Solaris 10 Security

Technical Deep Dive

- **Glenn Brunette**
  - Distinguished Engineer
- Sun Microsystems, Inc.

Agenda

- **Solaris Security Goals**
- **Solaris 9 Security Review**
  - an overview of features from past releases.
- **Solaris 10 Security Deep Dive**
  - a dive into new features including: Secure by Default, SMF, Privileges, ZFS, Zones, Trusted Extensions, and more!

Solaris Security Goals

- **Defending**
  - Provide strong assurance of system integrity.
  - Defend system from unauthorized access.
- **Enabling**
  - Secure authentication of all active subjects.
  - Protect communications between endpoints.
- **Deploying**
  - Emphasize an integratable stack architecture.
  - Interoperate with other security architectures.
  - Ease management and use of security features.
  - Receive independent assessment of security.

Solaris 9 Security Review

- Access Control Lists
- Role-based Access Control
- IPsec / IKE
- Solaris Auditing
- TCP Wrappers (inetd, rpcbind)
- Flexible Crypt
- Signed Patches
- Granular Packaging
- SSL-enabled LDAP
- WAN Boot
- IKE Hardware Accel.
- Solaris Fingerprint DB
- Solaris Secure Shell
- Kerberos
- /dev/[u]random
- Enhanced PAM Framework
- Smartcard Framework
- Java Security
- SunScreen 3.2
- Solaris Security Toolkit
- sadmind DES Auth
- LDAP Password Management

-
- Solaris 10 Technical Security Deep Dive

Reduced Networking Metacluster

Metacluster	Size (MB)	# Pkgs	# Set-UID	# Set-GID
Reduced Networking SUNWCrnet	321	147	31	12
Core SUNWCreq	352	206	38	13
End User SUNWCuser	2400	772	69	20
Developer SUNWCprog	3000	1017	70	20
Entire SUNWCall	3100	1074	84	21
Entire + OEM SUNWCXall	3100	1075	84	21

Reduced and Minimal Configurations

- Some environments remove or simply do not install software packages that are not needed (business or technical reasons)
  - Less software to install, upgrade, patch, and maintain.
  - Less software equates to reduced exposure to security vulnerabilities.
- Refer to Sun's Rules of Engagement for the Support of Reduced or Minimal Configurations
  - http://www.opensolaris.org/os/community/security/files/minimization-support-rules-ext.pdf
- Solaris Package Companion can be used to understand software package relationships and dependencies
  - http://www.opensolaris.org/os/project/svr4_packaging/package_companion/

Solaris Package Companion Examples

- **EXAMPLE 1: What packages depend on StarOffice?**

```
•$ spc-v0.8.ksh -r ./nv72.rep -l -F -f /opt/staroffice8/program/soffice
SUNWCstaroffice
```

```
•$ spc-v0.8.ksh -r ./nv72.rep -F -Z -v SUNWCstaroffice
SUNWCstaroffice                    [C] StarOffice
```

- **EXAMPLE 2: On what does SSH depend?**

```
•$ spc-v0.8.ksh -r ./nv72.rep -D -F -v SUNWCssh
SUNWCcs                            [C] Core Solaris
SUNWCfwcmp                         [C] Freeware Compression Utilities
SUNWCopenssl                       [C] OpenSSL
SUNWCssh                           [C] Secure Shell
SUNWcakr                           [P] Core Solaris Kernel Architecture (Root)
SUNWcar                            [P] Core Architecture, (Root)
SUNWgss                            [P] GSSAPI V2
SUNWgssc                           [P] GSSAPI CONFIG V2
SUNWkvm                            [P] Core Architecture, (Kvm)
SUNWloc                            [P] System Localization
```

- For more details and information, see the Solaris Package Companion OpenSolaris Project site at: http://opensolaris.org/os/project/svr4_packaging/package_companion/

Cryptographically Signed ELF Objects

- ELF Objects Cryptographically Signed
  - binaries, libraries, kernel modules, crypto modules, etc.

```
•# file /usr/lib/ssh/sshd
/usr/lib/ssh/sshd:      ELF 32-bit MSB
executable
SPARC Version 1, dynamically linked, stripped

# elfsign verify -e /usr/lib/ssh/sshd
elfsign: verification of /usr/lib/ssh/sshd
passed.

# elfsign list -f signer -e /usr/bin/ls
CN=SunOS 5.10, OU=Solaris Signed Execution,
O=Sun Microsystems Inc
```

- Cryptographic modules must be signed by Sun.
  - Signature must be validated before module can be loaded.
  - Crypto modules will not load if not signed or have invalid

Non-Executable Stack Example

```
•#include <stdio.h>
#include <string.h>

•typedef void (*fptr)(void);


•#ifdef __sparc
char shellcode[] =
"\x2d\x0b\xd8\x9a\xac\x15\xa1\x6e\x2f\x0b\xdc\xda\x90\x0b\x80\x0e"
"\x92\x03\xa0\x08\x94\x1a\x80\x0a\x9c\x03\xa0\x10\xec\x3b\xbf\xf0"
"\xdc\x23\xbf\xf8\xc0\x23\xbf\xfc\x82\x10\x20\x3b\x91\xd0\x20\x08";
#endif

•int
main(int argc, char **argv)
{
        fptr f;
        char code[100];

        memcpy(code, shellcode, sizeof(shellcode));
        printf("Attempting to start a shell...\n");
        f = (fptr)code;
        f();
        return (0);
}
```

Non-Executable Stack #1

```
•$ cc -o myshell shell.c
$ cc -o myshell-nx -M /usr/lib/ld/map.noexst shell.c

$ ./myshell
Attempting to start a shell...
$ exit

$ ./myshell-nx
Attempting to start a shell...
Segmentation Fault(coredump)

Sep 16 15:06:06 kilroy genunix: [ID 533030
kern.notice]
NOTICE: shell-noexstk[23132] attempt to execute code
on
stack by uid 101
```

>Stacks can be globally configured to be non-executable using the noexec_user_stack tunable in /etc/system.

Non-Executable Stack #2

```
$ telnet victimhost myshell
Trying 10.8.22.39...
Connected to victimhost.
Escape character is '^]'.
finger;
Login       Name               TTY         Idle    When    Where
gbrunett Glenn Brunette        pts/5            Wed 13:48  void
\377\277\375\364: ^M: not found
[...]
Connection to victimhost closed.
```

```
$ telnet victimhost myshell-nx
Trying 10.8.22.39...
Connected to victimhost.
Escape character is '^]'.
Connection to victimhost closed by foreign host.
```

- For more information on Solaris non-executable stack functionality, see: http://blogs.sun.com/gbrunett/tags/noexstk

Service Management Facility

- Provide a uniform mechanism to disable/manage services.
  - e.g., `svcadm [disable|enable] telnet`
- Support alternative service profiles
  - e.g., “Secure by Default” profile (since Solaris 10 11/06)
- Leverage authorizations to manage/configure services.
- Define context to permit services to be started as a specific user and group and with specific privileges.
- Support automatic service dependency resolution.
  - e.g., `svcadm enable -r nfs/client`
- Facilitate delegated service restarts.

SMF Example #1

```
•$ profiles
Service Operator
Basic Solaris User
All

•$ svcs network/inetd
STATE          STIME    FMRI
online          1:28:15 svc:/network/inetd:default

•$ svcadm disable network/inetd

•$ svcs -x -v network/inetd
svc:/network/inetd:default (inetd)
 State: disabled since Thu Jul 13 17:05:36 2008
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: man -M /usr/share/man -s 1M inetd
   See: /var/svc/log/network-inetd:default.log
Impact: 5 dependent services are not running:
```

SMF Example #2

```
•# svcprop -v -p defaults inetd
defaults/bind_addr astring ""
defaults/bind_fail_interval integer -1
defaults/bind_fail_max integer -1
defaults/con_rate_offline integer -1
[...]
defaults/stability astring Evolving
defaults/tcp_trace boolean false
defaults/tcp_wrappers boolean false
```

```
•# svcprop -p config/local_only rpc/bind
false
```

```
•# svcs -x sendmail
svc:/network/smtp:sendmail (sendmail SMTP mail transfer
agent)
 State: maintenance since Wed Dec 01 01:31:35 2007
Reason: Start method failed repeatedly, last exited with
status 208.
   See: http://sun.com/msg/SMF-8000-KS
   See: sendmail(1M)
Impact: 0 services are not running.
```

SMF Access Control

- Integrated with Solaris Roles (Rights Profiles)
  - *Service Administrator*
  - *Service Operator*
- Integrated with Solaris Authorizations
  - *Global:* *solaris.smf.modify*
  - Global: *solaris.smf.manage*
  - Per Service: *action_authorization*
- Services may have property-group specific authorizations
  - *value_authorization* – change existing property values
  - *modify_authorization* – add, modify, or delete properties

SMF Example #3

```
•# svcprop -p httpd -p general apache2
general/enabled boolean false
general/action_authorization astring sunw.apache.oper
general/entity_stability astring Evolving
httpd/ssl boolean false
httpd/stability astring Evolving
httpd/value_authorization astring sunw.apache.admin
```

•

- Example taken from the Sun BluePrint: Restricting Service Administration in the Solaris 10 Operating System, http://www.sun.com/blueprints/0605/819-2887.pdf

SMF Execution Context

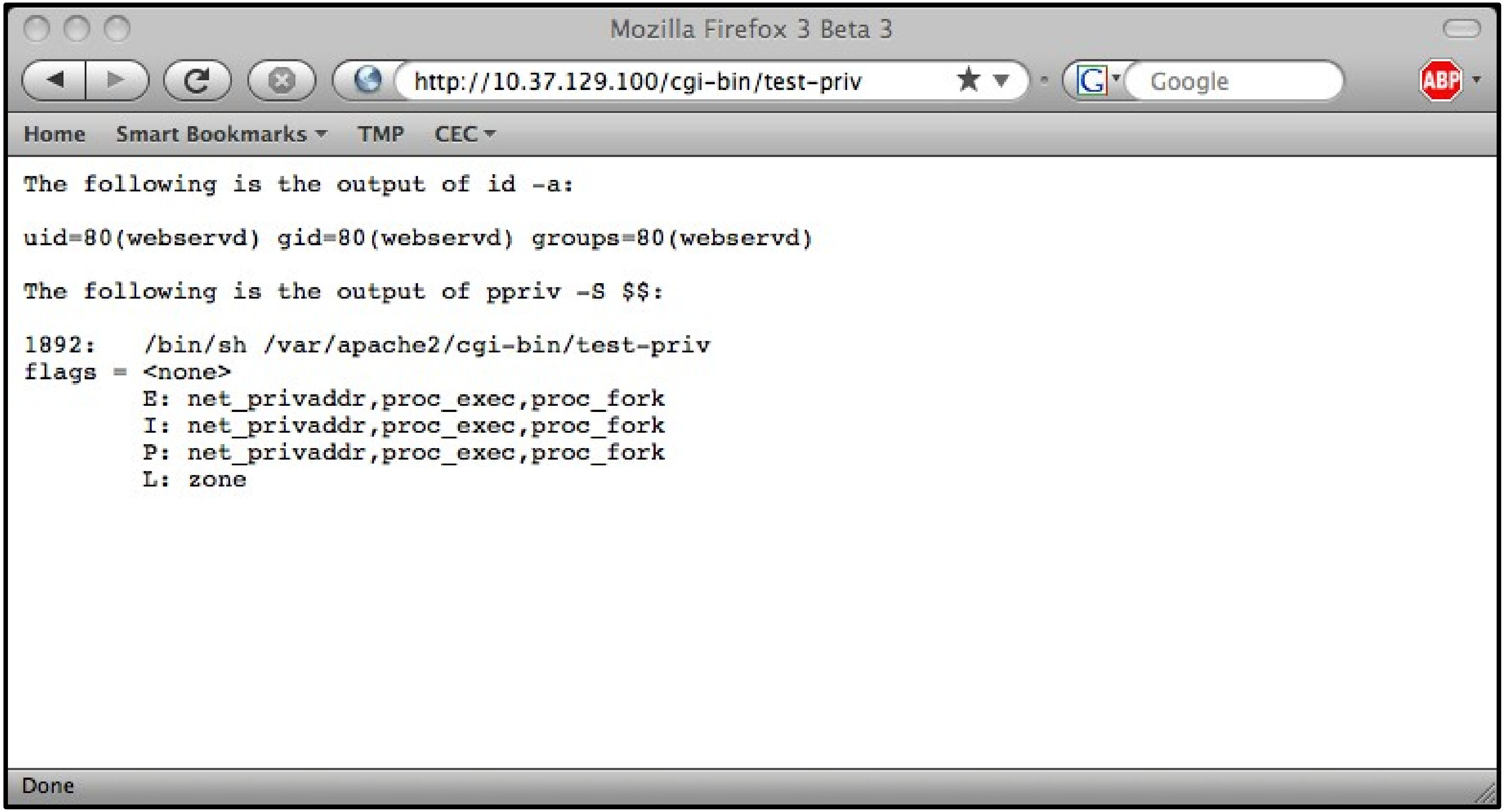
- `exec` methods can be forced to run as a given user:
  - `{start, stop, etc.}/user`
- `exec` methods can be forced to run as a given group:
  - `{start, stop, etc.}/group`
- `exec` methods can be forced to use specific privileges:
  - `{start, stop, etc.}/privileges`
  - `{start, stop, etc.}/limit_privileges`
- Other `exec` context can also be defined:
  - default project and resource pool, supplemental groups, etc.

SMF Example #4

```
•# svcprop -v -p start apache2
start/exec astring /lib/svc/method/http-apache2\ start
start/timeout_seconds count 60
start/type astring method
start/user astring webservd
start/group astring webservd
start/privileges astring basic,!proc_session,!
proc_info,!file_link_any,
net_privaddr
start/limit_privileges astring :default
start/use_profile boolean false
start/supp_groups astring :default
start/working_directory astring :default
start/project astring :default
start/resource_pool astring :default
```

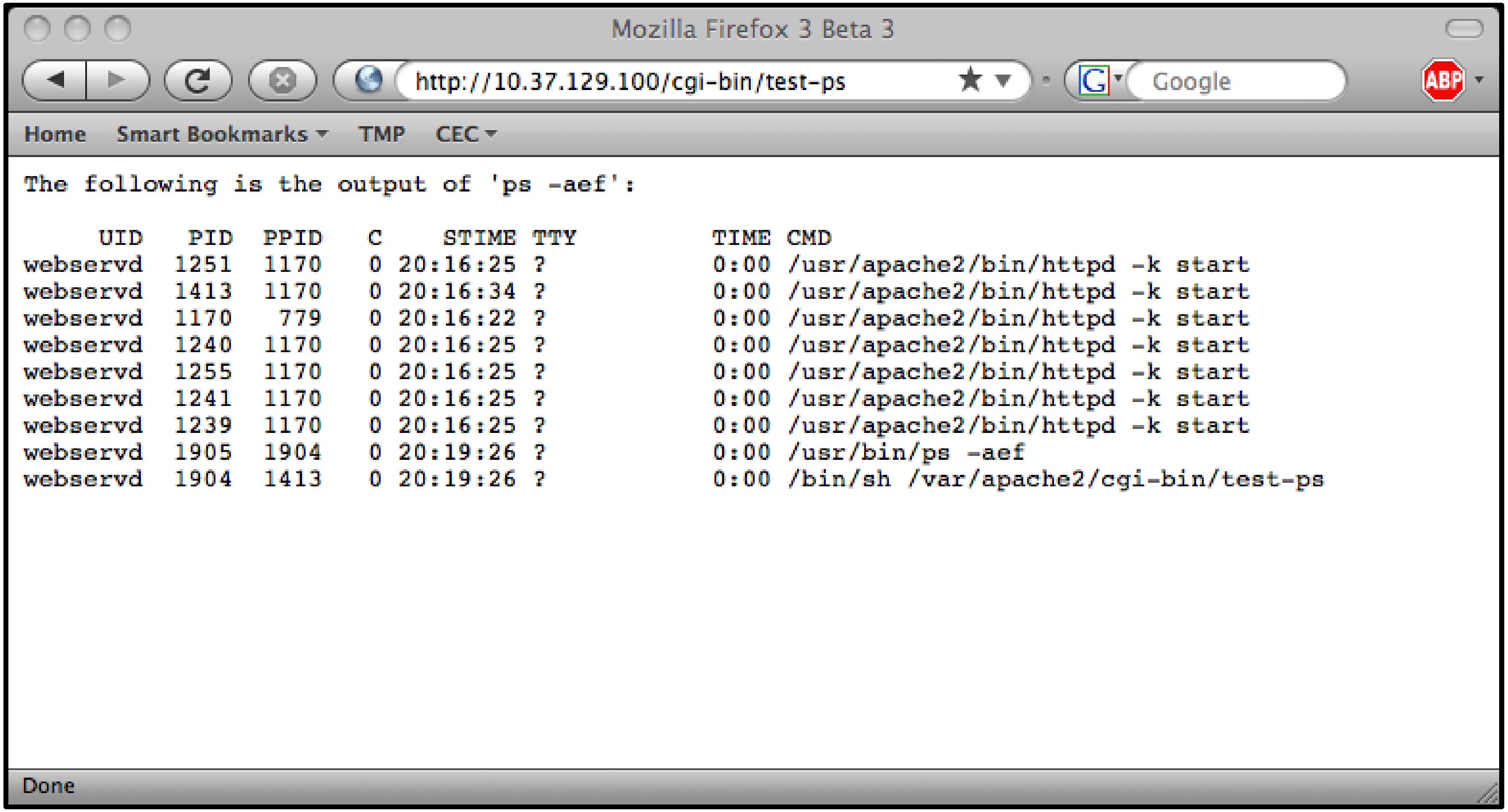
- Example taken from the Sun BluePrint: Limiting Service Privileges in the Solaris 10 Operating System, http://www.sun.com/blueprints/0505/819-2680.pdf

SMF Example #5



```
The following is the output of id -a:

uid=80(webservd) gid=80(webservd) groups=80(webservd)

The following is the output of ppriv -S $$:

1892:   /bin/sh /var/apache2/cgi-bin/test-priv
flags = <none>
        E: net_privaddr,proc_exec,proc_fork
        I: net_privaddr,proc_exec,proc_fork
        P: net_privaddr,proc_exec,proc_fork
        L: zone
```

SMF Example #6



The following is the output of 'ps -aef':

```
     UID   PID  PPID   C    STIME TTY         TIME CMD
webservd  1251  1170   0 20:16:25 ?           0:00 /usr/apache2/bin/httpd -k start
webservd  1413  1170   0 20:16:34 ?           0:00 /usr/apache2/bin/httpd -k start
webservd  1170   779   0 20:16:22 ?           0:00 /usr/apache2/bin/httpd -k start
webservd  1240  1170   0 20:16:25 ?           0:00 /usr/apache2/bin/httpd -k start
webservd  1255  1170   0 20:16:25 ?           0:00 /usr/apache2/bin/httpd -k start
webservd  1241  1170   0 20:16:25 ?           0:00 /usr/apache2/bin/httpd -k start
webservd  1239  1170   0 20:16:25 ?           0:00 /usr/apache2/bin/httpd -k start
webservd  1905  1904   0 20:19:26 ?           0:00 /usr/bin/ps -aef
webservd  1904  1413   0 20:19:26 ?           0:00 /bin/sh /var/apache2/cgi-bin/test-ps
```

Solaris Secure By Default

- Only Secure Shell is reachable by default.
  - `root` use of Secure Shell is not permitted by default.
- Existing services are configured in SMF to either be:
  - Disabled by default
  - Listening for local (e.g., loopback) connections only
- Configuration can be selected using CLI or JumpStart:
  - `netservices`: `open` (traditional) or `limited` (SBD)
  - `service_profile`: `open` or `limited_net`
- Default installation method in Nevada/OpenSolaris:
  - Solaris upgrades are not changed or impacted.
  - Solaris 10 initial (fresh) installations can select SBD mode.

Solaris Secure By Default Example #1

```
# netservices
netservices: usage: netservices [ open | limited ]
```

```
# netservices limited
restarting syslogd
restarting sendmail
dtlogin needs to be restarted. Restart now? [Y] y
restarting dtlogin
```

```
# netstat -af inet -P tcp | grep LISTEN
[...]
*.sunrpc             *.*     0    0 49152      0 LISTEN
*.ssh                *.*     0    0 49152      0 LISTEN
localhost.smtp       *.*     0    0 49152      0 LISTEN
localhost.submission *.*     0    0 49152      0 LISTEN
```

Solaris Secure By Default Example #2

Service	FMRI	Property	Values
rpcbind	svc:/network/rpc/bind	config/local_only	**true**, false
syslog	svc:/system/system-log	config/log_from_remote	true, **false**
sendmail	svc:/network/smtp:sendmail	config/local_only	**true**, false
smcwebserver	svc:/system/webconsole:console	options/tcp_listen	true, **false**
wbem	svc:/application/management/wbem	options/tcp_listen	true, **false**
X11	svc:/application/x11/x11-server	options/tcp_listen	true, **false**
CDE	svc:/application/graphical-login/cde-login	dtlogin/args	[null], **-udpPort 0**
ToolTalk	svc:/network/rpc/cde-ttdbserver:tcp	proto	tcp, **ticotsord**
calendar	svc:/network/rpc/cde-calendar-manager	proto	tcp, **ticlts**
BSD printing	svc:/application/print/rfc1179:default	bind_addr	[null], **localhost**

User/Password Management

- Enforced for All Naming Services
  - > Password Complexity Checks
    - > Login Name != Password
    - > White Space Permitted
    - > Minimum Characters by Class
      - – Alphabetic, Non-Alphabetic, Uppercase, Lowercase, Digits, Special
    - > Maximum Consecutive Repeating Characters
  - > Local Banned Password List (Dictionary)
- Enforced for “files” Naming Service Only
  - > Local Password History
  - > Local Account Lockout (3 Strikes)
- New “Account Locked” Semantics

Password Management Example

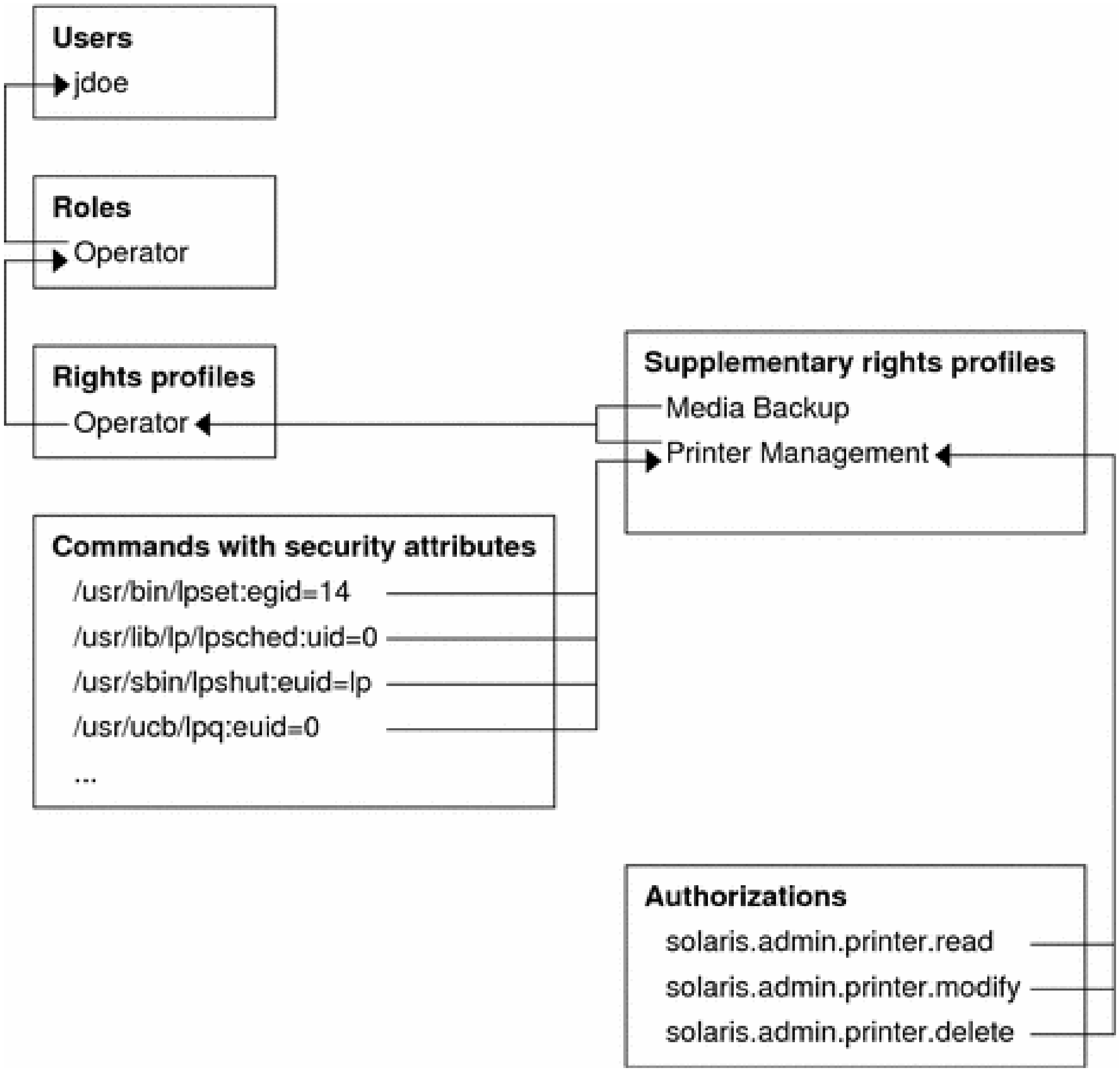
```
•$ passwd gbrunett
Enter existing login password:
New Password:
passwd: The password must contain at least 1 numeric or special
character(s).

•Please try again
New Password:
passwd: The password must contain at least 1 uppercase alpha
character(s).

•Please try again
New Password:
passwd: Too many consecutively repeating characters. Maximum allowed is
3.
Permission denied

•$ passwd gbrunett
Enter existing login password:
New Password:
passwd: Password in history list.
```

User Rights Management (Roles)

- Solaris Users versus Roles
  - Roles can only be accessed by users already logged in.
  - Users cannot assume a role unless authorized.

```
$ id -a
uid=80(webservd) gid=80(webservd)

$ roles
No roles

$ su - root
Password:
Roles can only be assumed by authorized users
su: Sorry
```

User Rights Management (Rights)



Users
jdoe

Roles
Operator

Rights profiles
Operator

Supplementary rights profiles
Media Backup
Printer Management

Commands with security attributes
/usr/bin/lpset:egid=14
/usr/lib/lp/lpsched:uid=0
/usr/sbin/lpshut:euid=lp
/usr/ucb/lpq:euid=0
...

Authorizations
solaris.admin.printer.read
solaris.admin.printer.modify
solaris.admin.printer.delete

User Rights Management Example #1

```
•$ profiles -l

•      Object Access Management:

•          /usr/bin/chgrp    privs=file_chown
          /usr/bin/chmod    privs=file_owner
          [...]

•      [...]

•$ ls -ld mnt
drwxr-xr-x   2 gbrunett gbrunett     512 Nov  7 12:54 mnt

•$ chown bin:bin mnt
chown: mnt: Not owner

•$ pfexec chown bin:bin mnt

•$ ls -ld mnt
drwxr-xr-x   2 bin      bin          512 Nov  7 12:54 mnt
```

User Rights Management Example #2

```
•# svcprop -p httpd -p general apache2
general/enabled boolean false
general/action_authorization astring sunw.apache.oper
general/entity_stability astring Evolving
httpd/ssl boolean false
httpd/stability astring Evolving

# auths weboper
sunw.apache.oper

# profiles -l weboper

•      Apache Operator:
           /usr/sbin/svcadm
           /usr/bin/svcs

•

•
```

User Rights Management Example #3

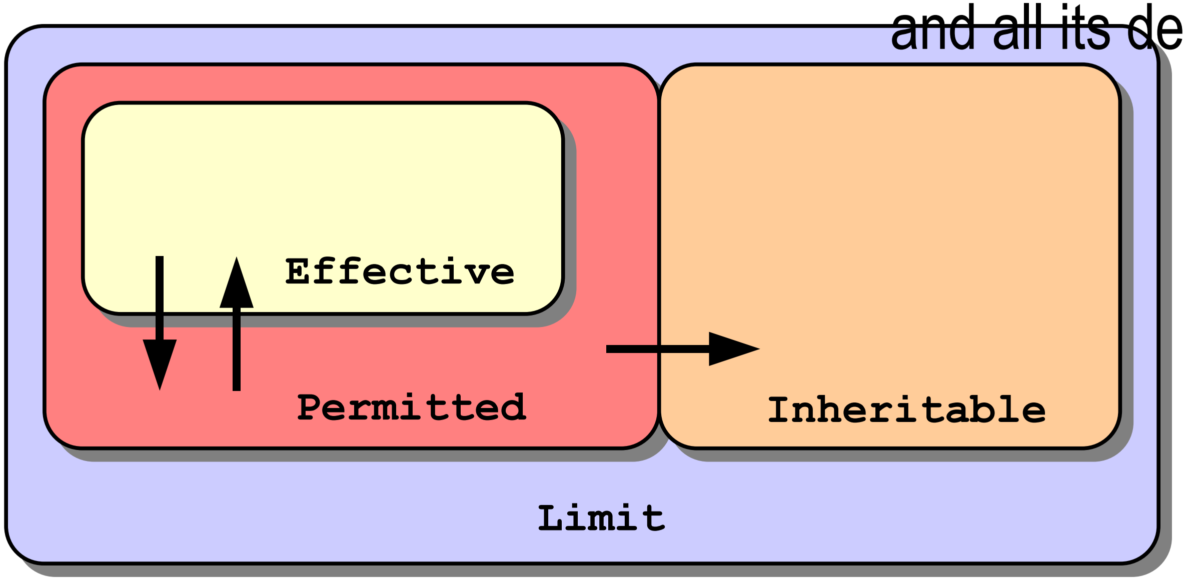
```
•$ svcs -o state,ctid,fmri apache2
STATE             CTID   FMRI
online           91050   svc:/network/http:apache2

•$ svcadm restart apache2

•$ svcs -o state,ctid,fmri apache2
STATE             CTID   FMRI
online           91064   svc:/network/http:apache2

$ ls
ls: not found

•$ echo *
local.cshrc local.login local.profile

•
```

Process Privileges

- Solaris kernel checks for privileges and not just `UID == 0`!
  - Division of `root` authority into over 60 discrete privileges.
  - Privileges can be granted to processes based on need.
  - Privileges can be disabled or dropped when not needed.
  - Child processes can have different (fewer) privileges than the parent.
- Completely backward compatible and extensible.
  - No changes required to use existing code.
- Privilege bracketing helps to mitigate effects of future flaws.
  - e.g., `proc_fork` and `proc_exec`
  - e.g., `proc_info`

Process Privilege Sets

- E - Effective
  - > Privileges in effect
- P - Permitted set
  - > Upper bound of E
- I - Inheritable set
  - > Privileges of executed programs
- L - Limit set
  - > Upper bound for the process and all its descendants



Process Privilege Inheritance

- Limit (L) is unchanged
- L is used to bound privileges in Inheritable (I)
  - $I' = I \cap L$
- Child's Permitted (P') & Effective (E') are:
  - $P' = E' = I'$
- Typical process
  - P = E = I = {basic}
  - L = {all privileges}
  - Since P = E = I, children run with same privileges

Process Privileges

- “basic” privileges
  - `file_link_any, proc_exec, proc_fork, proc_info, proc_session`
- “all” privileges
  - includes “basic” + over 60 administrative privileges
  - `dtrace_kernel, file_dac_write, net_privaddr, proc_priocntl, sys_net_config, etc.`
- “zone” privileges
  - the set of privileges available to a Solaris zone.
- Trusted Extensions privileges
  - privileges specific for use when TX is enabled.

Root Account Still Special

- *root* owns all configuration/system files
  - `UID 0` is therefore still very powerful
- Privilege escalation prevention
  - Require ALL privileges to modify objects owned by *root* when euid ≠ 0
  - Fine tuning in certain policy routines
    - Not all privileges, only *nosuid* mounts
- Prefer services be non-`UID 0` + privileges
  - Additive approach is safer than `UID 0` – privileges

Using Process Privileges

- ppriv(1)

```
# ppriv -e -D -s -proc_fork,-proc_exec /bin/sh -c finger
sh[387]: missing privilege "proc_fork" (euid = 0, syscall = 143)
needed at cfork+0x18
/bin/sh: permission denied
```

- User Rights Management (RBAC)

```
# grep “Network Management” /etc/security/exec_attr
Network
Management:solaris:cmd:::/sbin/ifconfig:privs=sys_net_config
Network Management:solaris:cmd:::/sbin/route:privs=sys_net_config
```

- Service Management Framework (SMF)

```
# svcprop -p start rpc/bind | grep privileges
start/privileges astring
basic,file_chown,file_chown_self,file_owner,net_privaddr,
proc_setid,sys_nfs,net_bindmlp
stop/limit_privileges astring :default
```

- Privilege Aware Commands / Services
  - e.g., *ping*, *rmformat*, *quota*, *rpcbind*, *nfsd*, *mountd*

Process Privileges Example #1

-
```
$ ppriv $$
28983:  bash
flags = <none>
        E: basic
        I: basic
        P: basic
        L: all
```

```
$ ppriv -l basic
file_link_any
proc_exec
proc_fork
proc_info
proc_session
```

```
$ ppriv -De cat /etc/shadow
cat[3988]: missing privilege "file_dac_read" (euid =
101, syscall = 225) needed at ufs_iaccess+0xc9
cat: cannot open /etc/shadow

$ ppriv -s -proc_fork,-proc_exec -De /bin/vi
[attempt to run a command/escape to a shell]
vi[4180]: missing privilege "proc_fork" (euid = 101,
syscall = 143) needed at cfork+0x3b
```

Process Privileges Example #2

```
# ppriv -S `pgrep rpcbind`
933:    /usr/sbin/rpcbind
flags = PRIV_AWARE
        E: net_bindmlp,net_privaddr,proc_fork,sys_nfs
        I: none
        P: net_bindmlp,net_privaddr,proc_fork,sys_nfs
        L: none
```

```
# ppriv -S `pgrep statd`
5139:   /usr/lib/nfs/statd
flags = PRIV_AWARE
        E: net_bindmlp,proc_fork
        I: none
        P: net_bindmlp,proc_fork
        L: none
```

Process Privileges Example #3

usr/src/lib/print/libpapi-lpd/common/lpd-port.c

```
•#ifdef PRIV_ALLSETS
    if ((priv_set(PRIV_ON, PRIV_EFFECTIVE,
            PRIV_FILE_DAC_READ, PRIV_FILE_DAC_WRITE, NULL)) < 0) {
        syslog(LOG_ERR, "lpd_port:next_job_id:priv_set fails: :
%m");
        return (-1);
    }
#else
    seteuid(0);
#endif

    /* open the sequence file */
    if (((fd = open(JOB_ID_FILE, O_RDWR)) < 0) && (errno ==
ENOENT))
        fd = open(JOB_ID_FILE, O_CREAT|O_EXCL|O_RDWR, 0644);

    syslog(LOG_DEBUG, "sequence file fd: %d", fd);

#ifdef  PRIV_ALLSETS
    /* drop file access privilege */
    priv_set(PRIV_OFF, PRIV_PERMITTED,
            PRIV_FILE_DAC_READ, PRIV_FILE_DAC_WRITE, NULL);
#else
    seteuid(getuid());
#endif
```

Process Privileges Example #3

usr/src/lib/print/libpapi-lpd/common/lpd-port.c

> Turn Required Privileges On

```
#ifdef PRIV_ALLSETS
    if ((priv_set(PRIV_ON, PRIV_EFFECTIVE,
            PRIV_FILE_DAC_READ, PRIV_FILE_DAC_WRITE, NULL)) < 0) {
        syslog(LOG_ERR, "lpd_port:next_job_id:priv_set fails: :
%m");
        return (-1);
    }
#else
    seteuid(0);
#endif

    /* open the sequence file */
    if (((fd = open(JOB_ID_FILE, O_RDWR)) < 0) && (errno ==
ENOENT))
        fd = open(JOB_ID_FILE, O_CREAT|O_EXCL|O_RDWR, 0644);

    syslog(LOG_DEBUG, "sequence file fd: %d", fd);

#ifdef  PRIV_ALLSETS
    /* drop file access privilege */
    priv_set(PRIV_OFF, PRIV_PERMITTED,
            PRIV_FILE_DAC_READ, PRIV_FILE_DAC_WRITE, NULL);
#else
    seteuid(getuid());
#endif
```

Process Privileges Example #3

usr/src/lib/print/libpapi-lpd/common/lpd-port.c

```
•#ifdef PRIV_ALLSETS
    if ((priv_set(PRIV_ON, PRIV_EFFECTIVE,
            PRIV_FILE_DAC_READ, PRIV_FILE_DAC_WRITE, NULL)) < 0) {
        syslog(LOG_ERR, "lpd-port:next_job_id:priv_set fails: :
%m");
        return (-1);
    }
#else
    seteuid(0);
#endif
```

>Turn Required Privileges On

```
    /* open the sequence file */
    if (((fd = open(JOB_ID_FILE, O_RDWR)) < 0) && (errno ==
ENOENT))
        fd = open(JOB_ID_FILE, O_CREAT|O_EXCL|O_RDWR, 0644);

    syslog(LOG_DEBUG, "sequence file fd: %d", fd);

#ifdef PRIV_ALLSETS
    /* drop file access privilege */
    priv_set(PRIV_OFF, PRIV_PERMITTED,
            PRIV_FILE_DAC_READ, PRIV_FILE_DAC_WRITE, NULL);
#else
    seteuid(getuid());
#endif
```

>Perform the Privileged Operation(s)

Process Privileges Example #3

usr/src/lib/print/libpapi-lpd/common/lpd-port.c

```
#ifdef PRIV_ALLSETS
    if ((priv_set(PRIV_ON, PRIV_EFFECTIVE,
            PRIV_FILE_DAC_READ, PRIV_FILE_DAC_WRITE, NULL)) < 0) {
        syslog(LOG_ERR, "lpd_port:next_job_id:priv_set fails: :
%m");
        return (-1);
    }
```

>Turn Required Privileges On

```
#else
    seteuid(0);
#endif

    /* open the sequence file */
    if (((fd = open(JOB_ID_FILE, O_RDWR)) < 0) && (errno ==
ENOENT))
        fd = open(JOB_ID_FILE, O_CREAT|O_EXCL|O_RDWR, 0644);
```

>Perform the Privileged Operation(s)

```
    syslog(LOG_DEBUG, "sequence file fd: %d", fd);

#ifdef  PRIV_ALLSETS
    /* drop file access privileges */
    priv_set(PRIV_OFF, PRIV_PERMITTED,
            PRIV_FILE_DAC_READ, PRIV_FILE_DAC_WRITE, NULL);
```

>Turn Required Privileges Off

```
#else
    seteuid(getuid());
#endif
```

Process Privilege Debugging

- web_svc zone: # **svcadm disable apache2**
- global zone: # **privdebug -v -f -n httpd**
- web_svc zone: # **svcadm enable apache2**
- global zone: [output of privdebug command]

```
•STAT   TIMESTAMP             PPID   PID     PRIV           CMD
USED    273414882013890   4642    4647   net_privaddr   httpd
USED    273415726182812   4642    4647   proc_fork      httpd
USED    273416683669622   1       4648   proc_fork      httpd
USED    273416689205882   1       4648   proc_fork      httpd
USED    273416694002223   1       4648   proc_fork      httpd
USED    273416698814788   1       4648   proc_fork      httpd
USED    273416703377226   1       4648   proc_fork      httpd
```

> privdebug is available from the OpenSolaris Security Community, http://www.opensolaris.org/os/community/security/projects/privdebug/

Zones

- Zones are virtualized application environments.
  - No direct access to hardware.
- Zones have security boundaries around them.
- Zones have their own:
  - root directory, naming service configuration, process containment, resource controls, devices, etc.
- Zones communicate via network only (default).
  - shared vs. exclusive IP
- Zones operate with fewer privileges (default).
  - some privileges can be added or removed

Why run services in Zones?

- Restricted Operations for Enhanced Security
  - > Individual Solaris OS hardening and RBAC configurations.
  - > Prohibited from directly accessing the kernel or raw memory.
  - > Prohibited from manipulating network interfaces and kernel modules.
- Enforcement with Integrity
  - > Configurable privileges, sparse root zones, IP Instances, IP Filter, etc.
- Resource Control and Management
  - > CPU, Memory, Disk, Networking, Devices, etc.
- Observability with Integrity
  - > BART, Solaris Auditing, etc.

Zones Security – System Calls

- Permitted System Calls:
  - *chmod(2)*, *chroot(2)*, *chown(2)*, and *setuid(2)*
- Prohibited System Calls:
  - *memcntl(2)*, *mknod(2)*, *stime(2)*, and *pset_create(2)*
- Limited System Calls:
  - *kill(2)*

Zones Security – Devices

- */dev* Permitted System Calls:
  - *chmod(2)*, *chown(2)*, and *chgrp(1)*
- */dev* Prohibited System Calls:
  - *rename(2)*, *unlink(2)*, *symlink(2)*, *link(2)*, *creat(2)*, and *mknod(2)*
- Forced *nodevices* mount option
  - Prevents import of malicious device files from NFS and other foreign sources.
- Security audit performed on all drivers included in default zone configuration.

Zones Security – Privileges

- Mandatory privileges
  - > Privileges required by a non-global zone.
  - > `proc_fork, proc_exec, proc_mount, ...`
- Restricted privileges
  - > Privileges prohibited from use in a non-global zone.
  - > `dtrace_kernel, sys_config, sys_net_config, ...`
- Optional privileges
  - > Privileges that can be added to a non-global zone.
  - > `dtrace_user, proc_lock_memory, sys_time, ...`
- Other default privileges can be taken away!

Zones Example #1

```
# modload autofs
Insufficient privileges to load a module
```

```
# modunload -i 101
Insufficient privileges to unload a module
```

```
# snoop
snoop: No network interface devices found
```

```
# mdb -k
mdb: failed to open /dev/ksyms: No such file or
directory
```

```
# dtrace -l
   ID   PROVIDER            MODULE                          FUNCTION
NAME
```

```
# ppriv -D -e route add net default 10.1.2.3
route[4676]: missing privilege "sys_net_config"
 (euid = 0, syscall = 4) needed at ip_rts_request+0x138
add net default: gateway 10.1.2.3: insufficient
privileges
```

Zones Example #2

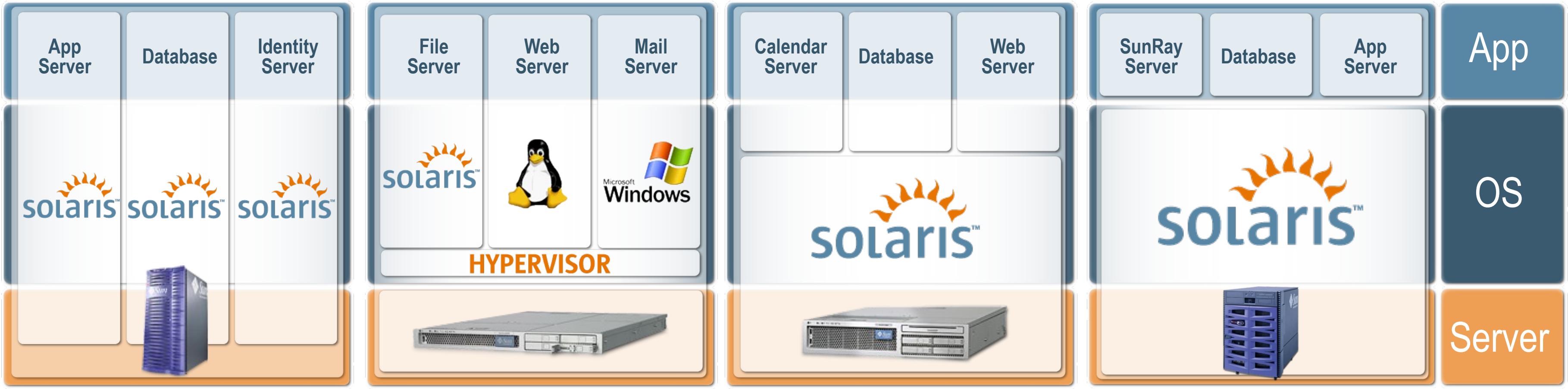
```
•# mount -p
/           -   /           zfs         - no

rw,devices,setuid,exec,atime
/dev        -   /dev        lofs    - no zonedevfs
/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
[...]

•# mv /usr/bin/login /usr/bin/login.foo
mv: cannot rename /usr/bin/login to /usr/bin/login.foo:

Read-only file system
```

Zones Example #3

```
•# zonecfg -z myzone info limitpriv
limitpriv: default,sys_time

•# zlogin myzone ppriv -l zone | grep sys_time
sys_time

•# zlogin myzone svcs -v ntp
STATE          NSTATE        STIME    CTID   FMRI
online         -             10:17:58    214
svc:/network/ntp:default

•# zlogin myzone ntpq -c peers
  remote      refid         st t when poll reach   [...]
================================================[...]
*blackhole 129.146.228.54  3 u   48   64   77    [...]

•# ssh blackhole date ; date ; zlogin myzone date
Thu Jun 15 10:25:25 EDT 2006
Thu Jun 15 10:25:25 EDT 2006
Thu Jun 15 10:25:25 EDT 2006
```

Virtualization / Compartmentalization

Hard Partitions | Virtual Machines | OS Virtualization | Resource Management



Multiple OSs ← | → **Single OS**

Trend to flexibility → | ← **Trend to isolation**

Dynamic System Domains

Logical Domains

Xen

VMware

Microsoft Virtual Server

Solaris Containers (Zones + SRM)

Solaris Trusted Extensions

Solaris Containers for Linux Applications

Solaris Resource Manager (SRM)

ZFS Data Integrity

- Everything is “copy on write”
  - > Never overwrites live data
  - > On disk state is always valid
  - > No need to fsck(1M)
- Everything is transactional
  - > Related changes succeed or fail as a whole
  - > No need for journaling
- Everything is validated with a 256-bit checksum
  - > No silent data corruption
  - > No panics due to corrupted meta-data
  - > “Bad data” can be healed using mirrored copies

ZFS Data Security

- NFSv4 / NTFS-style Access Control Lists
  - Granular access can be allowed/denied (w/inheritance)
- Authentication with Cryptographic Checksums
  - Selectable 256-bit checksum algorithms, including SHA-256
  - Uber-checksum provides check for the entire ZFS pool
- File system Snapshots
  - Read-only version of a file system at a specific point in time.
- File system Quotas and Reservations
  - Set maximum (quota) or minimum (reservation) usage limits.

ZFS Example #1

-
  ```
  $ touch testfile

  $ chmod 600 testfile
  $ chmod A+user:gmb:read_data:allow testfile

  $ ls -l testfile
  -rw-------+  1 gbrunett gbrunett       0 Nov  7 14:22 testfile

  $ ls -v testfile
  -rw-------+  1 gbrunett gbrunett       0 Nov  7 14:22 testfile
   0:user:gmb:read_data:allow
   1:owner@:execute:deny
   2:owner@:read_data/write_data/append_data/write_xattr/
     write_attributes/write_acl/write_owner:allow
   3:group@:read_data/write_data/append_data/execute:deny
   4:group@::allow
   5:everyone@:read_data/write_data/append_data/write_xattr/
     execute/write_attributes/write_acl/write_owner:deny
   6:everyone@:read_xattr/read_attributes/read_acl/
     synchronize:allow
  ```

ZFS Example #2

```
•$ touch test-xattr

•$ runat test-xattr cp /etc/motd .

•$ runat test-xattr ls
motd

•$ touch test-no-xattr

•$ chmod A+user:gbrunett:write_xattr:deny test-no-xattr

•$ runat test-no-xattr cp /etc/motd .
runat: cannot open attribute directory for test-no-xattr:
Permission denied
```

ZFS Example #3

- `$ profiles`

```
[...]
ZFS File System Management
[...]
Basic Solaris User
All
```

- `$ pfexec zfs set quota=4g laptop/ws`

- `$ pfexec zfs list -o name,mountpoint,quota`

```
NAME                  MOUNTPOINT            QUOTA
laptop                /laptop                none
laptop/briefcase      /laptop/briefcase      none
laptop/ws             /laptop/ws               4G
```

ZFS Delegation

- Grant or revoke specific rights to ZFS pools and volumes.
  - create, destroy, clone, snapshot, mount, etc.
- Set specific properties on ZFS pools and volumes.
  - mountpoint, sharenfs, compression, setuid, etc.
- Assignments can be made to both users and groups.
  - assigned rights can optionally be granted to other users and groups.

ZFS Example #4

```
•$ id
uid=102(gmb) gid=102(gmb)

•$ zfs list -r pool/home/gmb
NAME             USED  AVAIL  REFER  MOUNTPOINT
pool/home/gmb  19.5K  25.9G  19.5K  /pool/home/gmb

•$ zfs allow pool/home/gmb
$ zfs snapshot pool/home/gmb@backup
cannot create snapshot 'pool/home/gmb@backup': permission
denied

•$ pfexec zfs allow gmb snapshot,mount pool/home/gmb
$ zfs allow pool/home/gmb
-------------------------------------------------------------
Local+Descendent permissions on (pool/home/gmb)
        user gmb mount,snapshot
-------------------------------------------------------------
$ zfs snapshot pool/home/gmb@backup
$ zfs list -r pool/home/gmb
NAME                    USED  AVAIL  REFER  MOUNTPOINT
pool/home/gmb         19.5K  25.9G  19.5K  /pool/home/gmb
pool/home/gmb@backup      0      -  19.5K  -
```

Kerberos

- MIT Kerberos Code-base Refresh
- Kerberos Ticket / Credentials Auto-Renewal
- Kebreros LDAP Backend
- KDC Incremental Propagation
- kclient Auto-configuration Tool
- pam_krb5_migrate KDC Auto-population Tool
- TCP and IPv6 Support
- AES-128, AES-256, 3DES, RC4-HMAC Support
- SPNego – GSS-API Dynamic Security Negotiation
- Bundled Remote Applications (Clients & Servers)
  - telnet, ftp, rlogin, rsh, rcp, rdist, Secure Shell, Mozilla and Apache
- Public Kerberos Developer APIs

Secure Shell

- OpenSSH 3.6p2++ Refresh
- GSS-API Support
- Enhanced Password Aging Support
- Keyboard “Break” Sequence Support
- X11 Forwarding “on” by default
- RC4, AES CTR mode Encryption Support
- /etc/default/login Synchronization
- SSH2 Rekeying
- Server Side Keepalives

TCP Wrappers

- Supports both tcpd and libwrap and integrated with:
  - ssh and sendmail (automatically)
  - rpcbind (optionally)

```
$ svcprop -p config rpc/bind | grep wrappers
config/enable_tcpwrappers boolean false
```

  - inetd-services (optionally, globally or per-service)

```
$ svcprop -p defaults inetd | grep wrappers
defaults/tcp_wrappers boolean false
```

```
$ inetadm -l telnet | grep wrappers
default  tcp_wrappers=FALSE
```

- Configured using /etc/hosts.{allow, deny} and logs to syslog:

```
Nov 10 15:18:03 blackhole sshd[17568]:
[ID 947420 auth.warning] refused connect from
192.168.1.136
```

IP Filter

- Stateful and stateless packet inspection – IPv4, IPv6
- Kernel-based packet filtering
- Protocol proxies (TCP, UDP, FTP, rcmds, etc.)
- Transparent proxy support
- Text-based configuration
- Support for both NAT and PAT
- SYSLOG Logging
- Lightweight, small footprint, high performance

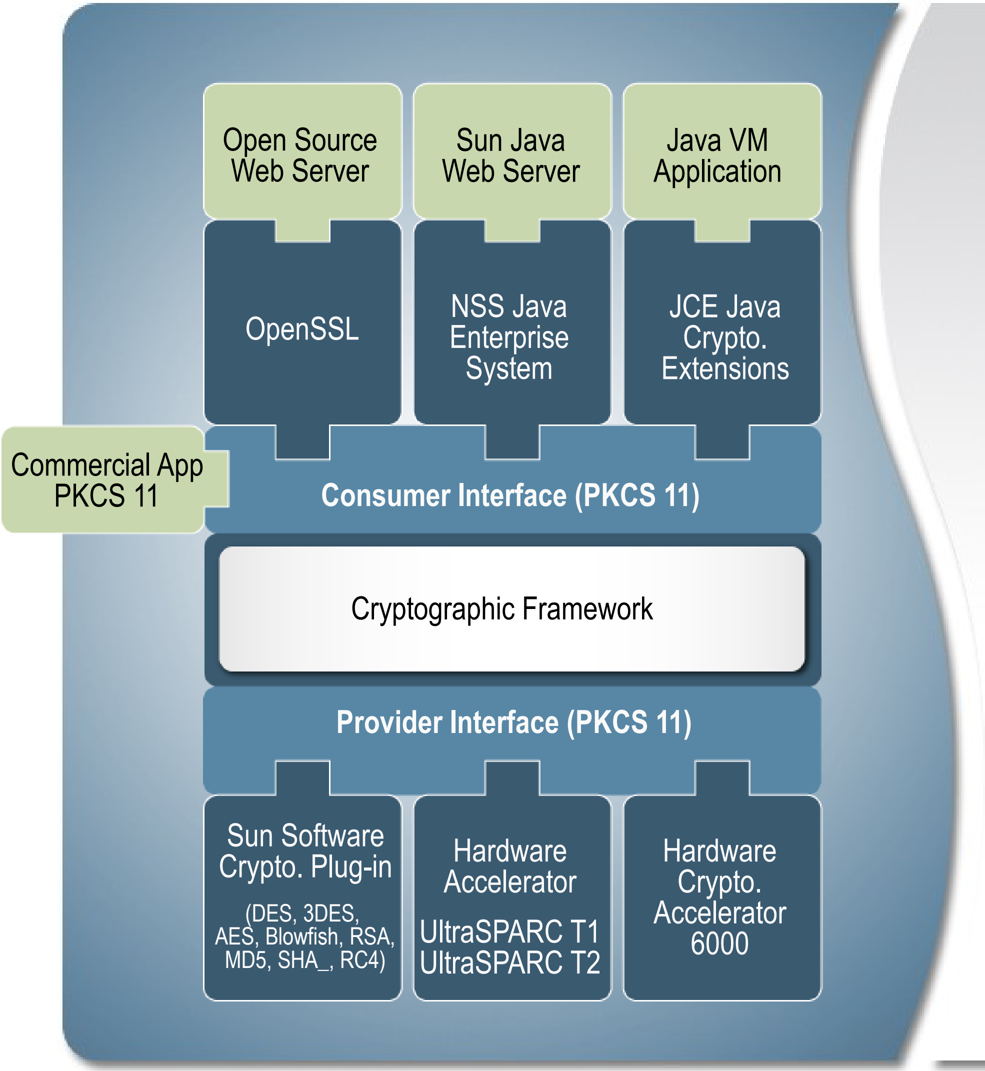
IP Filter Example

```
•pass out quick all keep state keep frags

•# Drop all NETBIOS traffic but don't log it.
block in quick from any to any port = 137 #netbios-ns
block in quick from any to any port = 138 #netbios-dgm
block in quick from any to any port = 139 #netbios-ssn

•# Allow incoming IKE/IPsec
pass in quick proto udp from any to any port = ike
pass in quick proto udp from any to any port = 4500
pass in proto esp from any to any

•# Allow ping
pass in quick proto icmp from any to any icmp-type echo

•# Allow routing info
pass in quick proto udp from any to port = route
pass in quick proto icmp from any to any icmp-type 9 #
routeradvert
pass in quick proto igmp from any to any

•# Block and log everything else that comes in
block in log all
block in from any to 255.255.255.255
block in from any to 127.0.0.1/32
```

Cryptographic Framework

- Standards-based, pluggable framework
  - > Kernel support as well as user-land (PKCS#11)
  - > Supports administrative policies (e.g., FIPS 140 algorithms only)
- By default, supports major algorithms.
  - > Encryption : AES, ECC, Blowfish, RC4, DES, 3DES, RSA
  - > Digest : MD5, SHA-1, SHA-256, SHA-384, SHA-512
  - > MAC : DES MAC, MD5 HMAC, SHA-1 HMAC, SHA-256 HMAC, SHA-384 HMAC, SHA-512 HMAC
  - > Optimized for both SPARC, Intel and AMD
- Framework supports pluggable hardware/software providers:
  - > e.g., UltraSPARC T1/T2 and the Sun CryptoAccelerator 6000

Cryptographic Framework



Open Source Web Server

Sun Java Web Server

Java VM Application

OpenSSL

NSS Java Enterprise System

JCE Java Crypto. Extensions

Commercial App PKCS 11

Consumer Interface (PKCS 11)

Cryptographic Framework

Provider Interface (PKCS 11)

Sun Software Crypto. Plug-in (DES, 3DES, AES, Blowfish, RSA, MD5, SHA_, RC4)

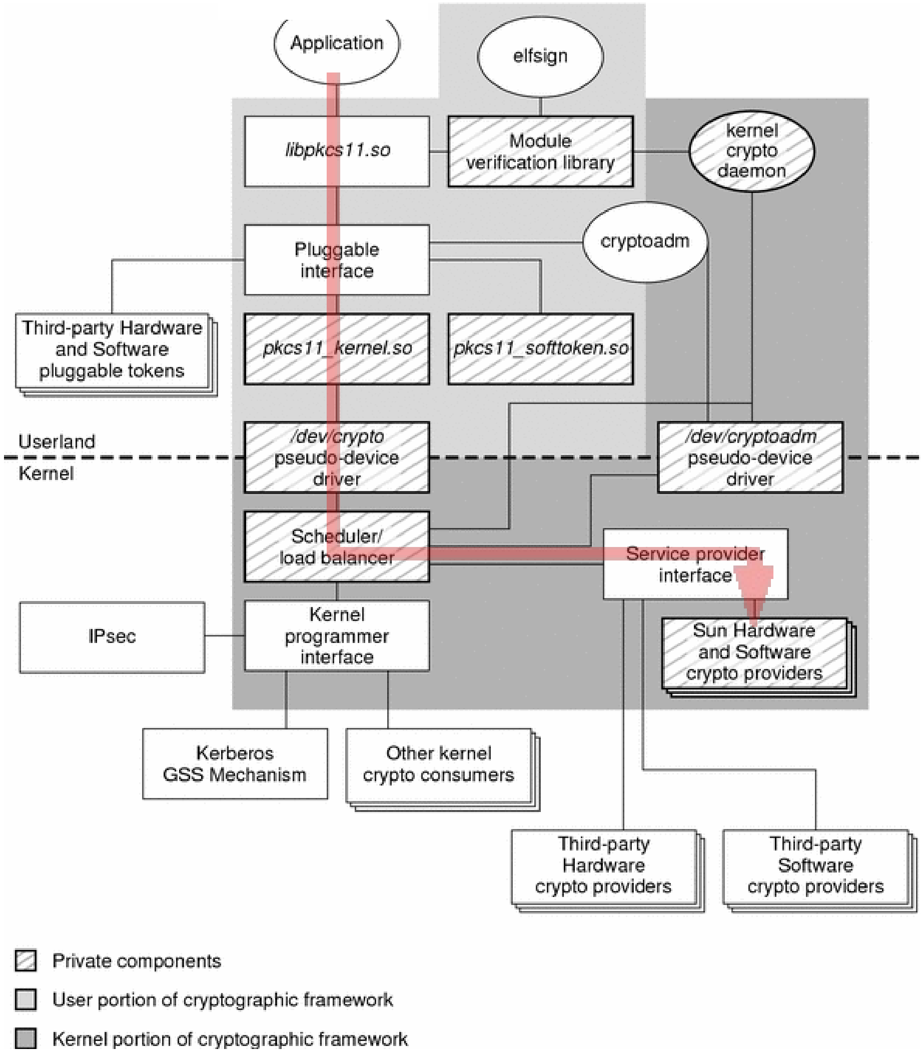
Hardware Accelerator UltraSPARC T1 UltraSPARC T2

Hardware Crypto. Accelerator 6000

- Now the framework for
- cryptography is standardized
- and extensible.
-
- Your current cryptographic
- choices and any future
- technology can easily plug in
- and just work.
-
- Standards-based framework
- Same API, software or hardware
- Extensible for future technologies

T2/Solaris Cryptographic Architecture

- Access to T2 accelerators is controlled by Solaris CF
- Userland access is via PKCS#11
  - Simple to modify applications to use PKCS#11 (if not used already)
  - Can interface via OpenSSL
  - Offload from Java (JCE)
- Kernel modules communicate directly with the kernel crypto



Application
elfsign
libpkcs11.so
Module verification library
kernel crypto daemon
Pluggable interface
cryptoadm
Third-party Hardware and Software pluggable tokens
pkcs11_kernel.so
pkcs11_softtoken.so
Userland
Kernel
/dev/crypto pseudo-device driver
/dev/cryptoadm pseudo-device driver
Scheduler/ load balancer
Service provider interface
IPsec
Kernel programmer interface
Sun Hardware and Software crypto providers
Kerberos GSS Mechanism
Other kernel crypto consumers
Third-party Hardware crypto providers
Third-party Software crypto providers
▨ Private components
□ User portion of cryptographic framework
■ Kernel portion of cryptographic framework

Copyright © 2008 Sun Microsystems, Inc.

UltraSPARC T2 Processor Performance

Competitive Cryptographic Performance

- Outperforms competing processors by up to 10X
  - With significant core idle time that can be used for other processing

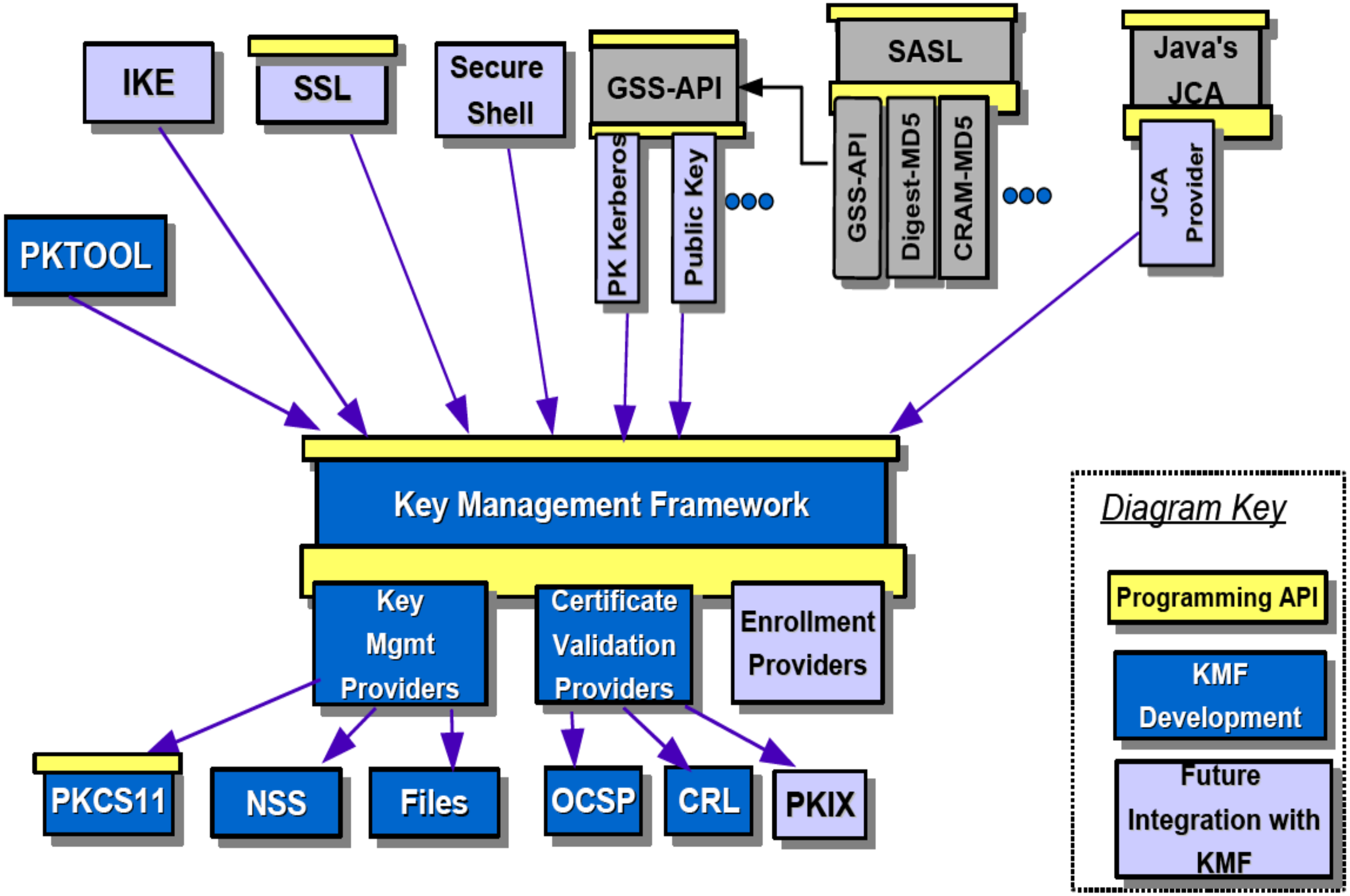
Cipher	2.2GHz dual-core Opteron	2.7GHz quad-core Clovertown	1.4GHz UltraSPARC T2
RSA1024	2.3K Ops/sec	4.8 K Ops/sec	37.0K Ops/sec
AES-128	1.6 Gb/sec	4.2 Gb/s	44.0 Gb/sec

- Outperforms accelerator cards by a wide margin

Cipher	Sun SCA6000	Cavium Nitrox PX	1.4GHz UltraSPARC T2
RSA1024	13K Ops/sec	12K Ops/sec	37K Ops/sec
AES-128	1.0Gb/sec	2.5Gb/sec	44Gb/sec

- On-chip accelerators are more versatile than off-chip solutions
  - Cost effective to off-load even small packets with UltraSPARC T2 processor

Key Management Framework



Basic Audit and Reporting Tool

- File-level integrity validation tool:
  - Evalutes: uid, gid, permissions/acls, contents, mtime, size, type, etc.
  - Enables point-in-time comparison against a previous snapshot.

```
# cat ./rules
/etc
CHECK all

# find /etc | bart create -I > newManifest

# bart compare -r ./rules ./oldManifest
./newManifest
/etc/user_attr:
size control:28268  test:23520
acl  control:user::rw-,group::rw-,mask:r-
x,other:r--
        test:user::rw-,group::rw-,mask:r-
x,other:rw-
  contents
```

Solaris Fingerprint Database

- Searchable database of MD5 fingerprints for files included in Solaris, Trusted Solaris, and bundled software.

-

```
# digest -v -a md5 /usr/lib/ssh/sshd
md5 (/usr/lib/ssh/sshd) =
b94b091a2d33dd4d6481dffa784ba632
```

- [Process fingerprint using the Solaris Fingerprint DB]

-

```
b94b091a2d33dd4d6481dffa784ba632 -
(/usr/lib/ssh/sshd)
 - 1 match(es)
      * canonical-path: /usr/lib/ssh/sshd
      * package: SUNWsshdu
      * version: 11.10.0,REV=2005.01.21.15.53
      * architecture: sparc
      * source: Solaris 10/SPARC
```

Solaris Audit

- Kernel auditing of system calls and administrative actions.
  - Can record events happening in any zone (from the global zone).
    - Can also delegate audit configuration to local zone administrators.
  - Can capture complete command line and environment.
  - Records original (audit) ID as well as current credentials.
  - Audit trail can be formatted as text, XML, and/or delivered via syslog.
- Example:

```
>header,77,2,su,,tundra,2006-11-06 21:55:31.386
-08:00
subject,joe,joe,other,joe,other,2444,1898931306,
12114 22 marduk
text,root
return,failure,Authentication failed
```

- Example adapted from the Sun BluePrint: Enforcing the Two-Person Rule Via Role-based Access Control in the Solaris 10 OS, http://www.sun.com/blueprints/0805/819-3164.pdf

Trusted Solaris History

Product	Year	Evaluation
SunOS MLS 1.0	1990	TCSEC Conformance (1985 Orange Book)
SunOS CMW 1.0	1992	ITSEC Certified for E3 / F-B1
Trusted Solaris 1.2	1995	ITSEC Certified for E3 / F-B1
Trusted Solaris 2.5.1	1996	ITSEC Certified for E3 / F-B1
Trusted Solaris 8	2000	Common Criteria Evaluated: CAPP, RBACPP, LSPP at EAL4+

- *Mandatory Access Control, Labeled Desktop, Labeled Printing, Labeled Networking, Labeled Filesystems, Device Allocation, etc.*

Solaris Trusted Extensions

- A redesign of the Trusted Solaris product using a layered architecture.
- An extension of the Solaris 10 security foundation providing access control policies based on the sensitivity/label of objects.
- A set of label-aware services which implement multilevel security.

Extending Solaris 10 Security Features

- Process Rights Management (Privileges)
  - > Fine-grained privileges for X windows
  - > Rights management applied to desktop actions
- User Rights Management (RBAC)
  - > Labels and clearances
  - > Additional desktop policies
- Solaris Containers (Zones)
  - > Unique Sensitivity Labels
  - > Trusted (label-based) Networking

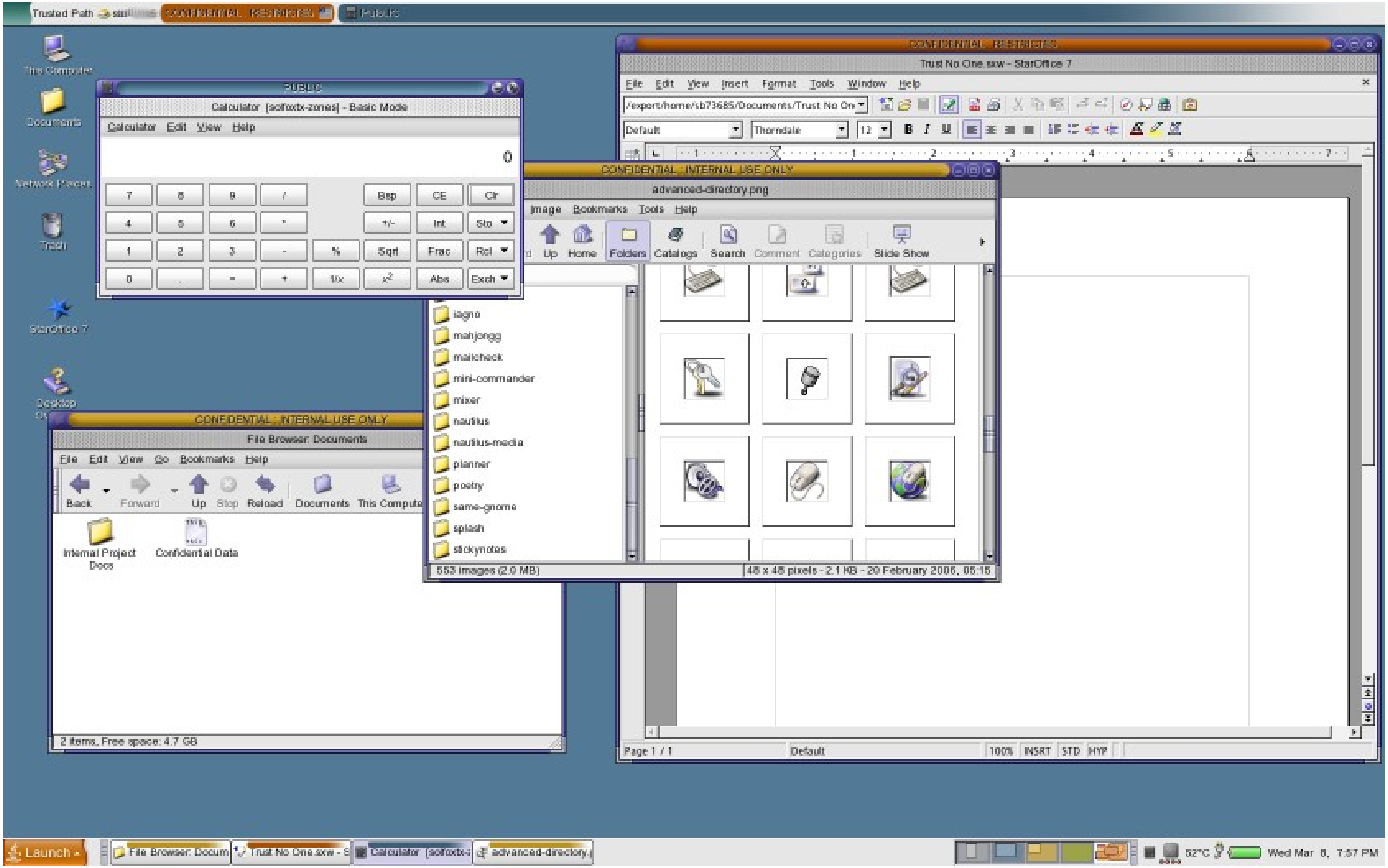
Trusted Extensions in a Nutshell

- Every object has a label associated with it.
  - Files, windows, printers, devices, network packets, network interfaces, processes, etc.
- Accessing or sharing data is controlled by the relationships between the labels of different objects.
  - 'Secret' objects can not see 'Top Secret' objects.
  - 'Company Internal' can not send to 'Partner' networks.
- Administrators utilize Solaris Roles for duty separation.
  - Installation, System Admin., Security Admin., etc.

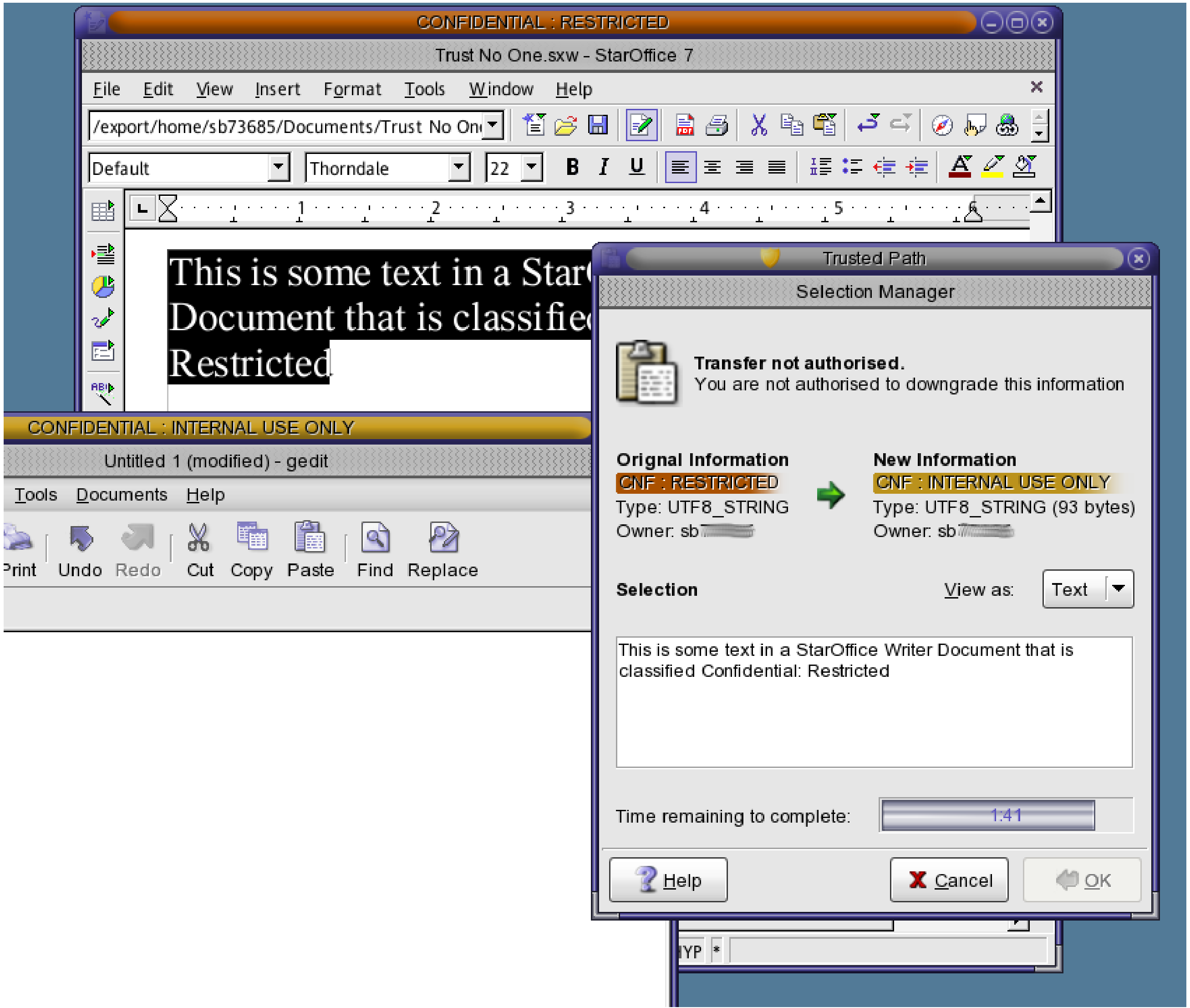
What are Label-Aware Services?

- Services that are trusted to protect multi-level information according to predefined policy.
- Trusted Extensions label-aware service include:
  - Labeled Desktops
  - Labeled Printing
  - Labeled Networking
  - Labeled Filesystems
  - Label Configuration and Translation
  - System Management Tools
  - Device Allocation

Labeled Desktop


Trusted Path
CONFIDENTIAL : RESTRICTED
PUBLIC
This Computer
Documents
Network Places
Trash
StarOffice 7
Desktop
PUBLIC
Calculator [solfoxtx-zones] - Basic Mode
Calculator Edit View Help
0
7 8 9 / Bsp CE Clr
4 5 6 * +/- Int Sto
1 2 3 - % Sqrt Frac Rcl
0 . = + 1/x x2 Abs Exch
CONFIDENTIAL : RESTRICTED
Trust No One.sxw - StarOffice 7
File Edit View Insert Format Tools Window Help
/export/home/sb73685/Documents/Trust No One
Default
Thorndale
12
CONFIDENTIAL : INTERNAL USE ONLY
advanced-directory.png
Image Bookmarks Tools Help
Up Home Folders Catalogs Search Comment Categories Slide Show
iagno
mahjongg
mailcheck
mini-commander
mixer
nautilus
nautilus-media
planner
poetry
same-gnome
splash
stickynotes
553 images (2.0 MB)
48 x 48 pixels - 2.1 KB - 20 February 2006, 05:15
CONFIDENTIAL : INTERNAL USE ONLY
File Browser: Documents
File Edit View Go Bookmarks Help
Back Forward Up Stop Reload Documents This Computer
Internal Project Docs
Confidential Data
2 items, Free space: 4.7 GB
Page 1 / 1
Default
100%
INSRT
STD
HYP
Launch
File Browser: Docum
Trust No One.sxw - S
Calculator [solfoxtx-z
advanced-directory.
52°C
Wed Mar 8, 7:57 PM

Mandatory Access Control



CONFIDENTIAL : RESTRICTED

Trust No One.sxw - StarOffice 7

/export/home/sb73685/Documents/Trust No On

Default

Thorndale

22

This is some text in a Star
Document that is classifie
Restricted

CONFIDENTIAL : INTERNAL USE ONLY

Untitled 1 (modified) - gedit

Trusted Path

Selection Manager

Transfer not authorised.
You are not authorised to downgrade this information

Orignal Information
CNF : RESTRICTED
Type: UTF8_STRING
Owner: sb[redacted]

New Information
CNF : INTERNAL USE ONLY
Type: UTF8_STRING (93 bytes)
Owner: sb[redacted]

Selection

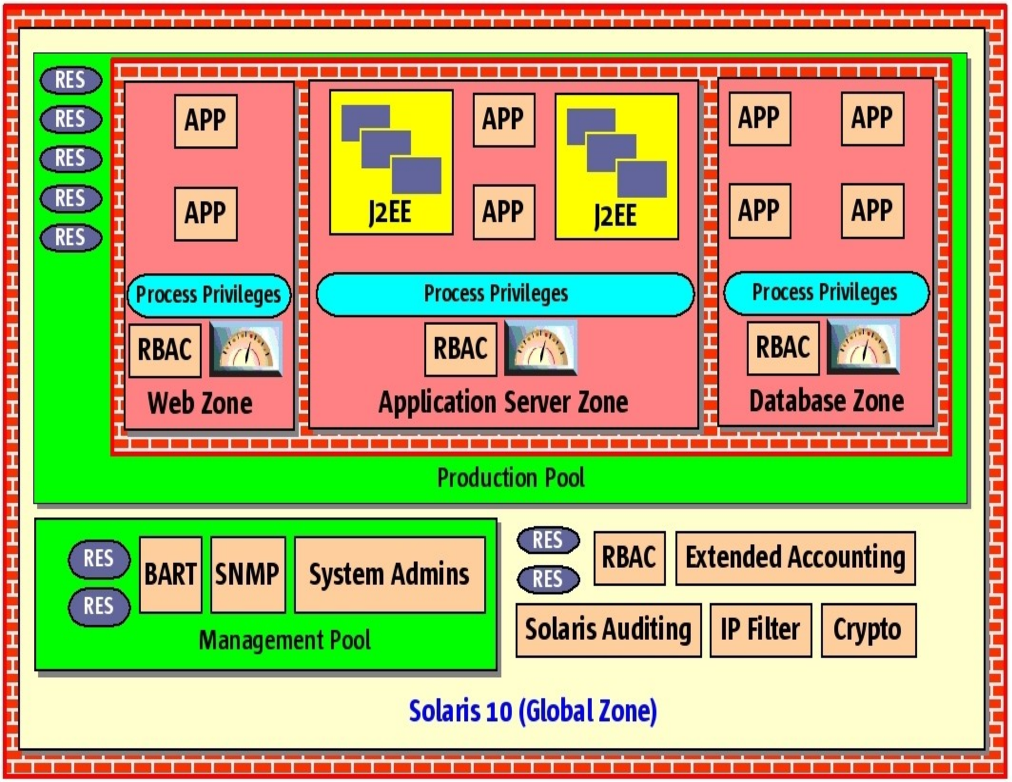
View as: Text

This is some text in a StarOffice Writer Document that is classified Confidential: Restricted

Time remaining to complete: 1:41

Putting It All Together

- Solaris 10 Security – A Secure Foundation for Success:
  - > Reduced Networking Meta (
  - > Signed Binary Execution
  - > Secure Service Managemer
  - > User Rights Management
  - > Process Rights Managemer
  - > Resource Management
  - > Kerberos, SSH, IPsec
  - > Cryptographic Framework
  - > Containers / Zones
  - > IP Filter, TCP Wrappers
  - > Auditing, BART
  - > Trusted Extensions



But wait! There's more!

- Network Security Improvements
  - Kernel SSL Proxy
  - IPsec/IKE NAT Traversal
  - RIPv2 Protocol Support
  - Packet Filtering Hooks
  - Randomized TCP/UDP Ephemeral Port Selection
- Auditing Improvements
  - Audit Trail Noise Reduction
  - Audit Event Reclassification
- New Mount Options
  - noexec, nodevices

and more...

- vacation(1) Mail Filtering
- “root” GID is now “0” (root) not “1” (other)
- ip_respond_to_timestamp now “0”.
- find(1) Support for ACLs
- “death by rm” safety
- OpenSSL libraries with a PKCS#11 engine
- Hardware RNG using Crypto Framework
- open(2) [O_NOFOLLOW], getpeerucred(3c), and many other developer enhancements...
- “Off the Record” plugin for pidgin (nee gaim)
- Sendmail support for TLS

and more...

- NFSv4
  - Support for GSS_API, ACLs, etc.
- Sendmail 8.13.8
  - Support for rate limiting and milters, TLS, etc.
- BIND 9.3.4
  - DNSSEC, Views, IPv6 Support
- Java 5 Security (1.5.0_14)
  - Security tokens, better support for more security standards (SASL, OCSP, TSP), various crypto and GSS security enhancements, etc.
- ... and the list keep right on going...

Actions...

1

1)Enjoy the benefits of Solaris 10 Security today!

2

1)Join the OpenSolaris Security Community!

3

1)Share your requirements, experiences, etc!

For More Information

- Sun Security Home
  - > http://www.sun.com/security
- OpenSolaris Security Community
  - > http://www.opensolaris.org/os/community/security
- Sun Security Coordination Center
  - > http://blogs.sun.com/security & security-alert@sun.com
- Sun Security BluePrints
  - > http://www.sun.com/blueprints
- Sun Security Bloggers
  - > http://blogs.sun.com

Acknowledgements

- Special thanks to the following people who contributed to this presentation:
  - Stephen Browne
  - Casper Dik
  - Shawn Emery
  - Glenn Faden
  - Darren Moffat
  - Scott Rotondo
  - Christoph Schuba
  - Mark Thacker
  - Gary Winiger

Solaris 10 Security
Technical Deep Dive

- **Glenn Brunette**
- Sun Microsystems, Inc.
  - glenn.brunette@sun.com
  - http://blogs.sun.com/gbrunett

OpenSolaris: ZFS Crypto

- Operational goals are to support:
  - software-only and hardware-accelerated environments as well as those requiring hardware key storage.
  - “secure delete” via “key destruction”
  - delegation of key management to individual zones
  - restrict data sets to/from specific zones
  - keep native ZFS copy on write semantics
  - local hardware security module (HSM), trusted platform module (TPM), smart card or password or remote key manager

OpenSolaris: ZFS Crypto

- Current design decisions:
  - Encryption policy will be set at the ZFS data set level.
    - Allows zones to have different keys/algorithms
    - Defined/set at data set creation time
  - Support for encrypted zvols
    - Encrypted raw storage (for databases, etc.)
- Other design considerations:
  - Support for encrypted root filesystems
  - Support for encrypted ZFS send/receive

OpenSolaris: ZFS Crypto

- Integrity protection of both data and meta-data
  - Fletcher and SHA-256
- Confidentiality of both data and filesystem meta-data
  - AES-128, AES-192, AES-256
  - Modes: CBC (Prototype), CCM/GCM (Production)
- No direct use of asymmetric cryptography

OpenSolaris: ZFS Crypto

What is Encrypted?

- YES:
  - All “application” and zvol data
  - POSIX layer data (e.g., permissions, owner, etc.)
  - Directory structure
  - ZFS clones and snapshots
- NO:
  - ZFS pool meta-data (e.g., disks, mount times, raid, etc.)
  - Data set names and properties

OpenSolaris: ZFS Crypto

Current Status

- Phase 1 of 4 in progress
  - Per data set policy for enabling encryption, including algorithm and key length.
  - Per data set keys wrapped by single per pool key
  - Pool key from passphrase using PKCS#5 PBE
  - Pool key stored in PKCS#11 token
- Design review completed
  - Scheduled for 11/2007 code integration into OpenSolaris
- More details at:
  - http://www.opensolaris.org/os/project/zfs-crypto/