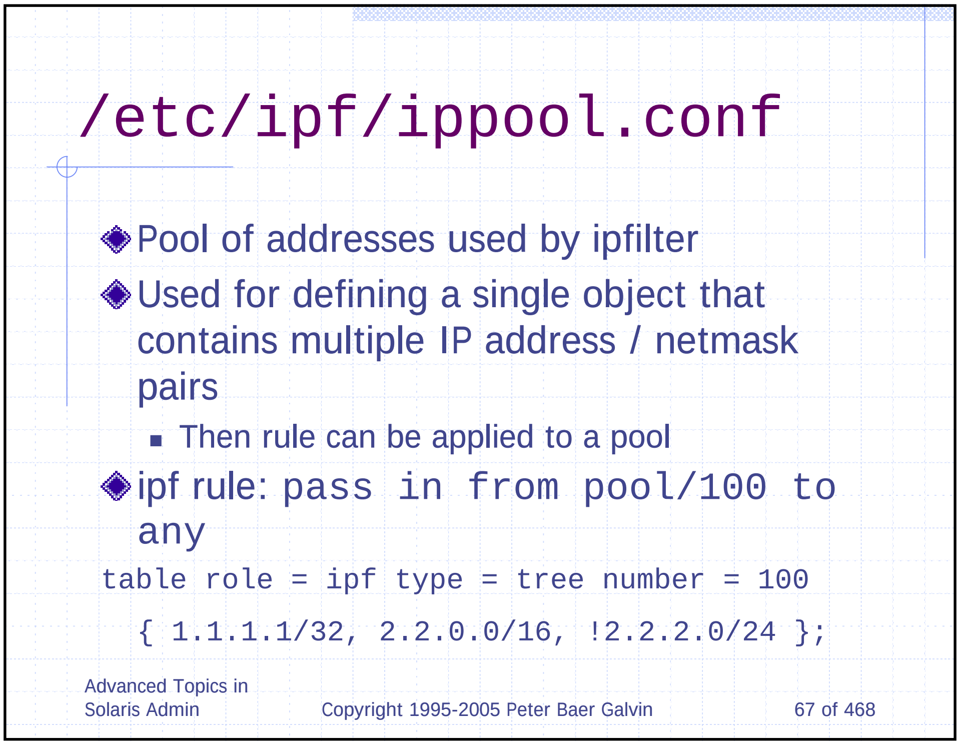
```
pass in quick on lo0 all
pass out quick on lo0 all
block in log all
block out all
pass in quick proto tcp from any to any port = 113 flags
   S keep state
pass in quick proto tcp from any to any port = 22 flags S
   keep state
pass in quick proto tcp from any port = 20 to any port
   39999 >< 45000 flags S keep state
pass out quick proto icmp from any to any keep state
pass out quick proto tcp/udp from any to any keep state
   keep frags
```



/etc/ipf/ipnat.conf

- Very feature rich translation of address and ports
- Some examples:

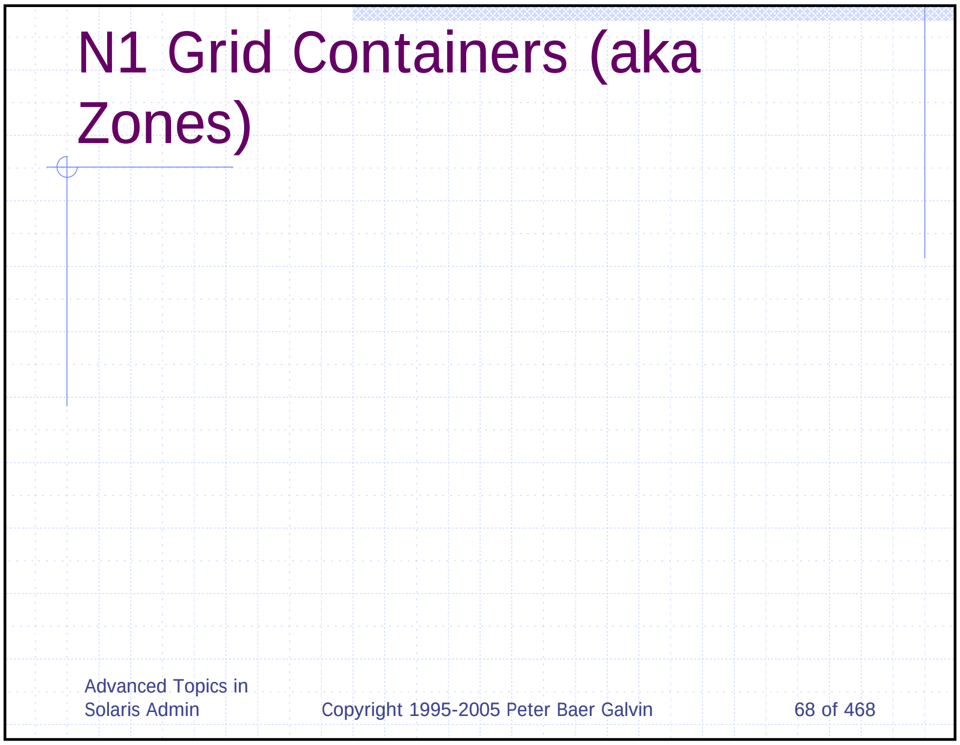
```
map eri1 192.168.1.0/24 ->
   20.20.20.1/32
map eri1 192.168.1.0/24 -> 0/32 portmap
   tcp/udp auto
map eri1 192.168.1.0/24 ->
   20.20.20.1/32 proxy port ftp ftp/tcp
rdr eri1 20.20.20.5/32 port 80 ->
   192.168.0.5, 192.168.0.6, port 8000
```



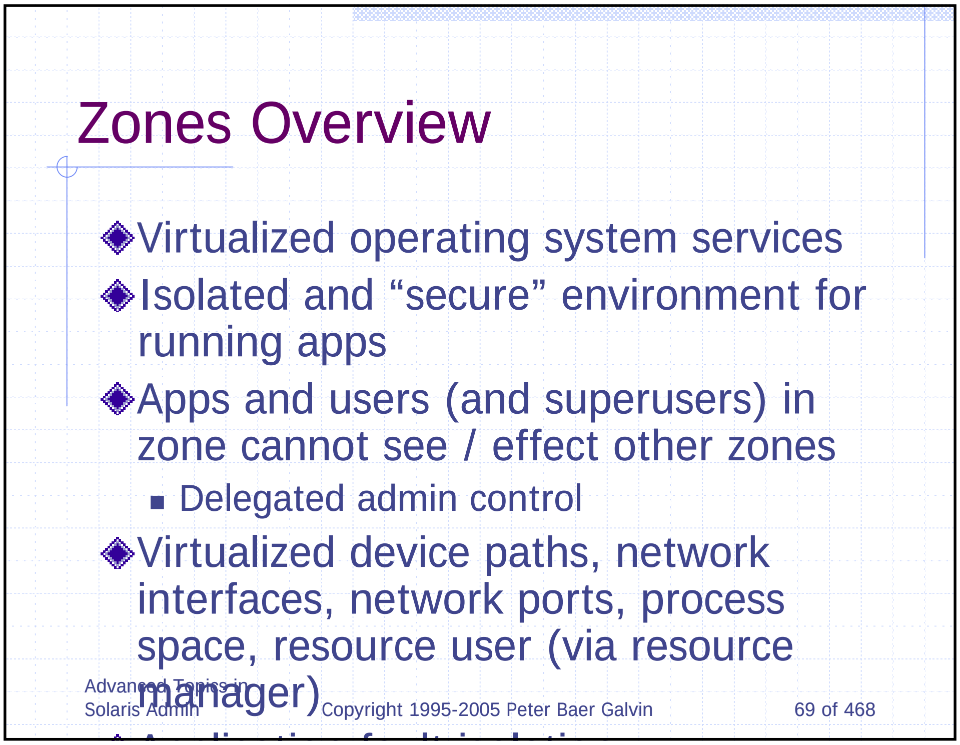
/etc/ipf/ippool.conf

- Pool of addresses used by ipfilter
- Used for defining a single object that contains multiple IP address / netmask pairs
  - Then rule can be applied to a pool
- ipf rule: `pass in from pool/100 to any`

```
table role = ipf type = tree number = 100
  { 1.1.1.1/32, 2.2.0.0/16, !2.2.2.0/24 };
```

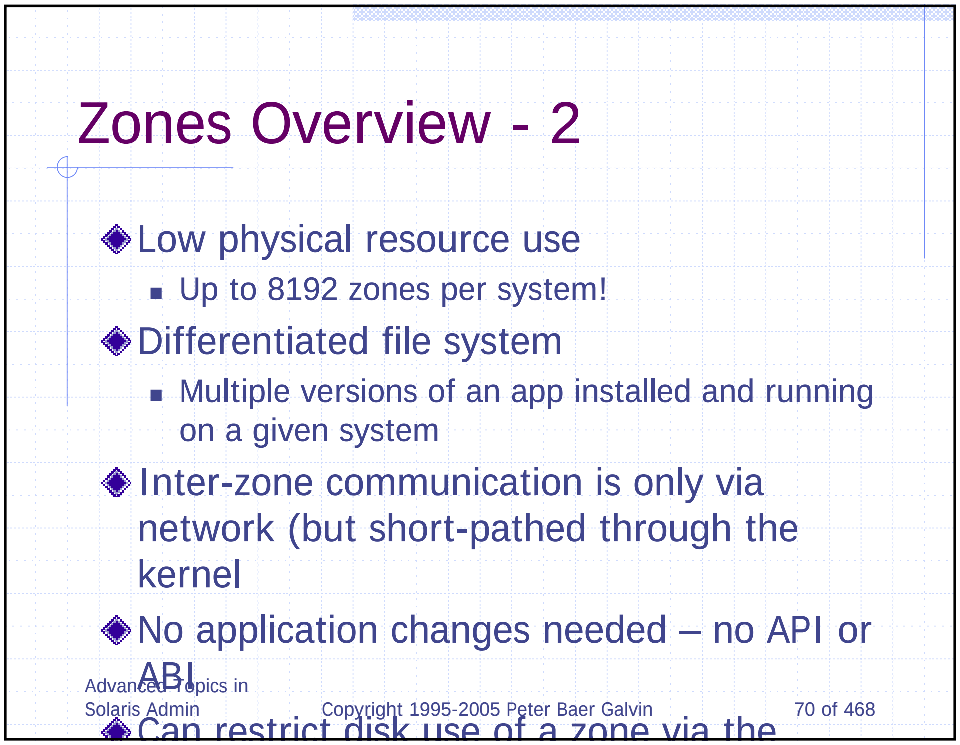


N1 Grid Containers (aka Zones)



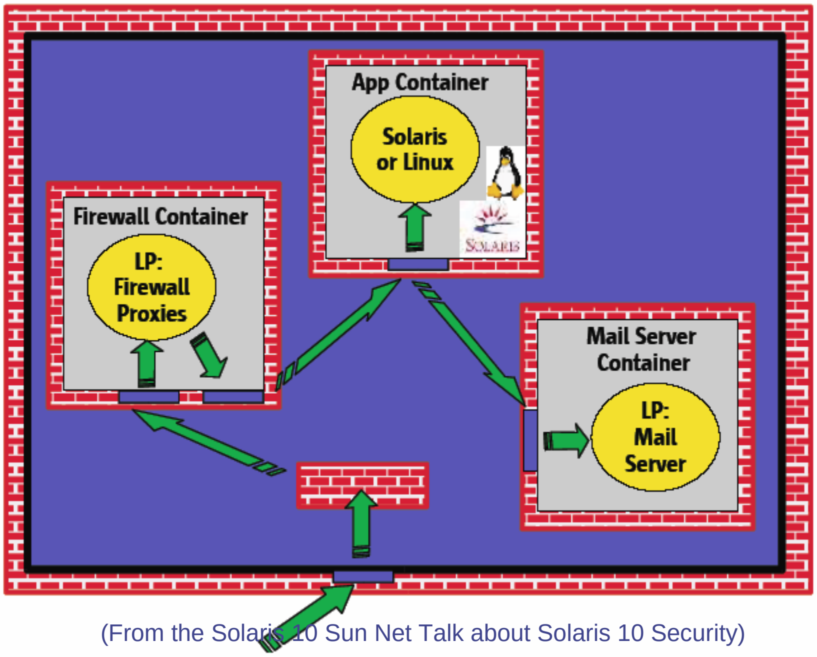
Zones Overview

- Virtualized operating system services
- Isolated and “secure” environment for running apps
- Apps and users (and superusers) in zone cannot see / effect other zones
  - Delegated admin control
- Virtualized device paths, network interfaces, network ports, process space, resource user (via resource manager)



Zones Overview - 2

- Low physical resource use
  - Up to 8192 zones per system!
- Differentiated file system
  - Multiple versions of an app installed and running on a given system
- Inter-zone communication is only via network (but short-pathed through the kernel
- No application changes needed – no API or ABI
- Can restrict disk use of a zone via the



(From the Solaris 10 Sun Net Talk about Solaris 10 Security)

Zone Limits

- Only one OS installed on a system
- One set of OS patches
- Only one `/etc/system`
  - Although Sun working to move as many settings as possible out of `/etc/system`
- System crash / OS crash -> all zones crash
- Zones cannot be moved between systems (yet)
- Each zone uses
  - ~ 100MB of disk

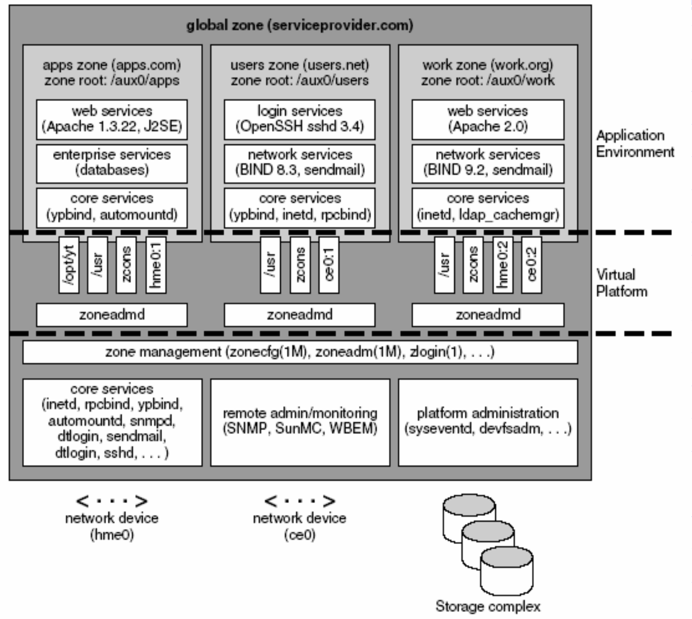
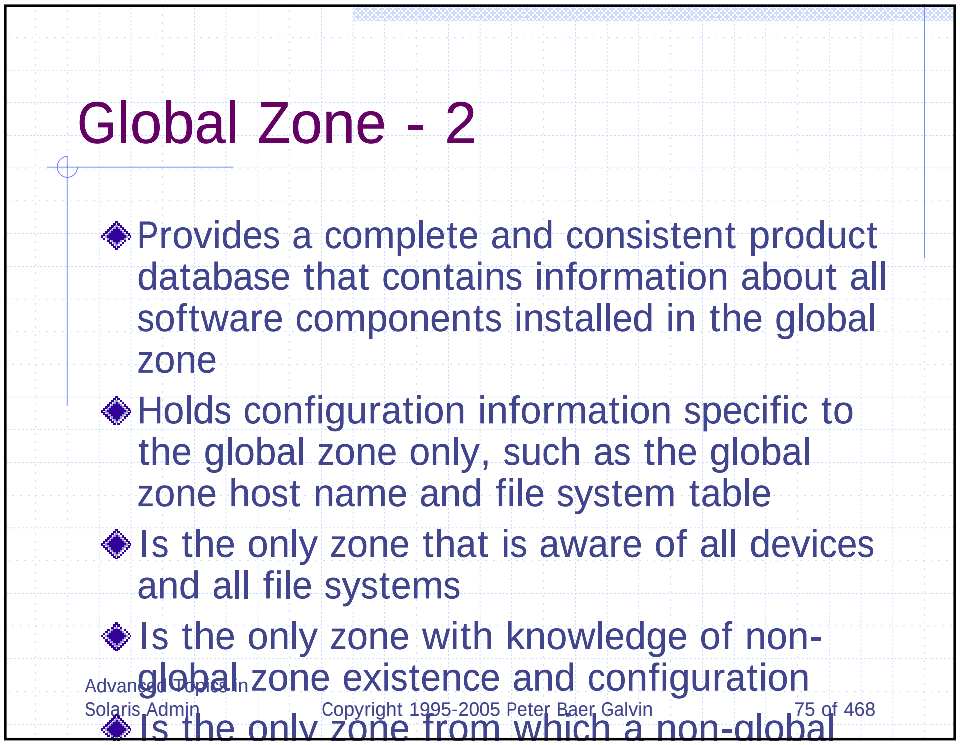


FIGURE 16–1 Zones Server Consolidation Example

(From System Administration Guide: N1 Grid Containers, Resource Management, and Solaris Zones)

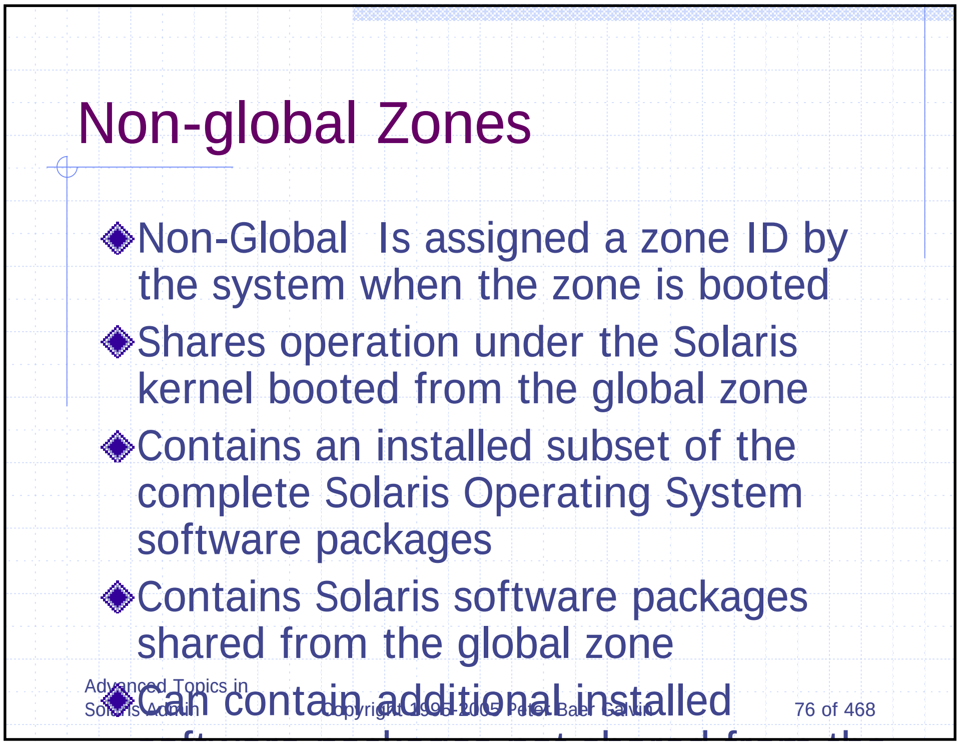
Global Zone

- Aka the usual system
- Global Is assigned ID 0 by the system
- Provides the single instance of the Solaris kernel that is bootable and running on the system
- Contains a complete installation of the Solaris system software packages
- Can contain additional software packages or additional software, directories, files, and other data not



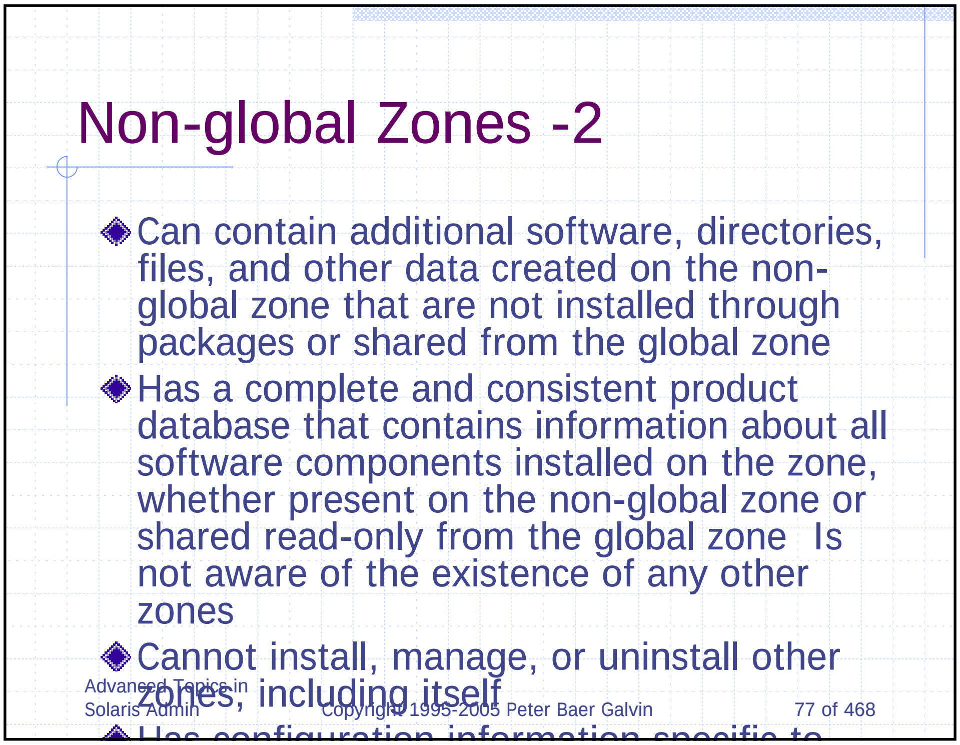
Global Zone - 2

- Provides a complete and consistent product database that contains information about all software components installed in the global zone
- Holds configuration information specific to the global zone only, such as the global zone host name and file system table
- Is the only zone that is aware of all devices and all file systems
- Is the only zone with knowledge of non-global zone existence and configuration
- Is the only zone from which a non-global



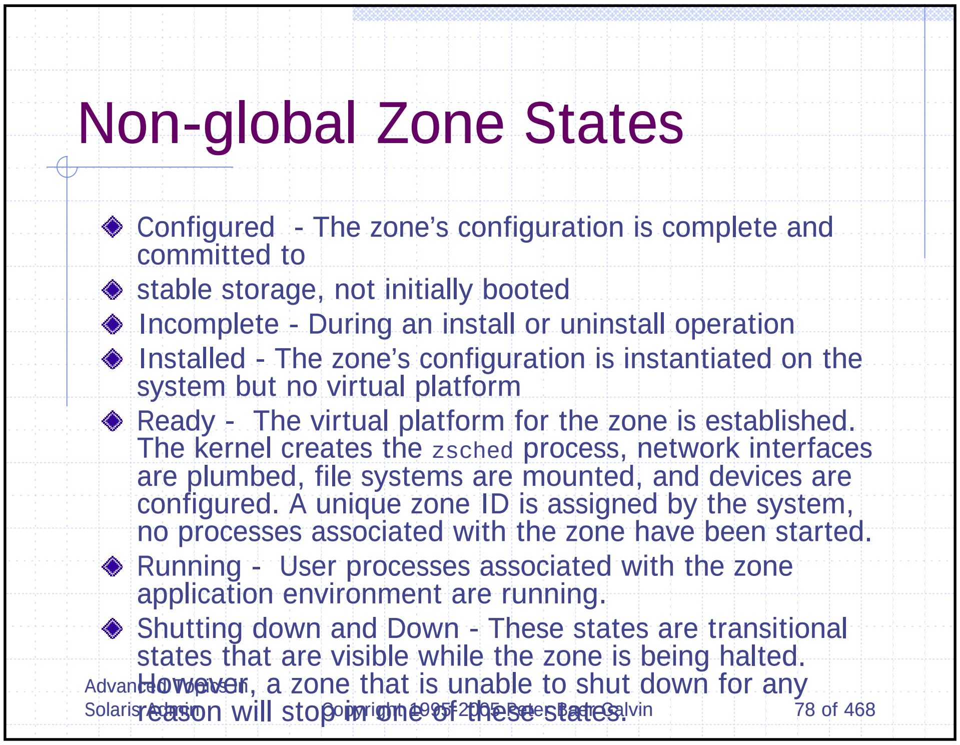
Non-global Zones

- Non-Global Is assigned a zone ID by the system when the zone is booted
- Shares operation under the Solaris kernel booted from the global zone
- Contains an installed subset of the complete Solaris Operating System software packages
- Contains Solaris software packages shared from the global zone
- Can contain additional installed



Non-global Zones -2

- Can contain additional software, directories, files, and other data created on the non-global zone that are not installed through packages or shared from the global zone
- Has a complete and consistent product database that contains information about all software components installed on the zone, whether present on the non-global zone or shared read-only from the global zone Is not aware of the existence of any other zones
- Cannot install, manage, or uninstall other zones, including itself
- Has configuration information specific to



Non-global Zone States

- Configured - The zone's configuration is complete and committed to
- stable storage, not initially booted
- Incomplete - During an install or uninstall operation
- Installed - The zone's configuration is instantiated on the system but no virtual platform
- Ready - The virtual platform for the zone is established. The kernel creates the `zsched` process, network interfaces are plumbed, file systems are mounted, and devices are configured. A unique zone ID is assigned by the system, no processes associated with the zone have been started.
- Running - User processes associated with the zone application environment are running.
- Shutting down and Down - These states are transitional states that are visible while the zone is being halted. However, a zone that is unable to shut down for any reason will stop in one of these states.

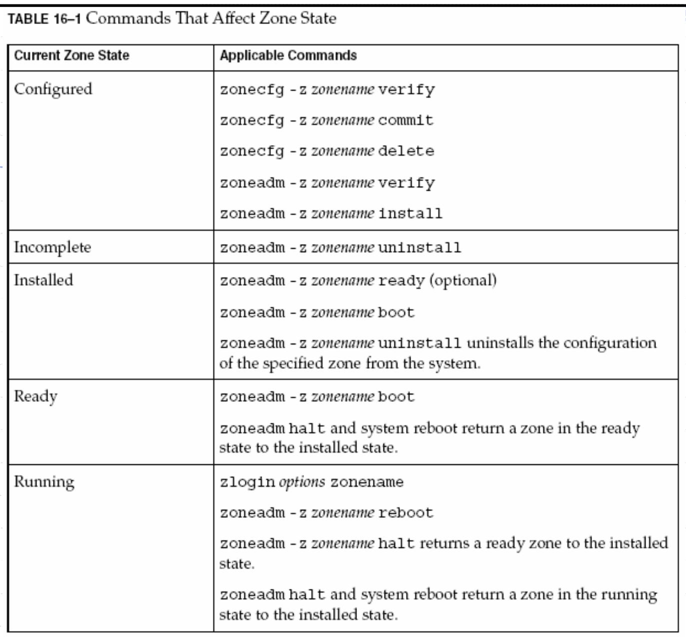


TABLE 16–1 Commands That Affect Zone State

Current Zone State	Applicable Commands

Configured	`zonecfg -z` *zonename* `verify`
	`zonecfg -z` *zonename* `commit`
	`zonecfg -z` *zonename* `delete`
	`zoneadm -z` *zonename* `verify`
	`zoneadm -z` *zonename* `install`

Incomplete	`zoneadm -z` *zonename* `uninstall`
Installed	`zoneadm -z` *zonename* `ready` (optional) `zoneadm -z` *zonename* `boot` `zoneadm -z` *zonename* `uninstall` uninstalls the configuration of the specified zone from the system.
Ready	`zoneadm -z` *zonename* `boot` `zoneadm halt` and system reboot return a zone in the ready state to the installed state.

Running	`zlogin` *options* `zonename`
	`zoneadm -z` *zonename* `reboot`
	`zoneadm -z` *zonename* `halt` returns a ready zone to the installed state.
	`zoneadm halt` and system reboot return a zone in the running state to the installed state.

(From System Administration Guide: N1Grid Containers, Resource Management, and Solaris Zones)

Zone Configuration

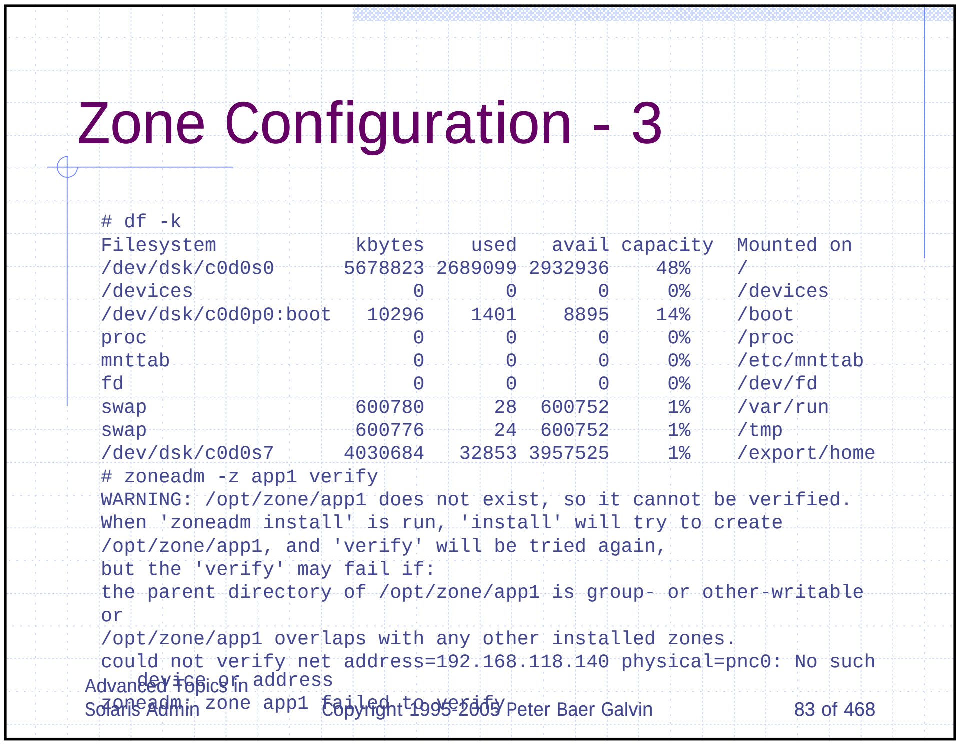
- Data from the following are not referenced or copied when a zone is installed:
  - Non-installed packages
  - Patches
  - Data on CDs and DVDs
  - Network installation images
  - Any prototype or other instance of a zone
- In addition, the following types of information, if present in the global zone, are not copied into a zone that is being installed:
  - New or changed users in the `/etc/passwd` file
  - New or changed groups in the `/etc/group` file
  - Configurations for networking services such as DHCP address assignment, UUCP, or `sendmail`
  - Configurations for network services such as naming services
  - New or changed `crontab`, printer, and mail files
  - System log, message, and accounting files

Zone Configuration

- `Zlogin –C` logs in to a just-boot virgin zone
  - Only root can `zlogin` – normal zone access is via network
- The usual `sysidconfig` questions are asked (hostname, name service, timezone, kerberos)
- Zone reboots to put configuration changes into effect (a few seconds)
  - Messages look like a system reboot (within your window)

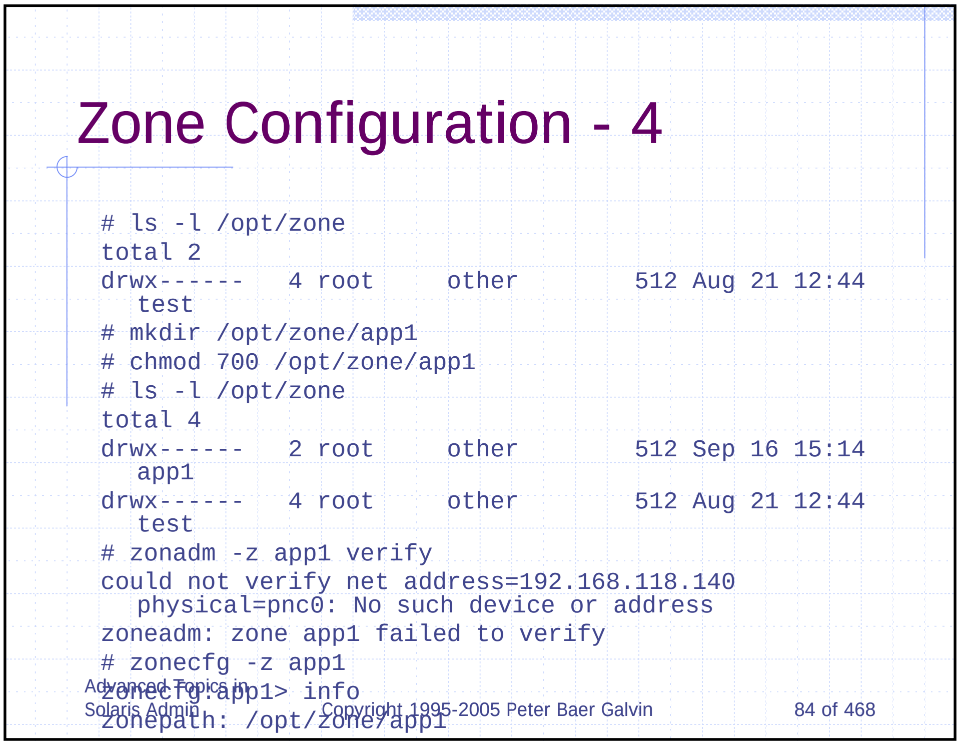
Zone Configuration - 2

```
# zonecfg -z app1
app1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:app1> create
zonecfg:app1> set zonepath=/opt/zone/app1
zonecfg:app1> set autoboot=false
zonecfg:app1> add net
zonecfg:app1:net> set physical=pnc0
zonecfg:app1:net> set address=192.168.118.140
zonecfg:app1:net> end
zonecfg:app1> add fs
zonecfg:app1:fs> set dir=/export/home
zonecfg:app1:fs> set special=/export/home
zonecfg:app1:fs> set type=lofs
zonecfg:app1> add inherit-package-dir
zonecfg:app1:inherit-pkg-dir> set dir=/opt/sfw
zonecfg:app1:inherit-pkg-dir> end
zonecfg:app1> verify
zonecfg:app1> commit
zonecfg:app1> exit
```



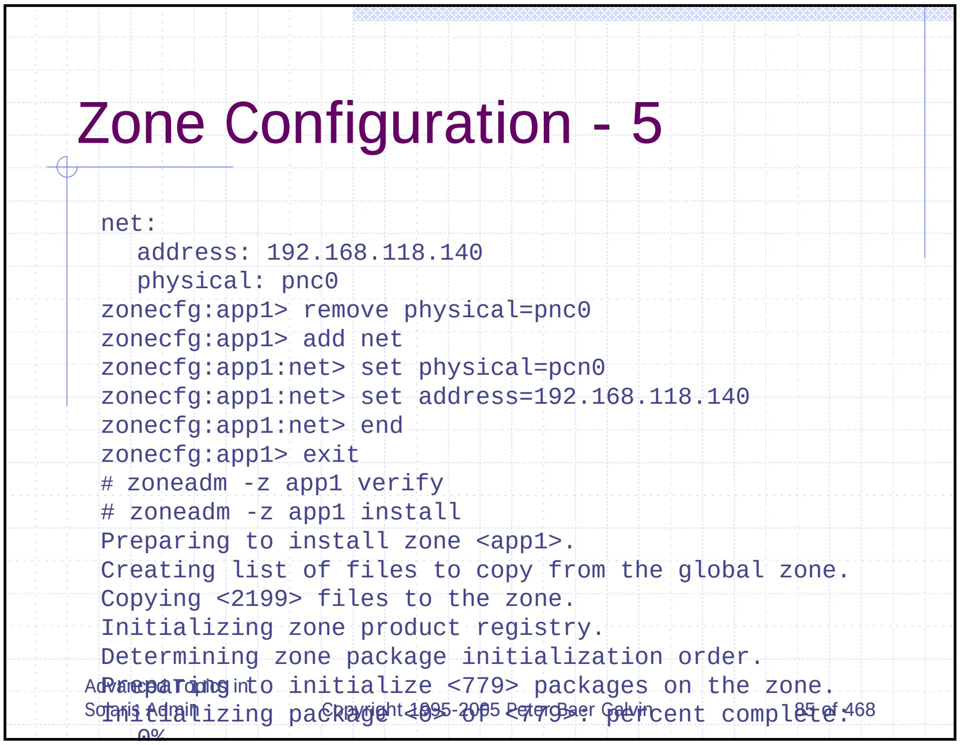
Zone Configuration - 3

```
# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0d0s0      5678823 2689099 2932936    48%    /
/devices                   0       0       0     0%    /devices
/dev/dsk/c0d0p0:boot   10296    1401    8895    14%    /boot
proc                       0       0       0     0%    /proc
mnttab                     0       0       0     0%    /etc/mnttab
fd                         0       0       0     0%    /dev/fd
swap                  600780      28  600752     1%    /var/run
swap                  600776      24  600752     1%    /tmp
/dev/dsk/c0d0s7      4030684   32853 3957525     1%    /export/home
# zoneadm -z app1 verify
WARNING: /opt/zone/app1 does not exist, so it cannot be verified.
When 'zoneadm install' is run, 'install' will try to create
/opt/zone/app1, and 'verify' will be tried again,
but the 'verify' may fail if:
the parent directory of /opt/zone/app1 is group- or other-writable
or
/opt/zone/app1 overlaps with any other installed zones.
could not verify net address=192.168.118.140 physical=pnc0: No such
   device or address
zoneadm: zone app1 failed to verify
```



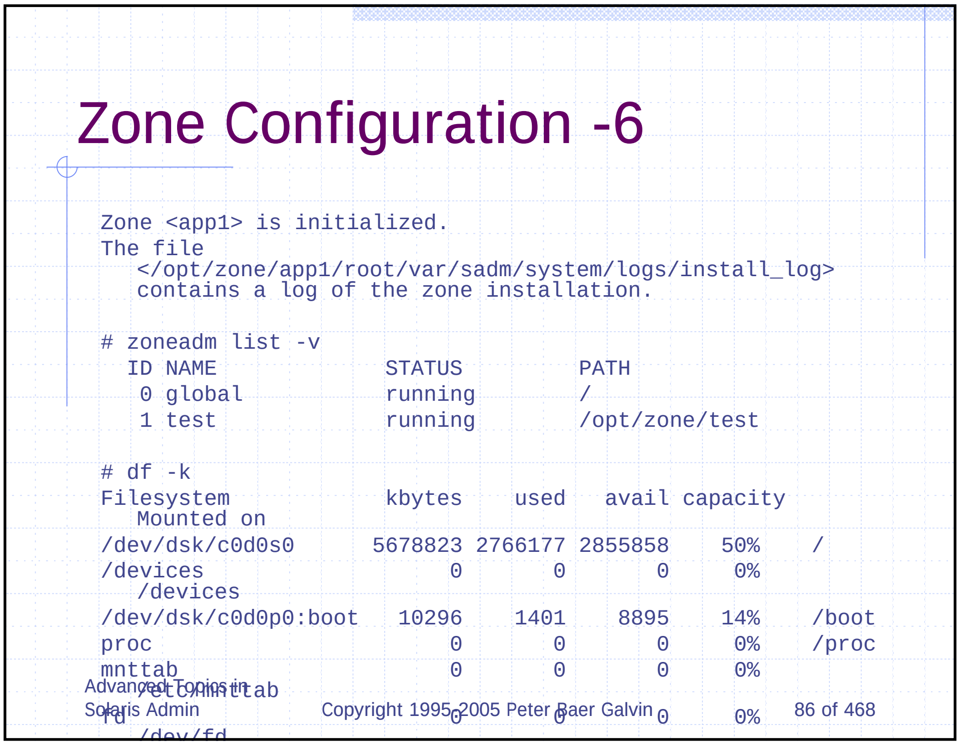
Zone Configuration - 4

```
# ls -l /opt/zone
total 2
drwx------   4 root     other        512 Aug 21 12:44
   test
# mkdir /opt/zone/app1
# chmod 700 /opt/zone/app1
# ls -l /opt/zone
total 4
drwx------   2 root     other        512 Sep 16 15:14
   app1
drwx------   4 root     other        512 Aug 21 12:44
   test
# zonadm -z app1 verify
could not verify net address=192.168.118.140
   physical=pnc0: No such device or address
zoneadm: zone app1 failed to verify
# zonecfg -z app1
zonecfg:app1> info
zonepath: /opt/zone/app1
```



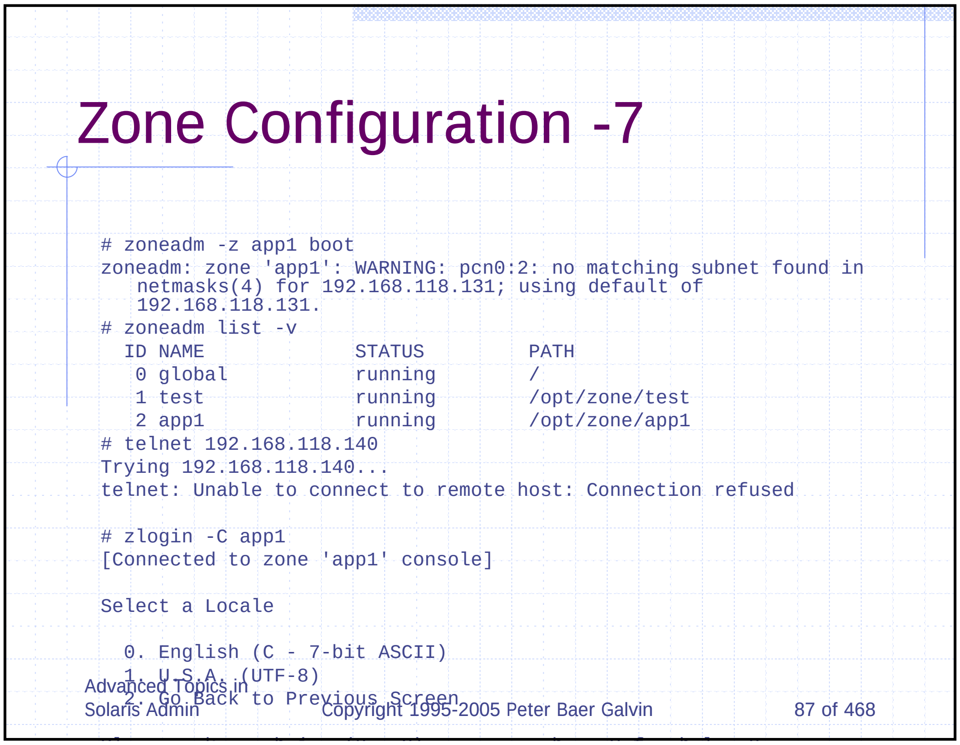
Zone Configuration - 5

```
net:
   address: 192.168.118.140
   physical: pnc0
zonecfg:app1> remove physical=pnc0
zonecfg:app1> add net
zonecfg:app1:net> set physical=pcn0
zonecfg:app1:net> set address=192.168.118.140
zonecfg:app1:net> end
zonecfg:app1> exit
# zoneadm -z app1 verify
# zoneadm -z app1 install
Preparing to install zone <app1>.
Creating list of files to copy from the global zone.
Copying <2199> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <779> packages on the zone.
Initializing package <0> of <779>: percent complete:
   0%
```



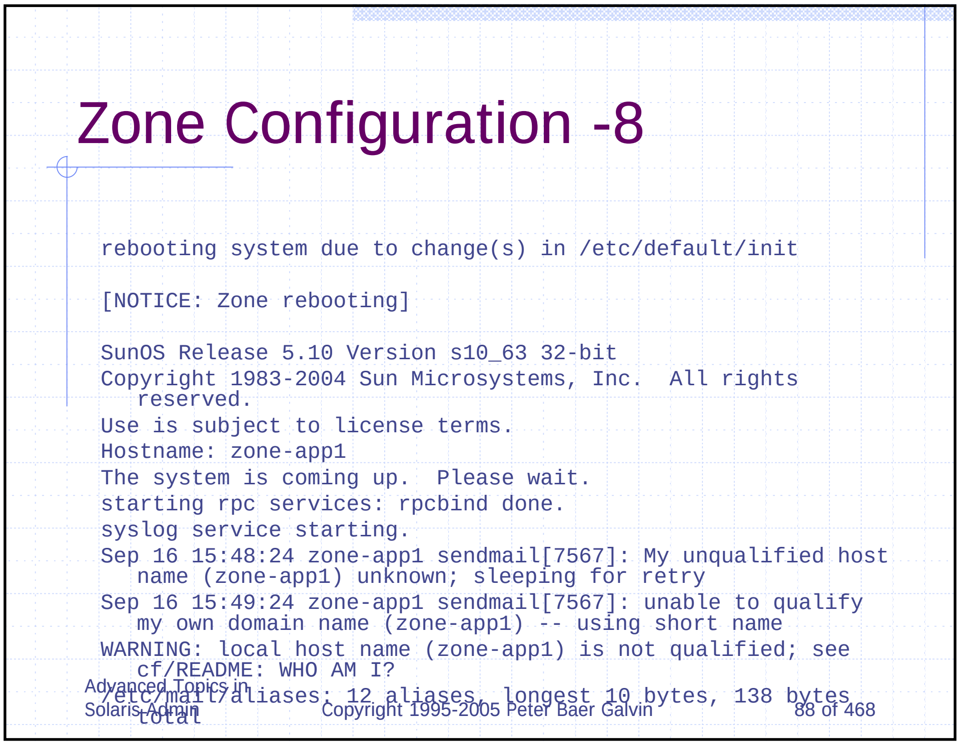
Zone Configuration -6

```
Zone <app1> is initialized.
The file
   </opt/zone/app1/root/var/sadm/system/logs/install_log>
   contains a log of the zone installation.

# zoneadm list -v
  ID NAME             STATUS         PATH
   0 global           running        /
   1 test             running        /opt/zone/test

# df -k
Filesystem            kbytes    used   avail capacity
   Mounted on
/dev/dsk/c0d0s0      5678823 2766177 2855858    50%    /
/devices                   0       0       0     0%
   /devices
/dev/dsk/c0d0p0:boot   10296    1401    8895    14%    /boot
proc                       0       0       0     0%    /proc
mnttab                     0       0       0     0%
   /etc/mnttab
fd                         0       0       0     0%
   /dev/fd
```



Zone Configuration -7

```
# zoneadm -z app1 boot
zoneadm: zone 'app1': WARNING: pcn0:2: no matching subnet found in
   netmasks(4) for 192.168.118.131; using default of
   192.168.118.131.
# zoneadm list -v
  ID NAME             STATUS         PATH
   0 global           running        /
   1 test             running        /opt/zone/test
   2 app1             running        /opt/zone/app1
# telnet 192.168.118.140
Trying 192.168.118.140...
telnet: Unable to connect to remote host: Connection refused

# zlogin -C app1
[Connected to zone 'app1' console]

Select a Locale

  0. English (C - 7-bit ASCII)
  1. U.S.A. (UTF-8)
  2. Go Back to Previous Screen
```



Zone Configuration -8

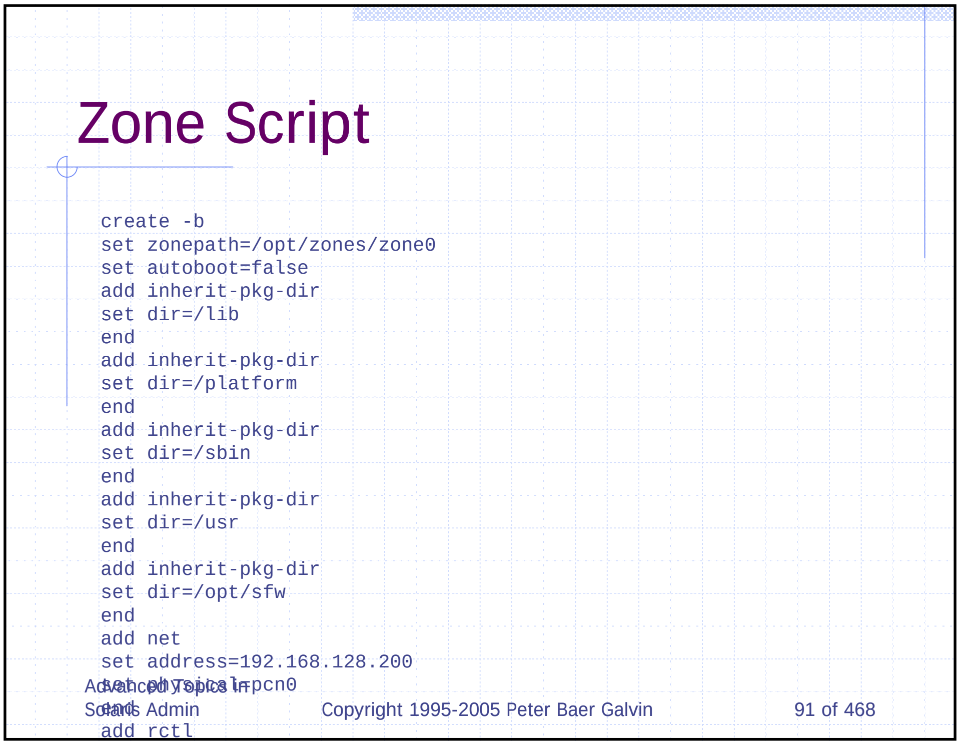
```
rebooting system due to change(s) in /etc/default/init

[NOTICE: Zone rebooting]

SunOS Release 5.10 Version s10_63 32-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights
   reserved.
Use is subject to license terms.
Hostname: zone-app1
The system is coming up.  Please wait.
starting rpc services: rpcbind done.
syslog service starting.
Sep 16 15:48:24 zone-app1 sendmail[7567]: My unqualified host
   name (zone-app1) unknown; sleeping for retry
Sep 16 15:49:24 zone-app1 sendmail[7567]: unable to qualify
   my own domain name (zone-app1) -- using short name
WARNING: local host name (zone-app1) is not qualified; see
   cf/README: WHO AM I?
/etc/mail/aliases: 12 aliases, longest 10 bytes, 138 bytes
   total
```

Zone Configuration -9

```
STSF Font Server Daemon.

Standard Type Services Framework 0.11.1
Copyright (c) 2001-2004 Sun Microsystems, Inc. All Rights
   Reserved.
STSF is Open Source Software. http://stsf.freedesktop.org

Creating new rsa public/private host key pair
Creating new dsa public/private host key pair
The system is ready.
zone-app1 console login: root
Password:
Sep 16 15:51:08 zone-app1 login: ROOT LOGIN /dev/console
Sun Microsystems Inc.   SunOS 5.10      s10_63  May 2004
# cat /etc/passwd
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
```

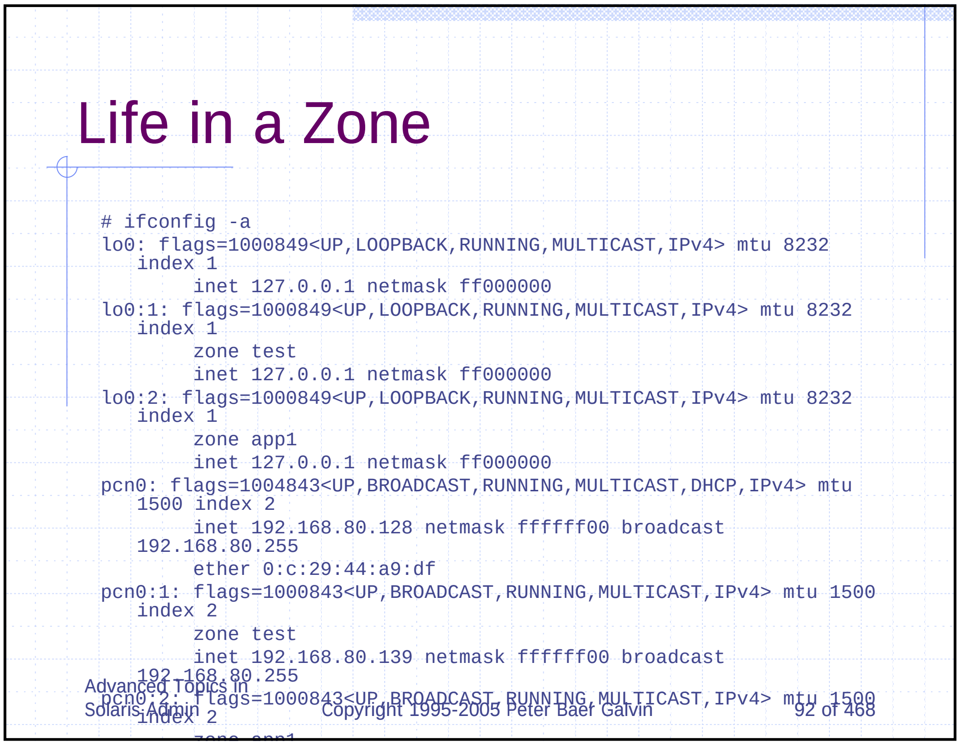
Zone Configuration -10

```
# useradd -u 101 -g 14 -d /export/home/pbg -s /bin/bash pbg
# passwd pbg
New Password:
Re-enter new Password:
passwd: password successfully changed for pbg
# zoneadm list -v
  ID NAME             STATUS         PATH
   3 app1             running        /

# exit
zone-app1 console login: ~.
[Connection to zone 'app1' console closed]

# zoneadm list -v
  ID NAME             STATUS         PATH
   0 global           running        /
   1 test             running        /opt/zone/test
   3 app1             running        /opt/zone/app1
# uptime
  3:53pm  up  5:14,  1 user,  load average: 0.23, 0.34, 0.43
# telnet 192.168.118.140
Trying 192.168.118.140…
Connected to 192.168.118.140.
Escape character is ‘^]’.
Login: pbg
```



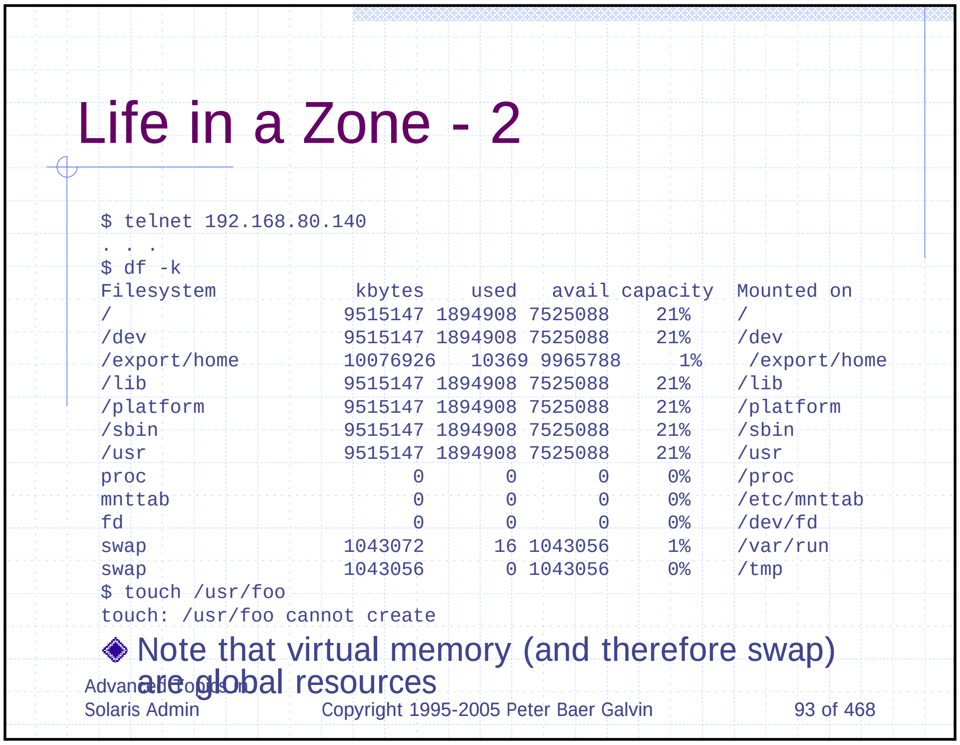
Zone Script

```
create -b
set zonepath=/opt/zones/zone0
set autoboot=false
add inherit-pkg-dir
set dir=/lib
end
add inherit-pkg-dir
set dir=/platform
end
add inherit-pkg-dir
set dir=/sbin
end
add inherit-pkg-dir
set dir=/usr
end
add inherit-pkg-dir
set dir=/opt/sfw
end
add net
set address=192.168.128.200
set physical=pcn0
end
add rctl
```



Life in a Zone

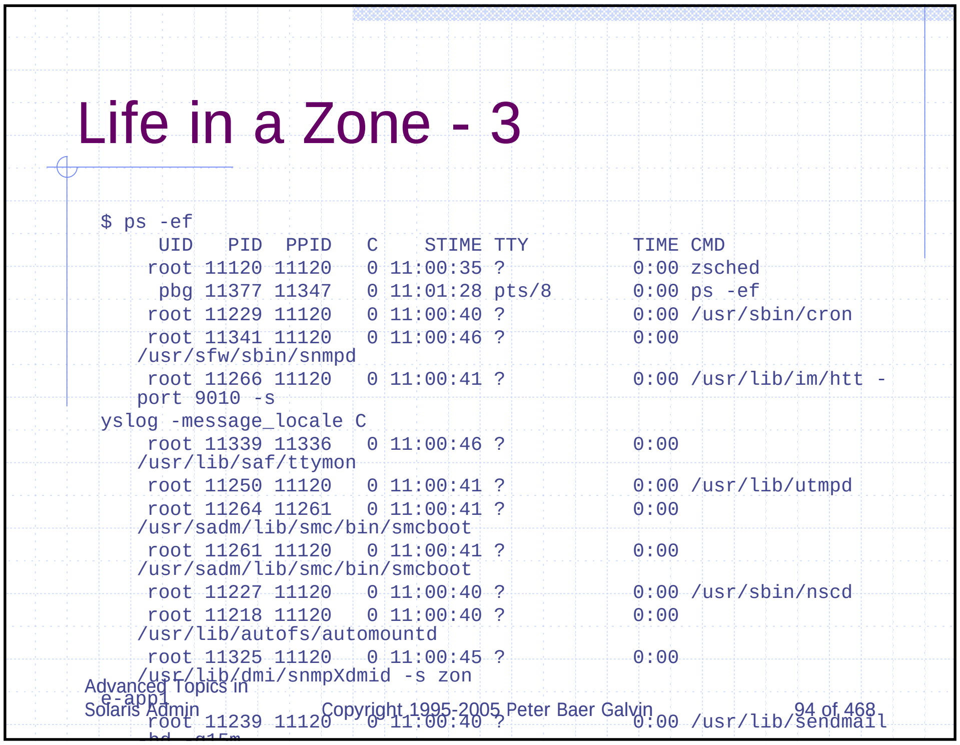
```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232
   index 1
        inet 127.0.0.1 netmask ff000000
lo0:1: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232
   index 1
        zone test
        inet 127.0.0.1 netmask ff000000
lo0:2: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232
   index 1
        zone app1
        inet 127.0.0.1 netmask ff000000
pcn0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu
   1500 index 2
        inet 192.168.80.128 netmask ffffff00 broadcast
   192.168.80.255
        ether 0:c:29:44:a9:df
pcn0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
   index 2
        zone test
        inet 192.168.80.139 netmask ffffff00 broadcast
   192.168.80.255
pcn0:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
   index 2
        zone app1
```



Life in a Zone - 2

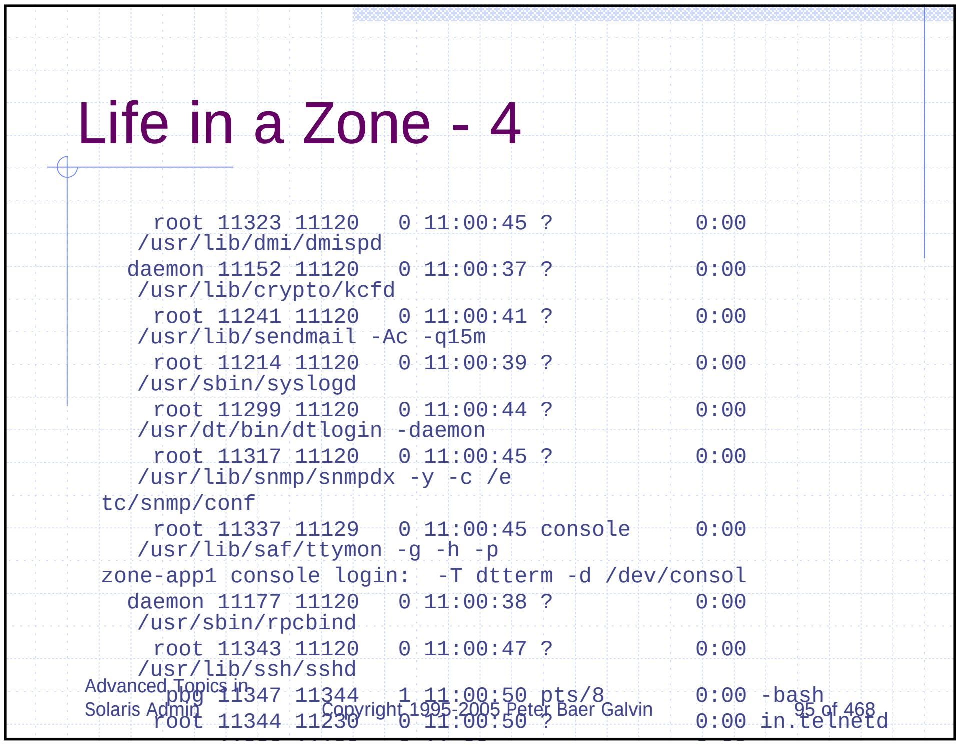
```
$ telnet 192.168.80.140
. . .
$ df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/                    9515147 1894908 7525088    21%    /
/dev                 9515147 1894908 7525088    21%    /dev
/export/home         10076926   10369 9965788     1%    /export/home
/lib                 9515147 1894908 7525088    21%    /lib
/platform            9515147 1894908 7525088    21%    /platform
/sbin                9515147 1894908 7525088    21%    /sbin
/usr                 9515147 1894908 7525088    21%    /usr
proc                       0       0       0     0%    /proc
mnttab                     0       0       0     0%    /etc/mnttab
fd                         0       0       0     0%    /dev/fd
swap                 1043072      16 1043056     1%    /var/run
swap                 1043056       0 1043056     0%    /tmp
$ touch /usr/foo
touch: /usr/foo cannot create
```

- Note that virtual memory (and therefore swap) are global resources



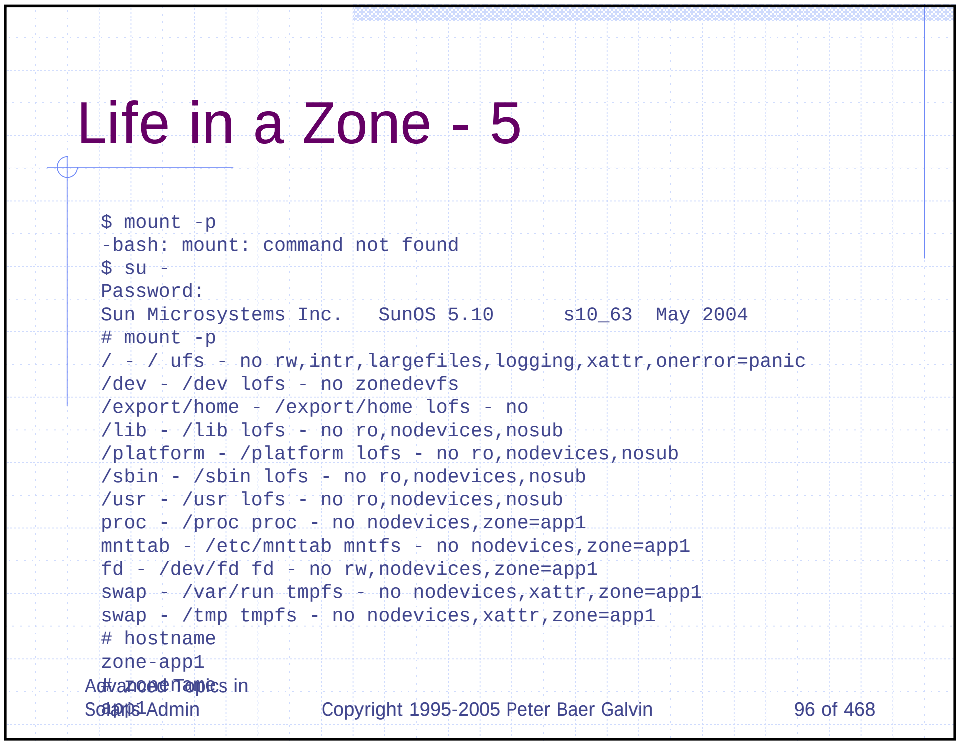
Life in a Zone - 3

```
$ ps -ef
     UID   PID  PPID   C    STIME TTY         TIME CMD
    root 11120 11120   0 11:00:35 ?           0:00 zsched
     pbg 11377 11347   0 11:01:28 pts/8       0:00 ps -ef
    root 11229 11120   0 11:00:40 ?           0:00 /usr/sbin/cron
    root 11341 11120   0 11:00:46 ?           0:00
   /usr/sfw/sbin/snmpd
    root 11266 11120   0 11:00:41 ?           0:00 /usr/lib/im/htt -
   port 9010 -s
yslog -message_locale C
    root 11339 11336   0 11:00:46 ?           0:00
   /usr/lib/saf/ttymon
    root 11250 11120   0 11:00:41 ?           0:00 /usr/lib/utmpd
    root 11264 11261   0 11:00:41 ?           0:00
   /usr/sadm/lib/smc/bin/smcboot
    root 11261 11120   0 11:00:41 ?           0:00
   /usr/sadm/lib/smc/bin/smcboot
    root 11227 11120   0 11:00:40 ?           0:00 /usr/sbin/nscd
    root 11218 11120   0 11:00:40 ?           0:00
   /usr/lib/autofs/automountd
    root 11325 11120   0 11:00:45 ?           0:00
   /usr/lib/dmi/snmpXdmid -s zon
e-app1
    root 11239 11120   0 11:00:40 ?           0:00 /usr/lib/sendmail
    bd -q15m
```



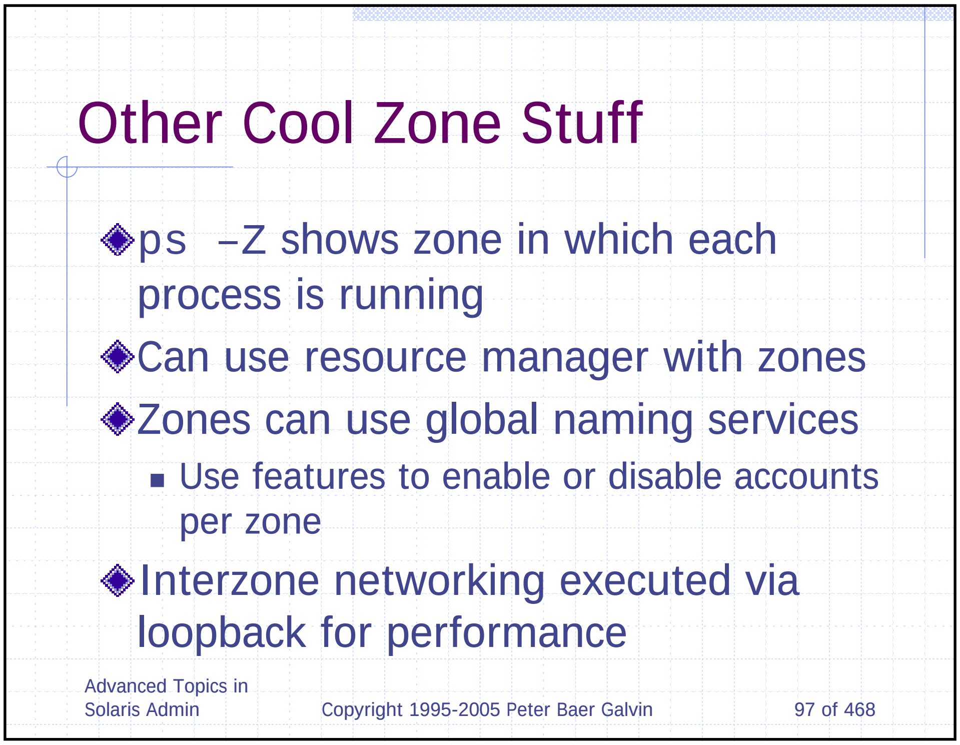
Life in a Zone - 4

```
     root 11323 11120   0 11:00:45 ?           0:00
    /usr/lib/dmi/dmispd
   daemon 11152 11120   0 11:00:37 ?           0:00
    /usr/lib/crypto/kcfd
     root 11241 11120   0 11:00:41 ?           0:00
    /usr/lib/sendmail -Ac -q15m
     root 11214 11120   0 11:00:39 ?           0:00
    /usr/sbin/syslogd
     root 11299 11120   0 11:00:44 ?           0:00
    /usr/dt/bin/dtlogin -daemon
     root 11317 11120   0 11:00:45 ?           0:00
    /usr/lib/snmp/snmpdx -y -c /e
tc/snmp/conf
     root 11337 11129   0 11:00:45 console     0:00
    /usr/lib/saf/ttymon -g -h -p
zone-app1 console login:  -T dtterm -d /dev/consol
   daemon 11177 11120   0 11:00:38 ?           0:00
    /usr/sbin/rpcbind
     root 11343 11120   0 11:00:47 ?           0:00
    /usr/lib/ssh/sshd
      pbg 11347 11344   1 11:00:50 pts/8       0:00 -bash
     root 11344 11230   0 11:00:50 ?           0:00 in.telnetd
```



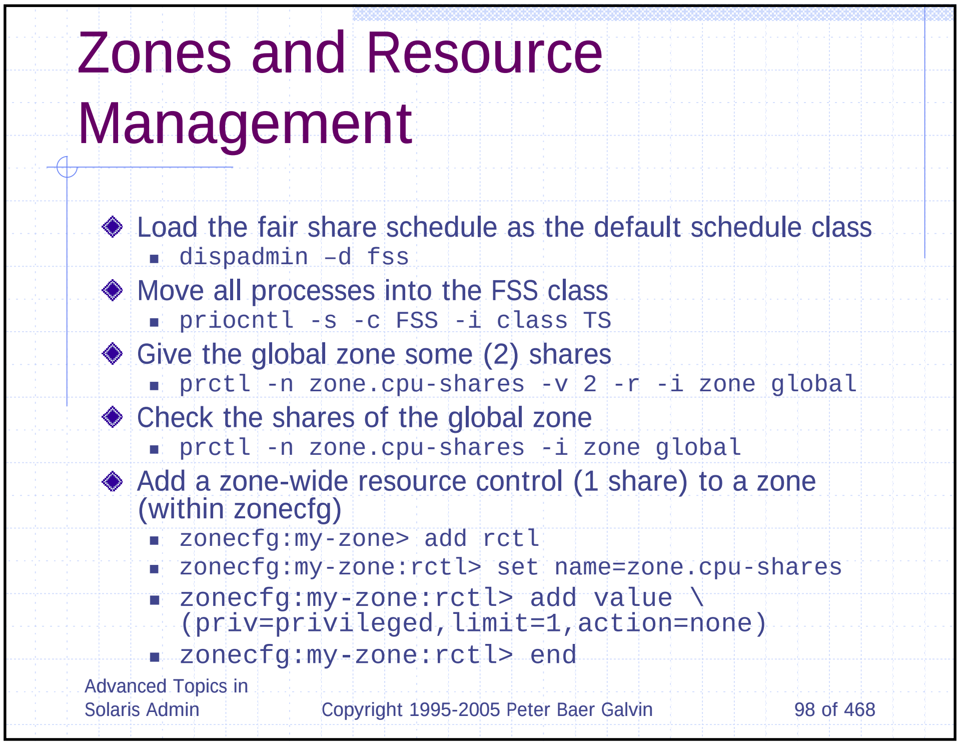
Life in a Zone - 5

```
$ mount -p
-bash: mount: command not found
$ su -
Password:
Sun Microsystems Inc.   SunOS 5.10      s10_63  May 2004
# mount -p
/ - / ufs - no rw,intr,largefiles,logging,xattr,onerror=panic
/dev - /dev lofs - no zonedevfs
/export/home - /export/home lofs - no
/lib - /lib lofs - no ro,nodevices,nosub
/platform - /platform lofs - no ro,nodevices,nosub
/sbin - /sbin lofs - no ro,nodevices,nosub
/usr - /usr lofs - no ro,nodevices,nosub
proc - /proc proc - no nodevices,zone=app1
mnttab - /etc/mnttab mntfs - no nodevices,zone=app1
fd - /dev/fd fd - no rw,nodevices,zone=app1
swap - /var/run tmpfs - no nodevices,xattr,zone=app1
swap - /tmp tmpfs - no nodevices,xattr,zone=app1
# hostname
zone-app1
# zonename
app1
```



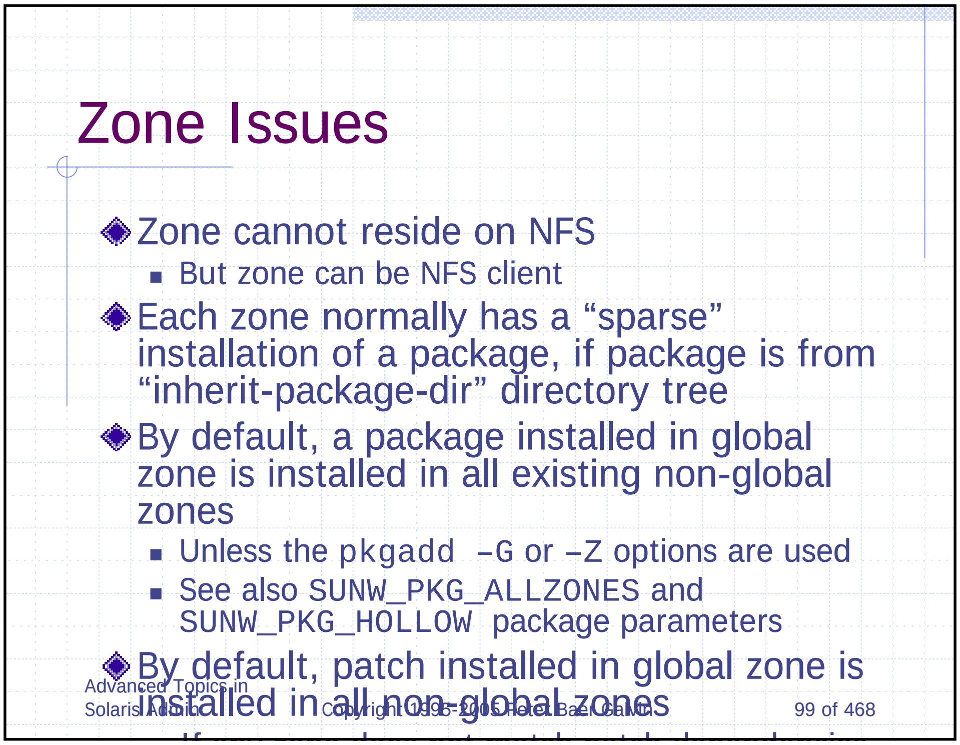
Other Cool Zone Stuff

- `ps –Z` shows zone in which each process is running
- Can use resource manager with zones
- Zones can use global naming services
  - Use features to enable or disable accounts per zone
- Interzone networking executed via loopback for performance



Zones and Resource Management

- Load the fair share schedule as the default schedule class
  - `dispadmin –d fss`
- Move all processes into the FSS class
  - `priocntl -s -c FSS -i class TS`
- Give the global zone some (2) shares
  - `prctl -n zone.cpu-shares -v 2 -r -i zone global`
- Check the shares of the global zone
  - `prctl -n zone.cpu-shares -i zone global`
- Add a zone-wide resource control (1 share) to a zone (within zonecfg)
  - `zonecfg:my-zone> add rctl`
  - `zonecfg:my-zone:rctl> set name=zone.cpu-shares`
  - `zonecfg:my-zone:rctl> add value \`
    `(priv=privileged,limit=1,action=none)`
  - `zonecfg:my-zone:rctl> end`

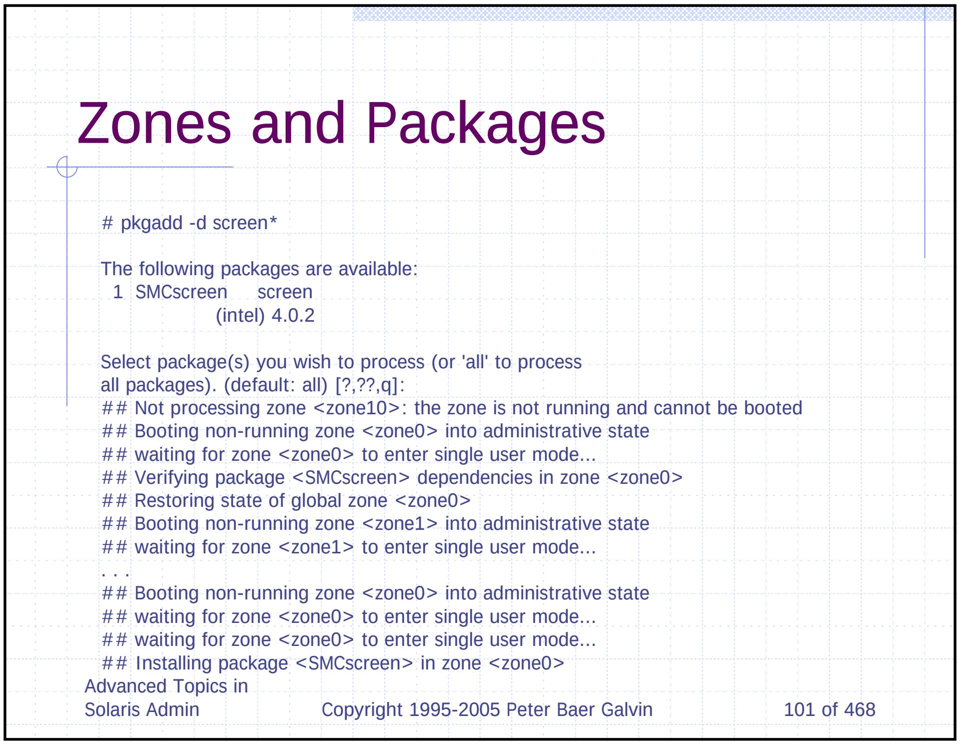


Zone Issues

- Zone cannot reside on NFS
  - But zone can be NFS client
- Each zone normally has a “sparse” installation of a package, if package is from “inherit-package-dir” directory tree
- By default, a package installed in global zone is installed in all existing non-global zones
  - Unless the `pkgadd -G` or `-Z` options are used
  - See also `SUNW_PKG_ALLZONES` and `SUNW_PKG_HOLLOW` package parameters
- By default, patch installed in global zone is installed in all non-global zones

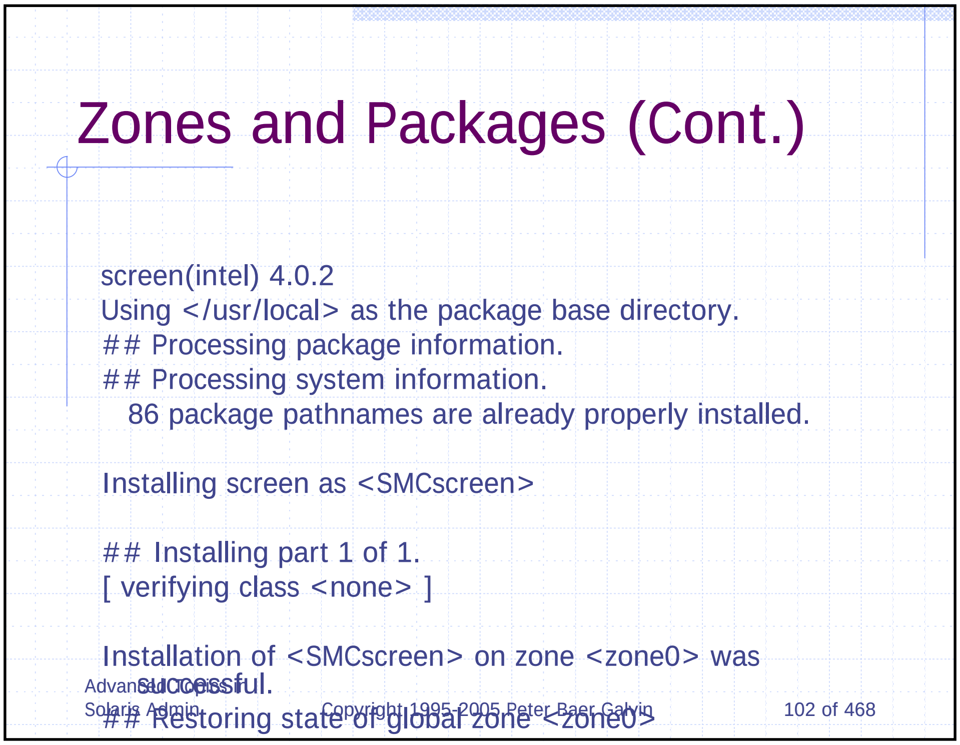
Zone issues - cont

- Upgrading the global zone to a new Solaris release upgrades the non-global zones (but only by using live upgrade)
- Best practice is to keep packages and patches synced between global and all non-global zones
- Best practice – prebuild a bunch of zones, even if you won’t need them
  - Packages and patches stay in sync or as in generic initial system
  - Low resource use
  - Use one of them for all applications & non-sys admin users
- Watch out for giving users root in a zone –



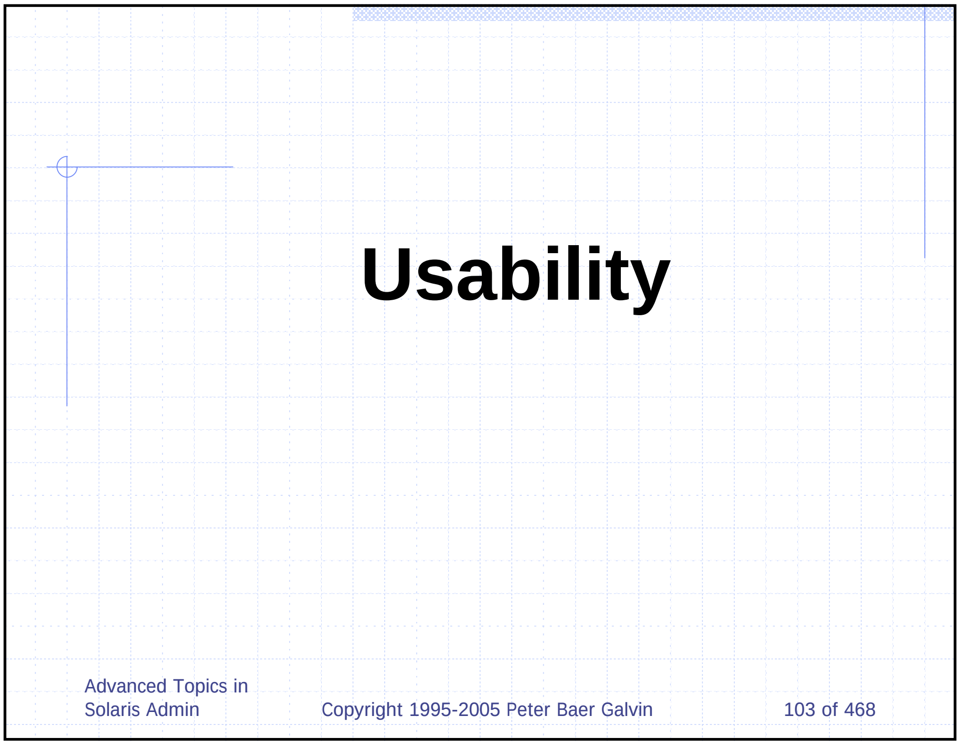
Zones and Packages

```
# pkgadd -d screen*

The following packages are available:
  1  SMCscreen     screen
                 (intel) 4.0.2

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:
## Not processing zone <zone10>: the zone is not running and cannot be booted
## Booting non-running zone <zone0> into administrative state
## waiting for zone <zone0> to enter single user mode...
## Verifying package <SMCscreen> dependencies in zone <zone0>
## Restoring state of global zone <zone0>
## Booting non-running zone <zone1> into administrative state
## waiting for zone <zone1> to enter single user mode...
. . .
## Booting non-running zone <zone0> into administrative state
## waiting for zone <zone0> to enter single user mode...
## waiting for zone <zone0> to enter single user mode...
## Installing package <SMCscreen> in zone <zone0>
```

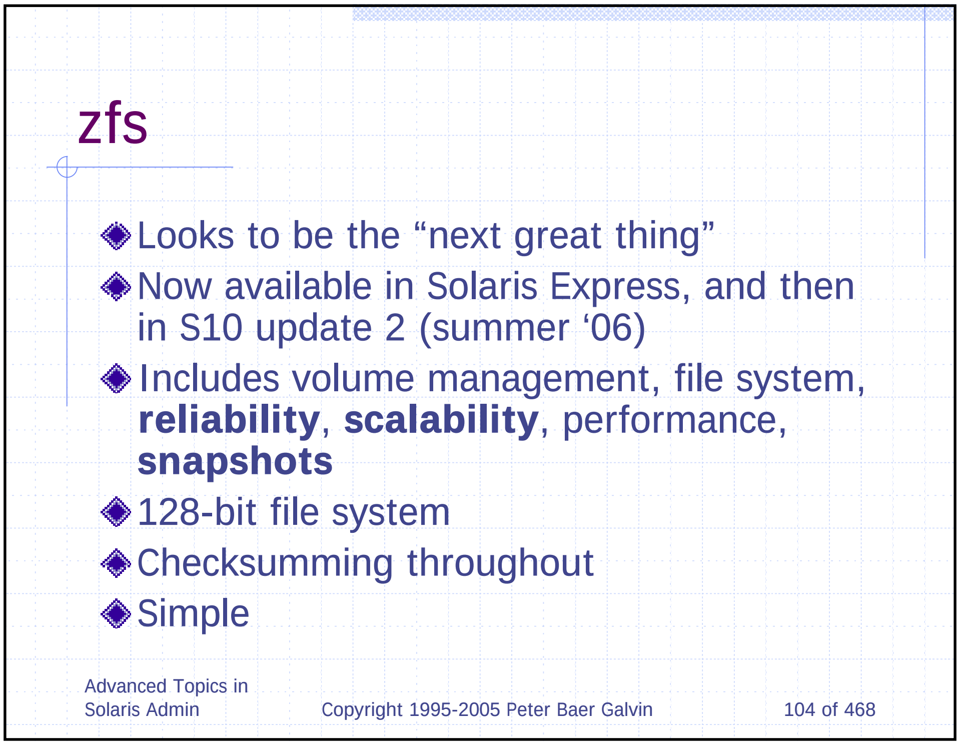


Zones and Packages (Cont.)

```
screen(intel) 4.0.2
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   86 package pathnames are already properly installed.

Installing screen as <SMCscreen>

## Installing part 1 of 1.
[ verifying class <none> ]

Installation of <SMCscreen> on zone <zone0> was
    successful.
## Restoring state of global zone <zone0>
```



Usability



zfs

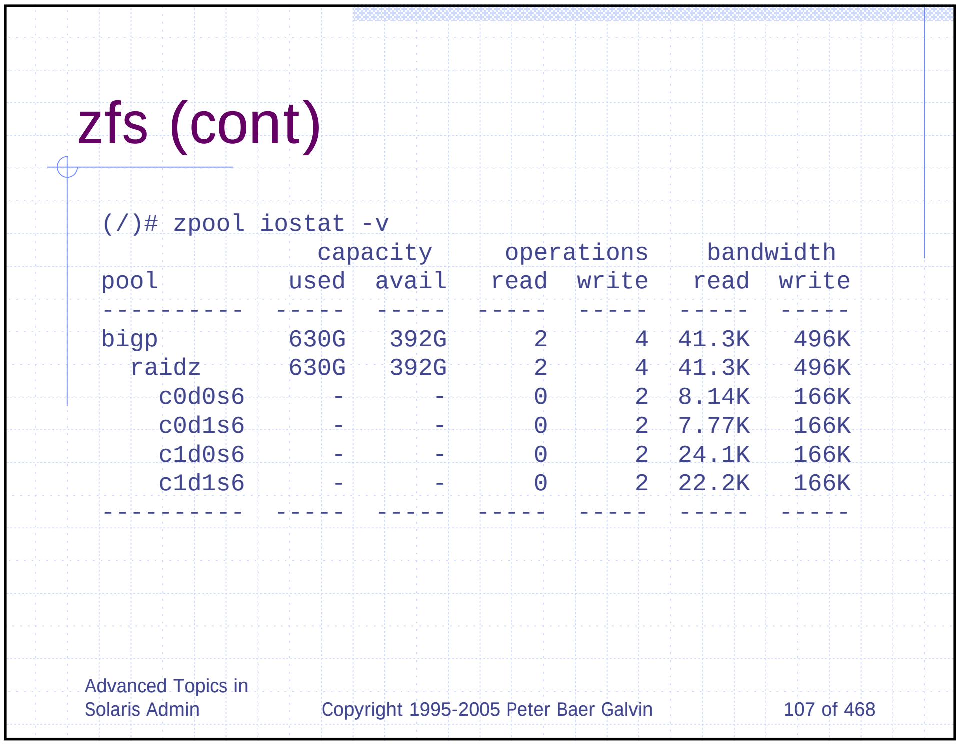
- Looks to be the “next great thing”
- Now available in Solaris Express, and then in S10 update 2 (summer ‘06)
- Includes volume management, file system, **reliability**, **scalability**, performance, **snapshots**
- 128-bit file system
- Checksumming throughout
- Simple

zfs (cont)

```
(/)# zpool
missing command
usage: zpool command args ...
where 'command' is one of the following:

        create  [-fn] [-R root] <pool> <vdev> ...
        destroy [-f] <pool>

        add [-fn] <pool> <vdev> ...

        list [-H] [-o field[,field]*] [pool] ...
        iostat [-v] [pool] ... [interval [count]]
        status [-vx] [pool] ...

        attach [-f] <pool> <device> <new_device>
        detach [-f] <pool> <device>
        replace [-f] <pool> <device> <new_device>

        online [-t] <pool> <device>
        offline [-ft] <pool> <device>

        import [-d dir]
        import [-d dir] [-f] [-o opts] [-R root] -a
        import [-d dir] [-f] [-o opts] [-R root ]<pool | id> [newpool]
        export [-f] <pool> ...
```

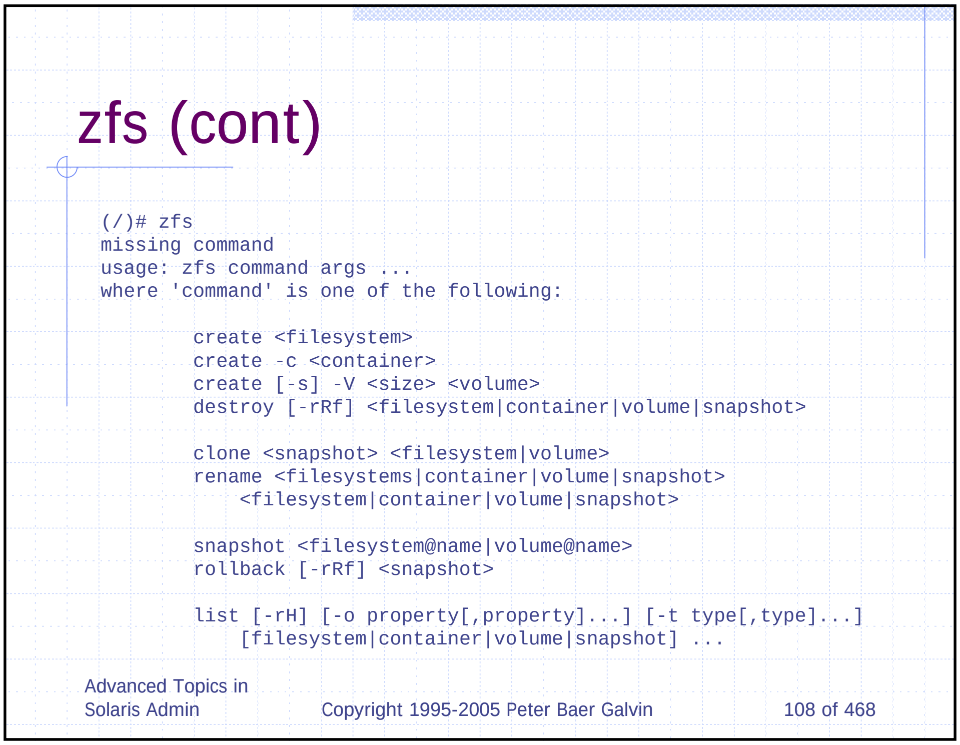
zfs (cont)

```
(/)# zpool status -v
  pool: bigp
 state: ONLINE
config:

        NAME                  STATE     READ WRITE CKSUM
        bigp                  ONLINE       0     0     0
          raidz               ONLINE       0     0     0
            c0d0s6            ONLINE       0     0     0
            c0d1s6            ONLINE       0     0     0
            c1d0s6            ONLINE       0     0     0
            c1d1s6            ONLINE       0     0     0
```



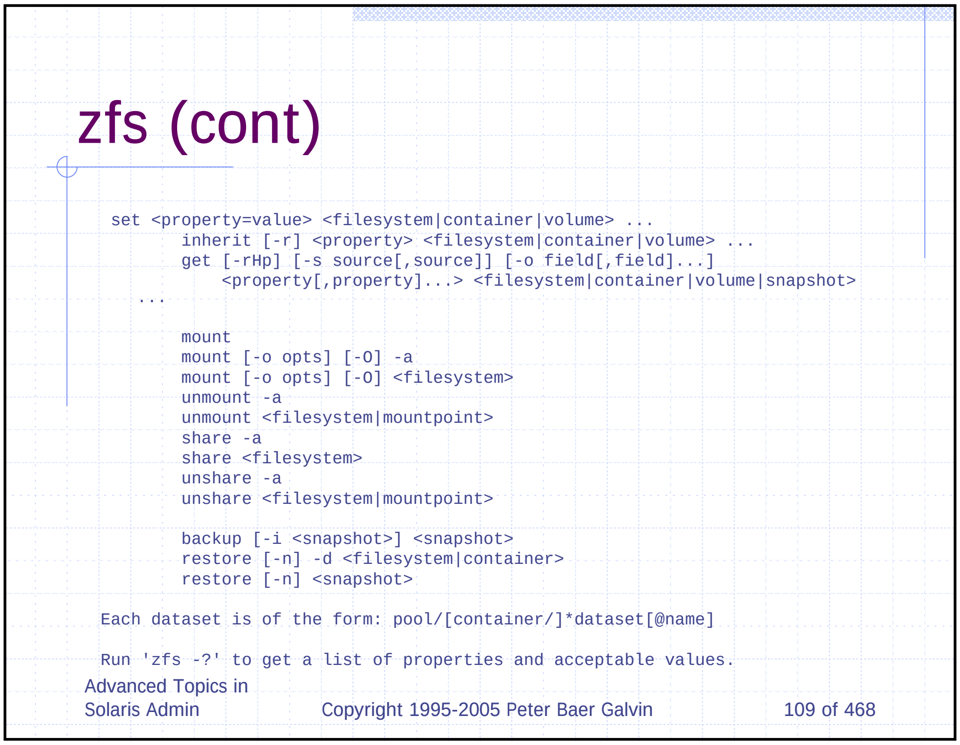
zfs (cont)

```
(/)# zpool iostat -v
                 capacity     operations    bandwidth
pool           used  avail   read  write   read  write
----------    -----  -----  -----  -----  -----  -----
bigp           630G   392G      2      4  41.3K   496K
  raidz        630G   392G      2      4  41.3K   496K
    c0d0s6        -      -      0      2  8.14K   166K
    c0d1s6        -      -      0      2  7.77K   166K
    c1d0s6        -      -      0      2  24.1K   166K
    c1d1s6        -      -      0      2  22.2K   166K
----------    -----  -----  -----  -----  -----  -----
```



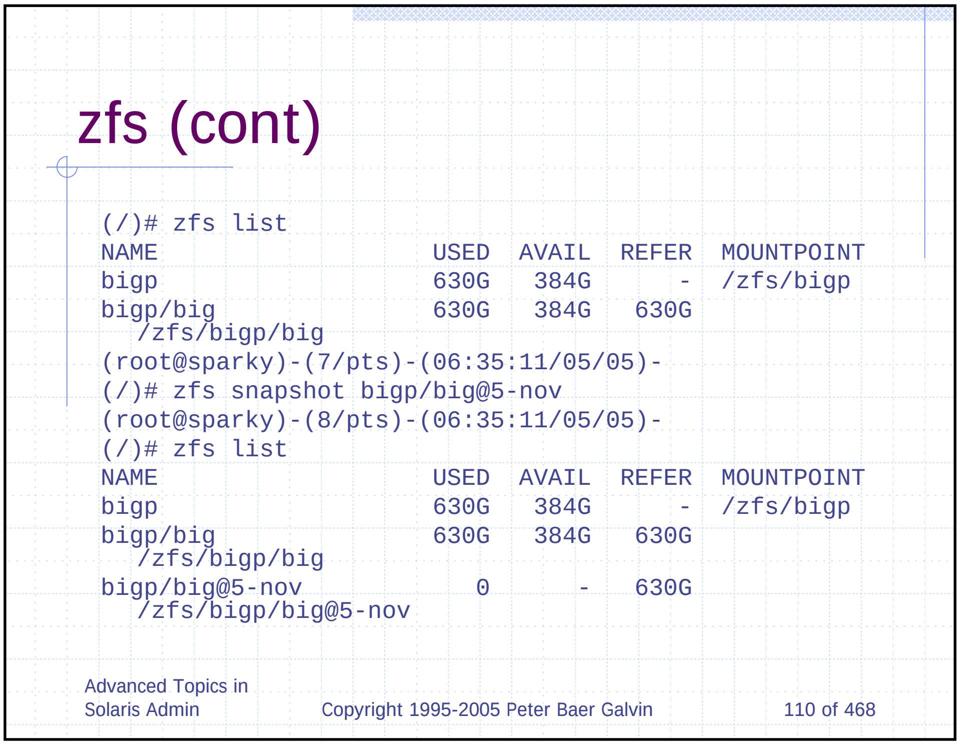
zfs (cont)

```
(/)# zfs
missing command
usage: zfs command args ...
where 'command' is one of the following:

        create <filesystem>
        create -c <container>
        create [-s] -V <size> <volume>
        destroy [-rRf] <filesystem|container|volume|snapshot>

        clone <snapshot> <filesystem|volume>
        rename <filesystems|container|volume|snapshot>
            <filesystem|container|volume|snapshot>

        snapshot <filesystem@name|volume@name>
        rollback [-rRf] <snapshot>

        list [-rH] [-o property[,property]...] [-t type[,type]...]
            [filesystem|container|volume|snapshot] ...
```



zfs (cont)

```
 set <property=value> <filesystem|container|volume> ...
        inherit [-r] <property> <filesystem|container|volume> ...
        get [-rHp] [-s source[,source]] [-o field[,field]...]
            <property[,property]...> <filesystem|container|volume|snapshot>
    ...

        mount
        mount [-o opts] [-O] -a
        mount [-o opts] [-O] <filesystem>
        unmount -a
        unmount <filesystem|mountpoint>
        share -a
        share <filesystem>
        unshare -a
        unshare <filesystem|mountpoint>

        backup [-i <snapshot>] <snapshot>
        restore [-n] -d <filesystem|container>
        restore [-n] <snapshot>

Each dataset is of the form: pool/[container/]*dataset[@name]

Run 'zfs -?' to get a list of properties and acceptable values.
```

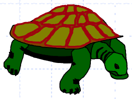


zfs (cont)

```
(/)# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
bigp                   630G   384G      -  /zfs/bigp
bigp/big               630G   384G   630G
   /zfs/bigp/big
(root@sparky)-(7/pts)-(06:35:11/05/05)-
(/)# zfs snapshot bigp/big@5-nov
(root@sparky)-(8/pts)-(06:35:11/05/05)-
(/)# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
bigp                   630G   384G      -  /zfs/bigp
bigp/big               630G   384G   630G
   /zfs/bigp/big
bigp/big@5-nov            0      -   630G
   /zfs/bigp/big@5-nov
```

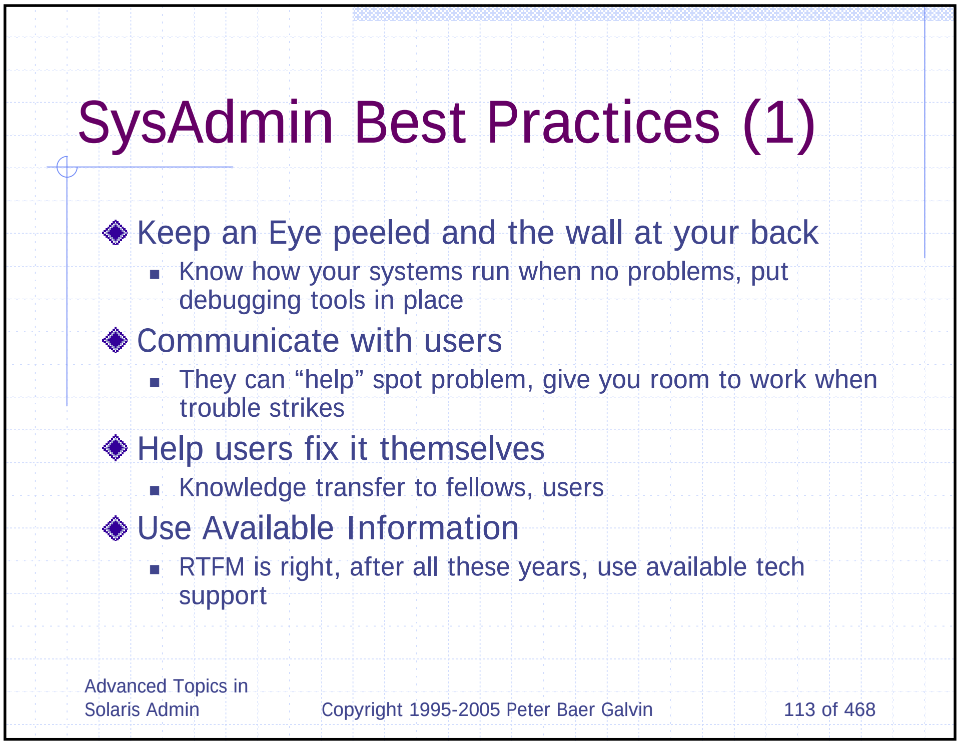
Philosophy

In the Liberal Arts Tradition



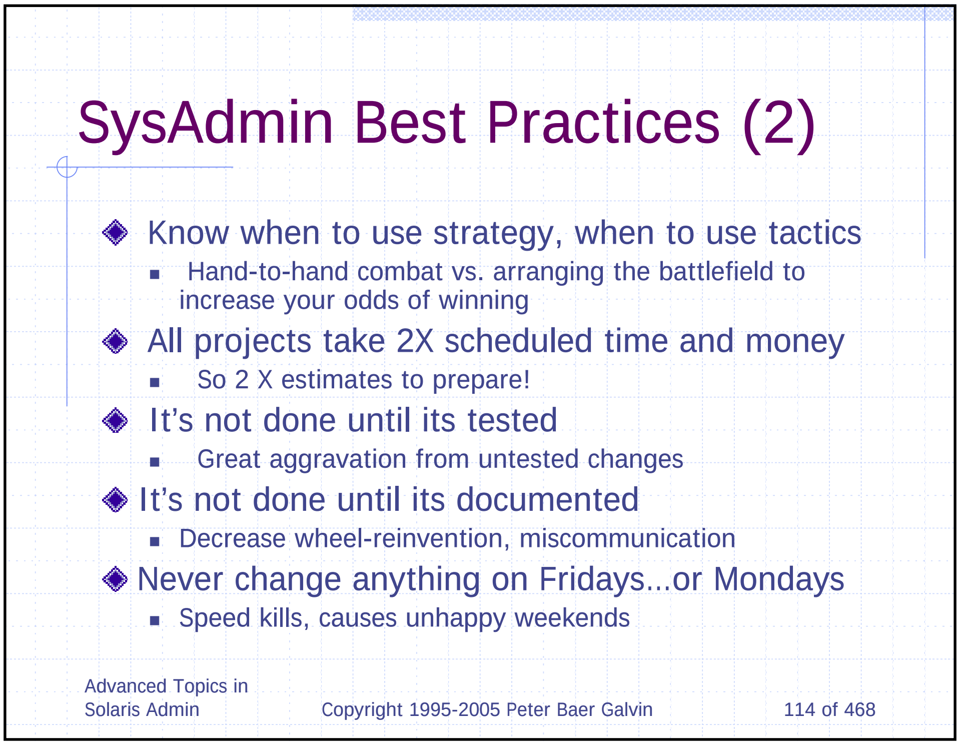
Topics

- System Administration Best Practices
  - From March 2003 SysAdmin Magazine column
    - Full version at end of tutorial material
  - Consensus administration best practices (Solaris and general) with contributions from many experienced sysadmins
  - Contribute at bestpractice@petergalvin.info



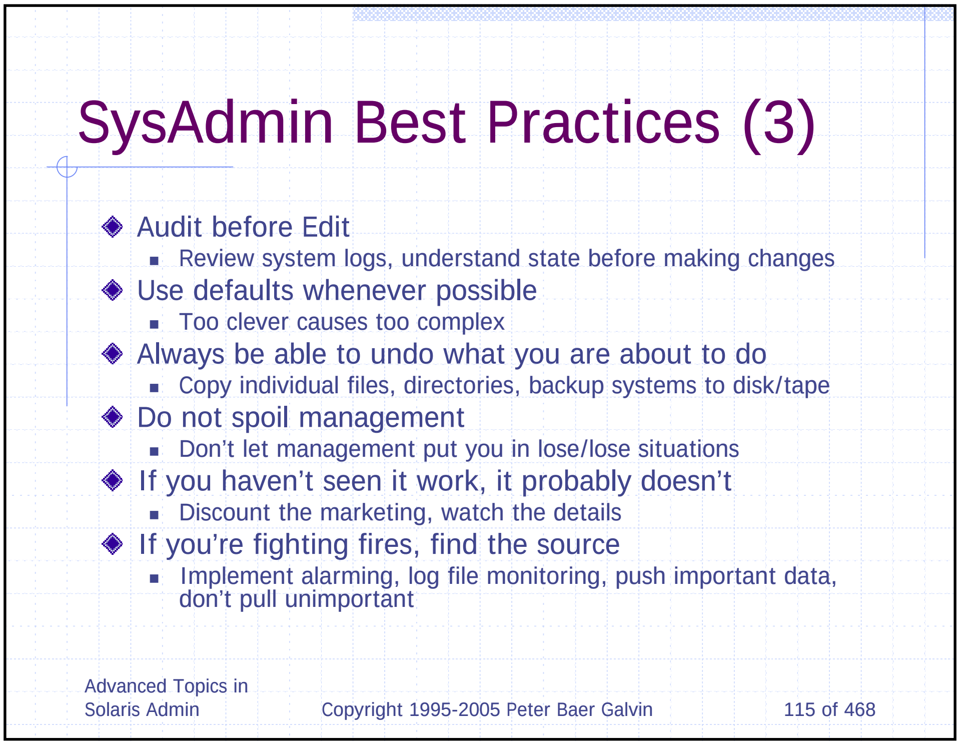
SysAdmin Best Practices (1)

- Keep an Eye peeled and the wall at your back
  - Know how your systems run when no problems, put debugging tools in place
- Communicate with users
  - They can “help” spot problem, give you room to work when trouble strikes
- Help users fix it themselves
  - Knowledge transfer to fellows, users
- Use Available Information
  - RTFM is right, after all these years, use available tech support



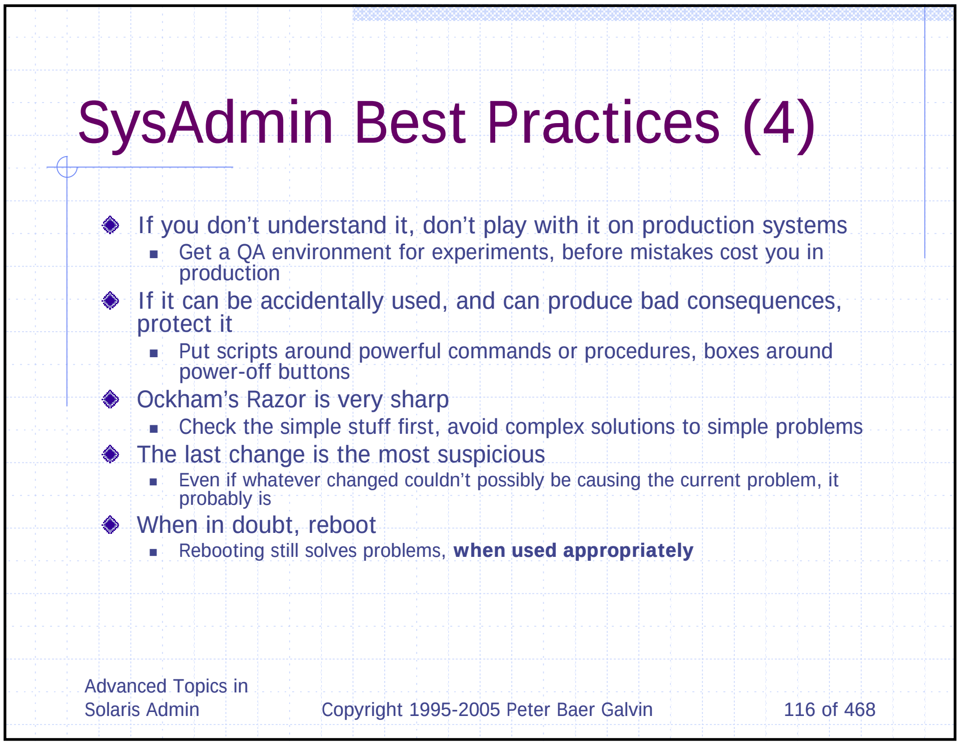
SysAdmin Best Practices (2)

- Know when to use strategy, when to use tactics
  - Hand-to-hand combat vs. arranging the battlefield to increase your odds of winning
- All projects take 2X scheduled time and money
  - So 2 X estimates to prepare!
- It’s not done until its tested
  - Great aggravation from untested changes
- It’s not done until its documented
  - Decrease wheel-reinvention, miscommunication
- Never change anything on Fridays...or Mondays
  - Speed kills, causes unhappy weekends



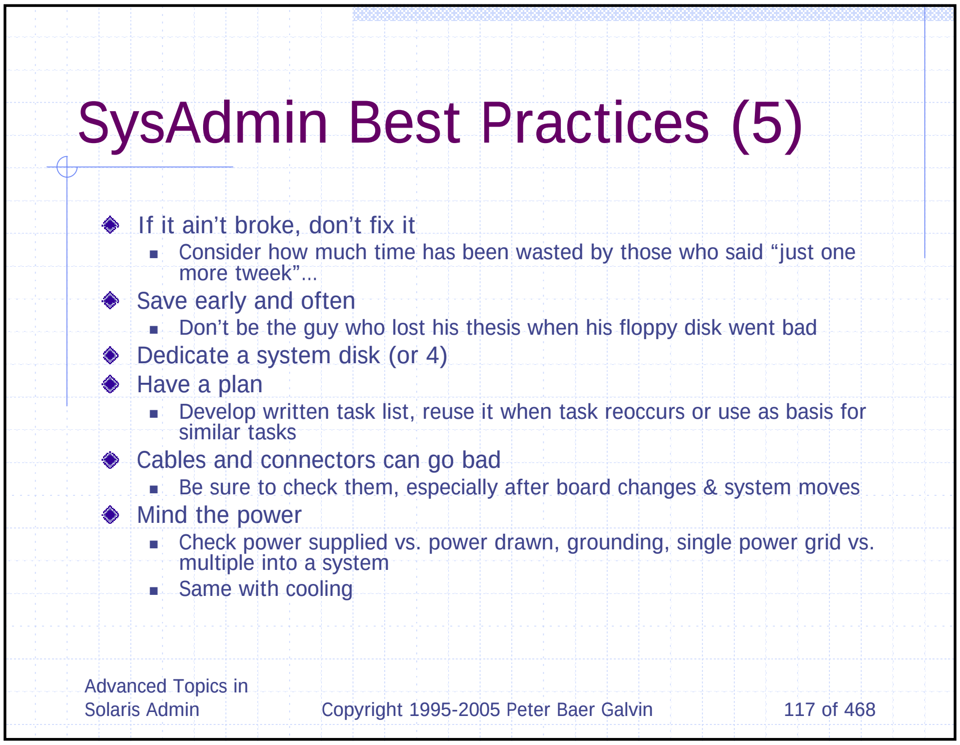
SysAdmin Best Practices (3)

- Audit before Edit
  - Review system logs, understand state before making changes
- Use defaults whenever possible
  - Too clever causes too complex
- Always be able to undo what you are about to do
  - Copy individual files, directories, backup systems to disk/tape
- Do not spoil management
  - Don’t let management put you in lose/lose situations
- If you haven’t seen it work, it probably doesn’t
  - Discount the marketing, watch the details
- If you’re fighting fires, find the source
  - Implement alarming, log file monitoring, push important data, don’t pull unimportant



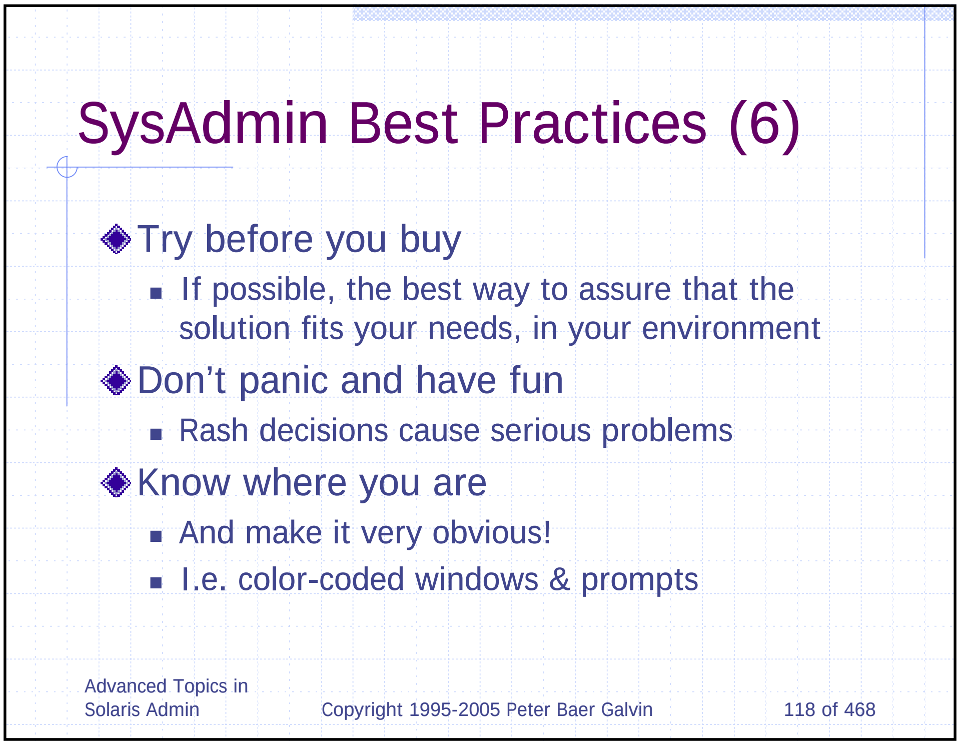
SysAdmin Best Practices (4)

- If you don’t understand it, don’t play with it on production systems
  - Get a QA environment for experiments, before mistakes cost you in production
- If it can be accidentally used, and can produce bad consequences, protect it
  - Put scripts around powerful commands or procedures, boxes around power-off buttons
- Ockham’s Razor is very sharp
  - Check the simple stuff first, avoid complex solutions to simple problems
- The last change is the most suspicious
  - Even if whatever changed couldn’t possibly be causing the current problem, it probably is
- When in doubt, reboot
  - Rebooting still solves problems, **when used appropriately**



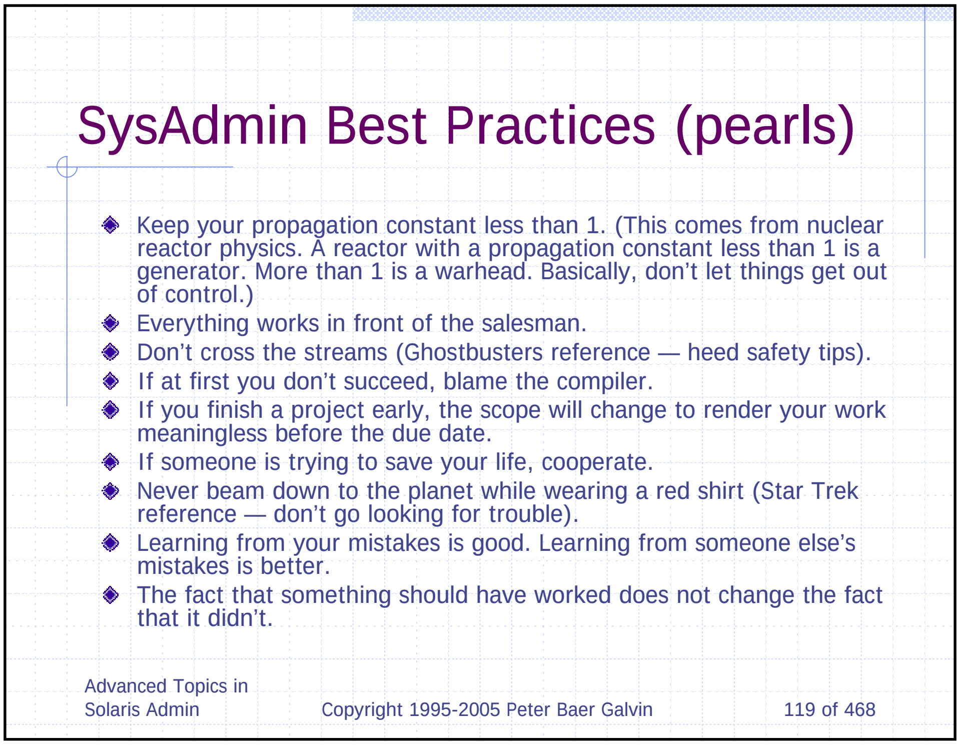
SysAdmin Best Practices (5)

- If it ain’t broke, don’t fix it
  - Consider how much time has been wasted by those who said “just one more tweek”…
- Save early and often
  - Don’t be the guy who lost his thesis when his floppy disk went bad
- Dedicate a system disk (or 4)
- Have a plan
  - Develop written task list, reuse it when task reoccurs or use as basis for similar tasks
- Cables and connectors can go bad
  - Be sure to check them, especially after board changes & system moves
- Mind the power
  - Check power supplied vs. power drawn, grounding, single power grid vs. multiple into a system
  - Same with cooling



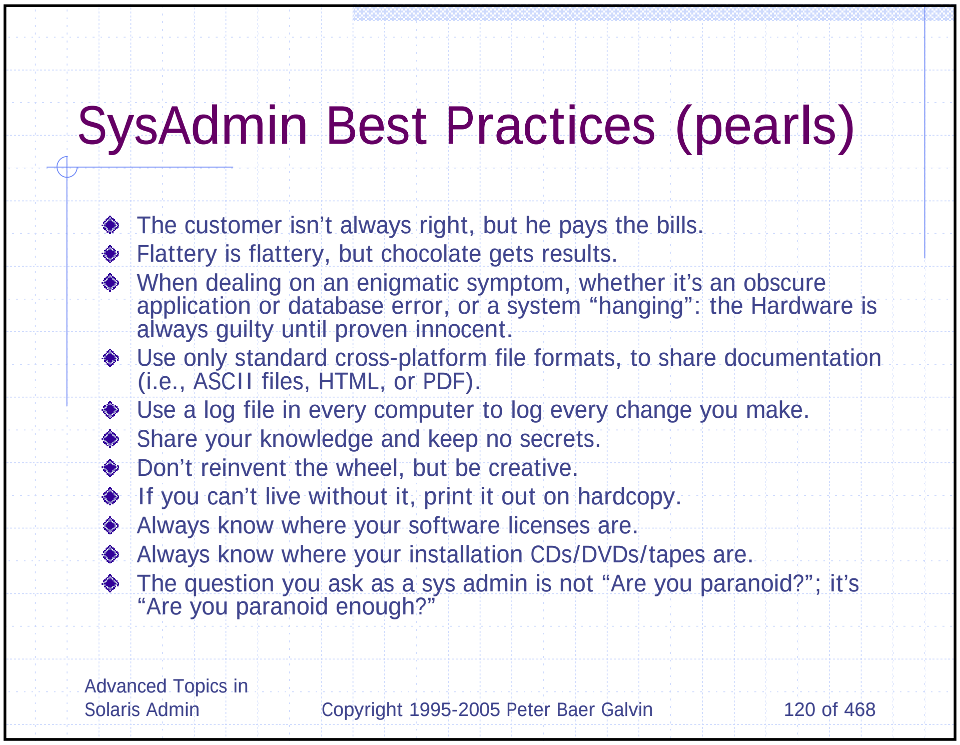
SysAdmin Best Practices (6)

- Try before you buy
  - If possible, the best way to assure that the solution fits your needs, in your environment
- Don’t panic and have fun
  - Rash decisions cause serious problems
- Know where you are
  - And make it very obvious!
  - I.e. color-coded windows & prompts



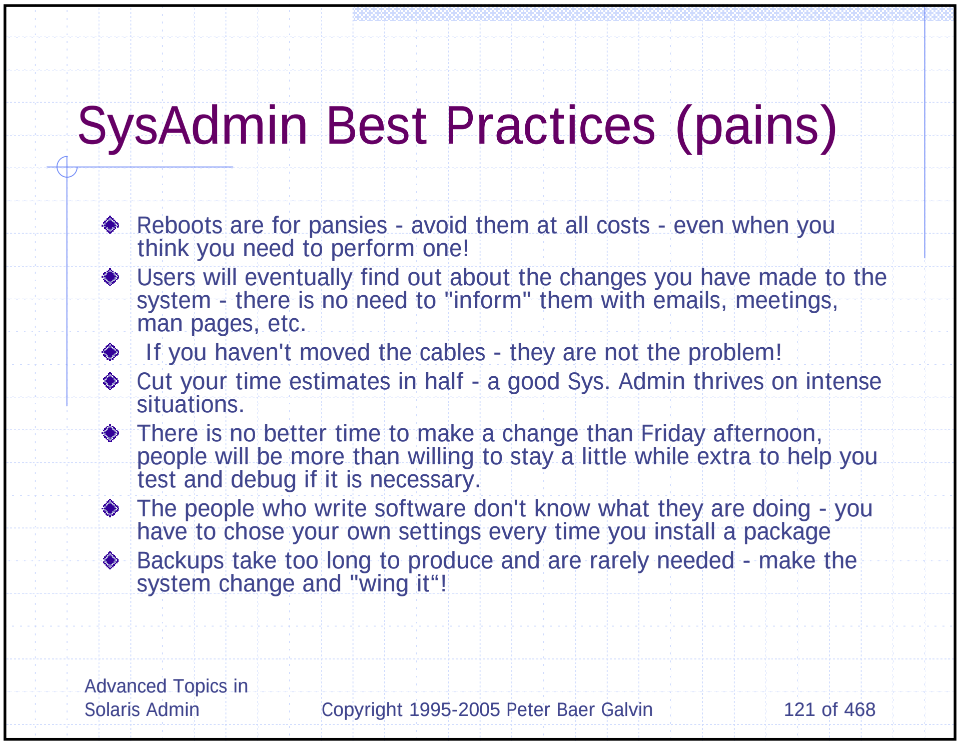
SysAdmin Best Practices (pearls)

- Keep your propagation constant less than 1. (This comes from nuclear reactor physics. A reactor with a propagation constant less than 1 is a generator. More than 1 is a warhead. Basically, don’t let things get out of control.)
- Everything works in front of the salesman.
- Don’t cross the streams (Ghostbusters reference — heed safety tips).
- If at first you don’t succeed, blame the compiler.
- If you finish a project early, the scope will change to render your work meaningless before the due date.
- If someone is trying to save your life, cooperate.
- Never beam down to the planet while wearing a red shirt (Star Trek reference — don’t go looking for trouble).
- Learning from your mistakes is good. Learning from someone else’s mistakes is better.
- The fact that something should have worked does not change the fact that it didn’t.



SysAdmin Best Practices (pearls)

- The customer isn’t always right, but he pays the bills.
- Flattery is flattery, but chocolate gets results.
- When dealing on an enigmatic symptom, whether it’s an obscure application or database error, or a system “hanging”: the Hardware is always guilty until proven innocent.
- Use only standard cross-platform file formats, to share documentation (i.e., ASCII files, HTML, or PDF).
- Use a log file in every computer to log every change you make.
- Share your knowledge and keep no secrets.
- Don’t reinvent the wheel, but be creative.
- If you can’t live without it, print it out on hardcopy.
- Always know where your software licenses are.
- Always know where your installation CDs/DVDs/tapes are.
- The question you ask as a sys admin is not “Are you paranoid?”; it’s “Are you paranoid enough?”



SysAdmin Best Practices (pains)

- Reboots are for pansies - avoid them at all costs - even when you think you need to perform one!
- Users will eventually find out about the changes you have made to the system - there is no need to "inform" them with emails, meetings, man pages, etc.
- If you haven't moved the cables - they are not the problem!
- Cut your time estimates in half - a good Sys. Admin thrives on intense situations.
- There is no better time to make a change than Friday afternoon, people will be more than willing to stay a little while extra to help you test and debug if it is necessary.
- The people who write software don't know what they are doing - you have to chose your own settings every time you install a package
- Backups take too long to produce and are rarely needed - make the system change and "wing it“!