Solaris 10 Security
Technical Deep Dive

Glenn Brunette
Distinguished Engineer
Sun Microsystems, Inc.

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 Overview

- Access Control Lists
- Role-based Access Control
- IPsec / IKE
- Solaris Auditing
- TCP Wrappers (inetd)
- 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

Meta Cluster	Size (MB)	# Pkgs	# Set-UID	# Set-GID
Reduced Networking SUNWCrnet	191	92	28	11
Core SUNWCreq	219	139	34	13
End User SUNWCuser	2100	604	57	21
Developer SUNWCprog	2900	844	59	21
Entire SUNWCall	3000	908	72	22
Entire + OEM SUNWCXall	3000	988	80	22

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 signature.

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
```

Non-Executable Stack Example

```
$ cc -o shell-exstk shell.c
$ cc -o shell-noexstk -M /usr/lib/ld/map.noexst shell.c

$ ./shell-exstk
Attempting to start a shell...
$ exit

$ ./shell-noexstk
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
```

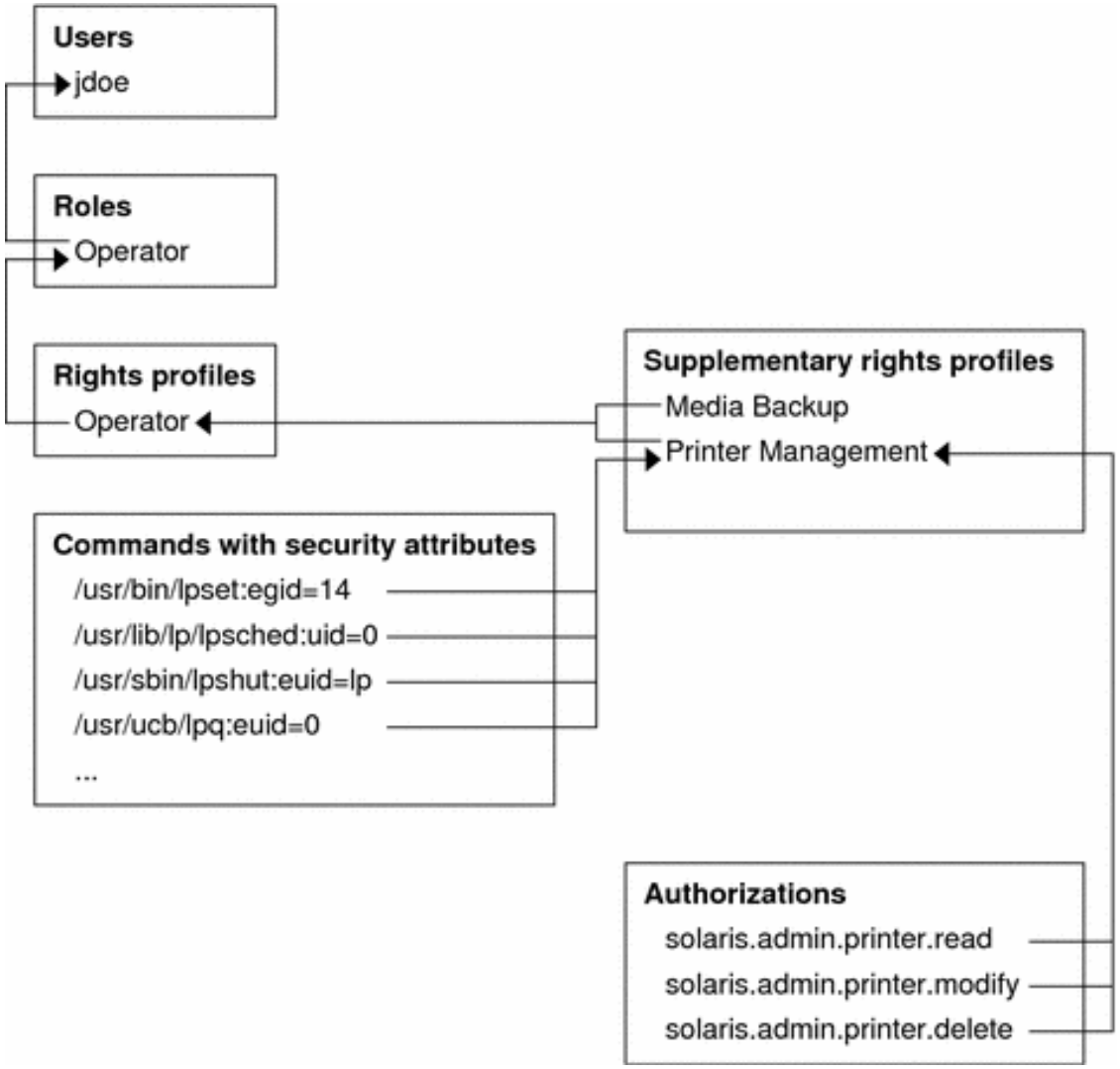
User Rights Management (Roles)

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)



User Rights Management Example

```
# 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

```
$ 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
```

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 (in 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 2006
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 2004
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* – only 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

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

- Local 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 Password History
- Local Banned Password List (Dictionary)
- Local Account Lockout (3 Strikes)
- New “Account Locked” Semantics

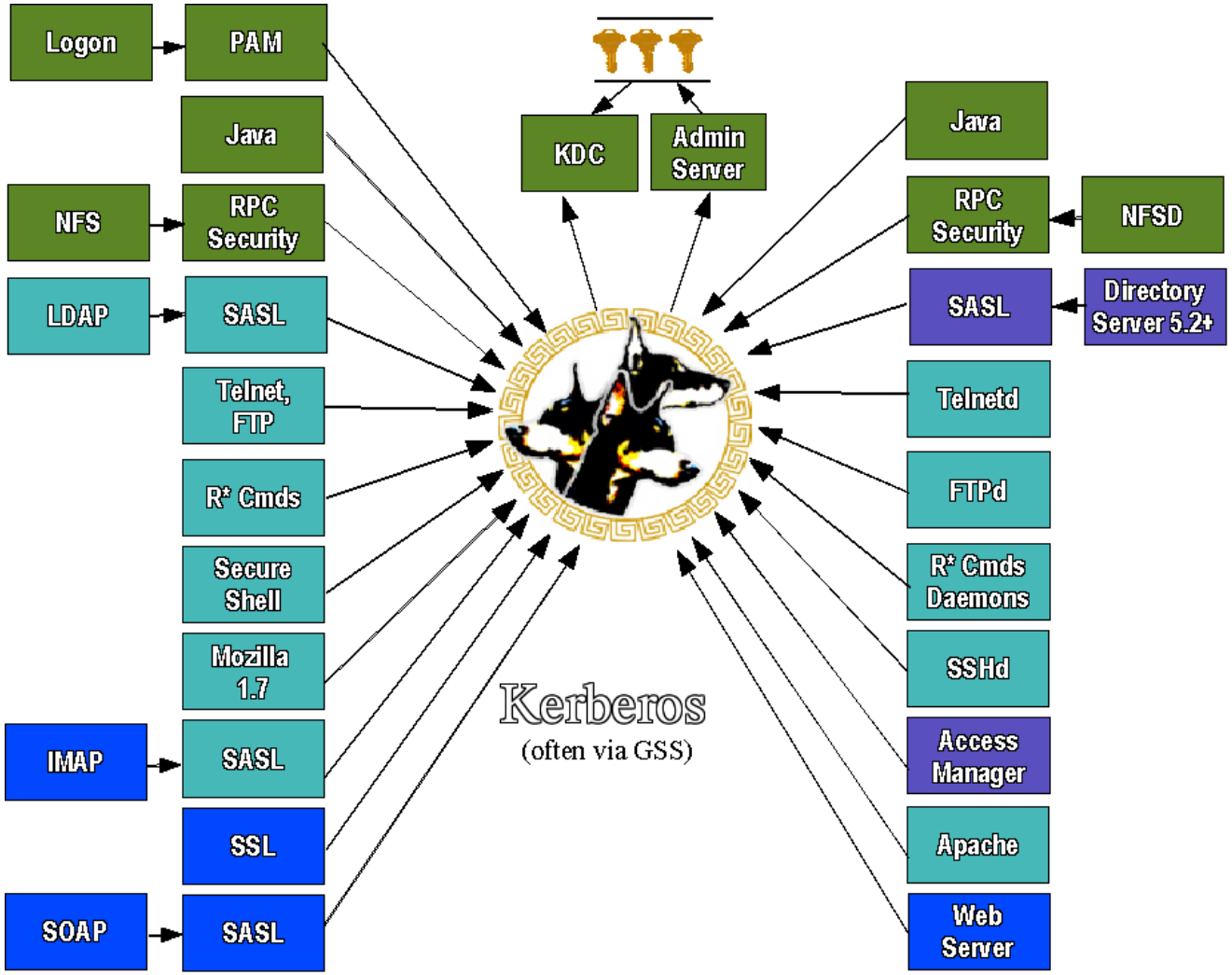
Password Management Example

```
$ 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.
```

Kerberos

- MIT Kerberos 1.3.2 Refresh
- Kerberos Ticket Auto-Renewal
- 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

Kerberos Ecosystem Progress



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

Process Privileges

- Solaris kernel checks for privileges and not just `UID == 0`!
  - Division of `root` authority into discrete privileges (67 and counting)
  - 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`

Zones Privileges Listing

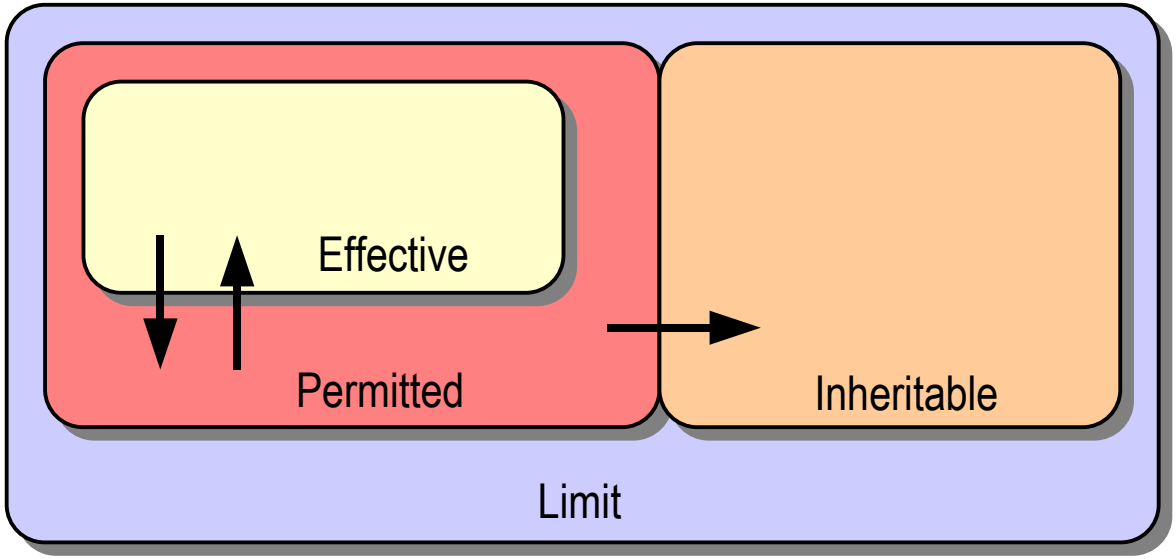
contract_event	contract_observer	cpc_cpu	dtrace_kernel
dtrace_proc	dtrace_user	file_chown	file_chown_self
file_dac_execute	file_dac_read	file_dac_search	file_dac_write
file_downgrade_sl	**file_link_any**	file_owner	file_setid
file_upgrade_sl	graphics_access	graphics_map	ipc_dac_read
ipc_dac_write	ipc_owner	net_bindmlp	net_icmpaccess
net_mac_aware	net_privaddr	net_rawaccess	proc_audit
proc_chroot	proc_clock_highres	**proc_exec**	**proc_fork**
proc_info	proc_lock_memory	proc_owner	proc_priocntl
proc_session	proc_setid	proc_taskid	proc_zone
sys_acct	sys_admin	sys_audit	sys_config
sys_devices	sys_ipc_config	sys_linkdir	sys_mount
sys_net_config	sys_nfs	sys_res_config	sys_resource
sys_suser_compat	sys_time	sys_trans_label	win_colormap
win_config	win_dac_read	win_dac_write	win_devices
win_dga	win_downgrade_sl	win_fontpath	win_mac_read
win_mac_write	win_selection	win_upgrade_sl	

Legend

a = basic

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 ∩ 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

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

Solaris 9 Network Management Rights Profile

```
# grep "Network Management" /etc/security/exec_attr
Network Management:suser:cmd:::/usr/sbin/ifconfig:uid=0
Network Management:suser:cmd:::/usr/sbin/route:uid=0
[...]
```

Solaris 10 Network Management Rights Profile

```
# 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
[...]
```

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/

Zones

- Every system has one “global” zone.
  - `root` in the global zone can see and do anything.
- Every system can have zero or more non-global zones:
  - Non-global zones are virtualized application environments.
    - No direct access to hardware.
  - Non-global zones have security boundaries around them.
    - Restricted access to system calls, device policies, etc.
  - Non-global zones have their own:
    - root directory, naming service configuration, process containment, resource controls, devices, etc.
  - Non-global zones communicate via network only (default).
  - Non-global zones operate with fewer privileges (default).

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 Privileges Listing

contract_event	contract_observer	cpc_cpu	dtrace_kernel
dtrace_proc	dtrace_user	file_chown	file_chown_self
file_dac_execute	file_dac_read	file_dac_search	file_dac_write
file_downgrade_sl	file_link_any	file_owner	file_setid
file_upgrade_sl	graphics_access	graphics_map	ipc_dac_read
ipc_dac_write	ipc_owner	net_bindmlp	net_icmpaccess
net_mac_aware	net_privaddr	net_rawaccess	proc_audit
proc_chroot	proc_clock_highres	proc_exec	proc_fork
proc_info	proc_lock_memory	proc_owner	proc_priocntl
proc_session	proc_setid	proc_taskid	proc_zone
sys_acct	sys_admin	sys_audit	sys_config
sys_devices	sys_ipc_config	sys_linkdir	sys_mount
sys_net_config	sys_nfs	sys_res_config	sys_resource
sys_suser_compat	sys_time	sys_trans_label	win_colormap
win_config	win_dac_read	win_dac_write	win_devices
win_dga	win_downgrade_sl	win_fontpath	win_mac_read
win_mac_write	win_selection	win_upgrade_sl	

Legend

a = mandatory

a = optional

a = prohibited

a = default

a = TX

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
```

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 Filter, etc.
- Resource Control and Management
  - CPU, Memory, Disk, Networking, Devices, etc.
- Observability with Integrity
  - BART, Solaris Auditing, etc.

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 incomming 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
```

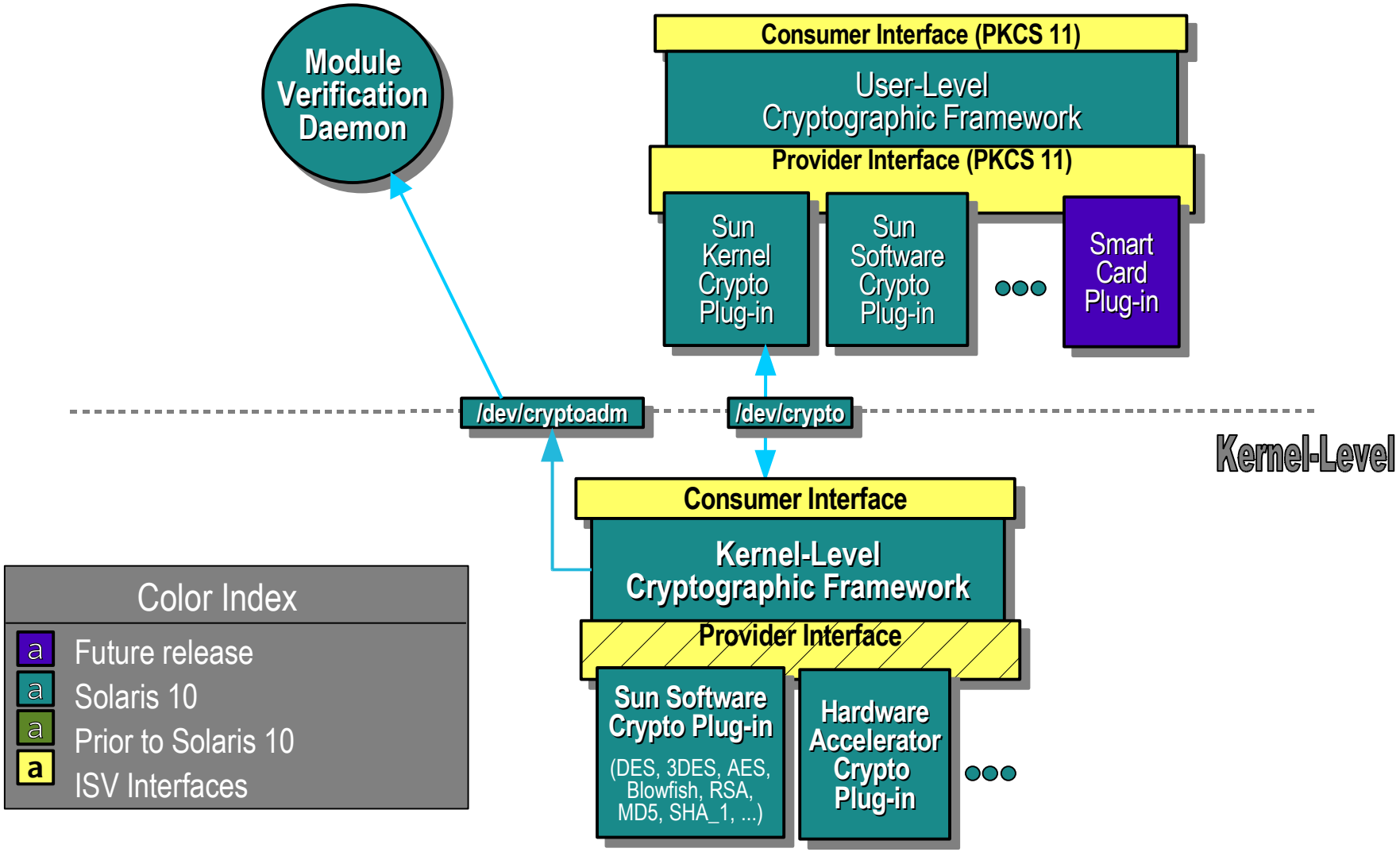
Cryptographic Framework

Extensible cryptographic interfaces.

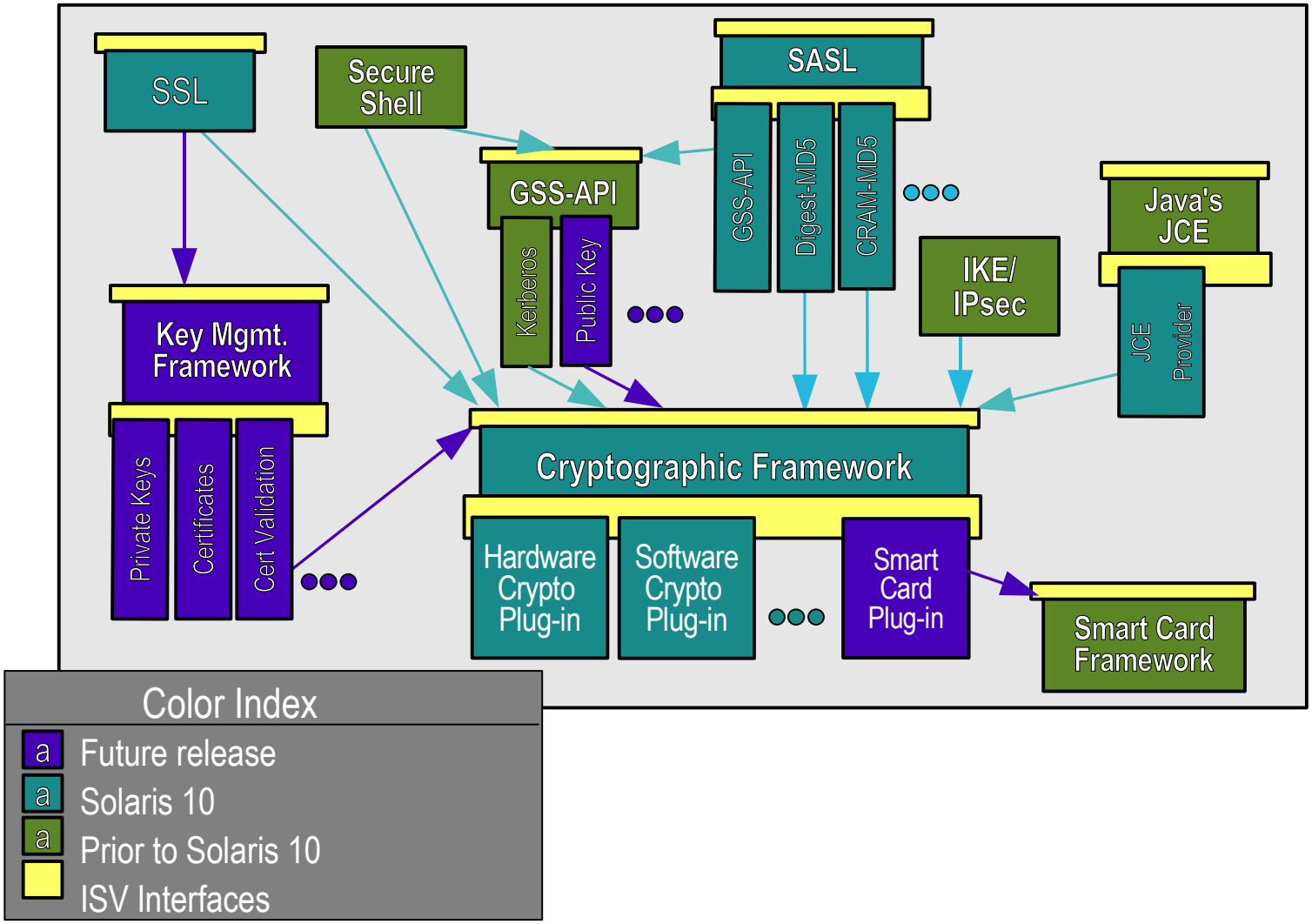
- A common interface for providing/consuming crypto!
  - kernel or user-land
  - hardware and software
- Extensible in order to permit custom functionality.

- By default, supports major algorithms.
  - Encryption : AES, 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

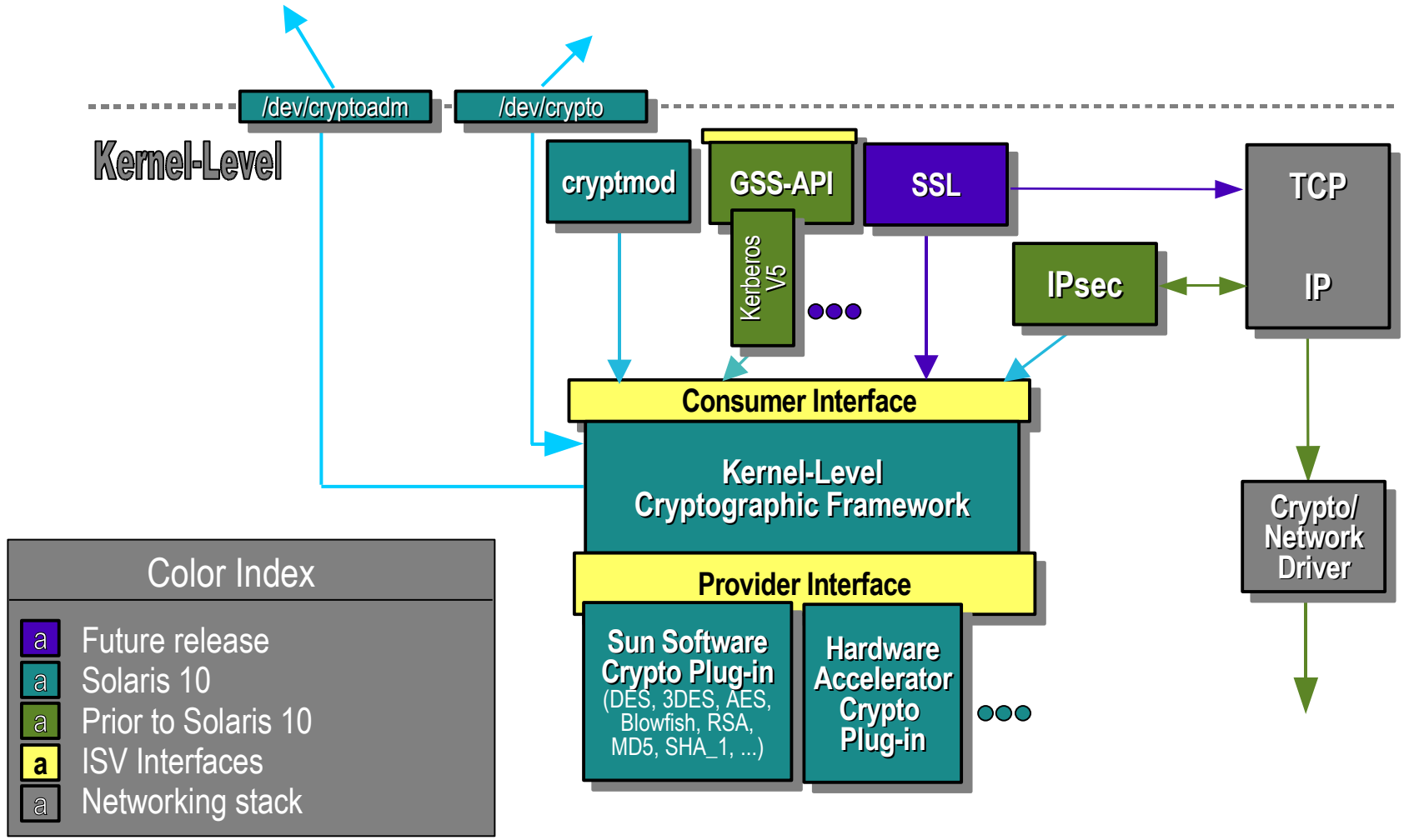
Crypto Framework Architecture



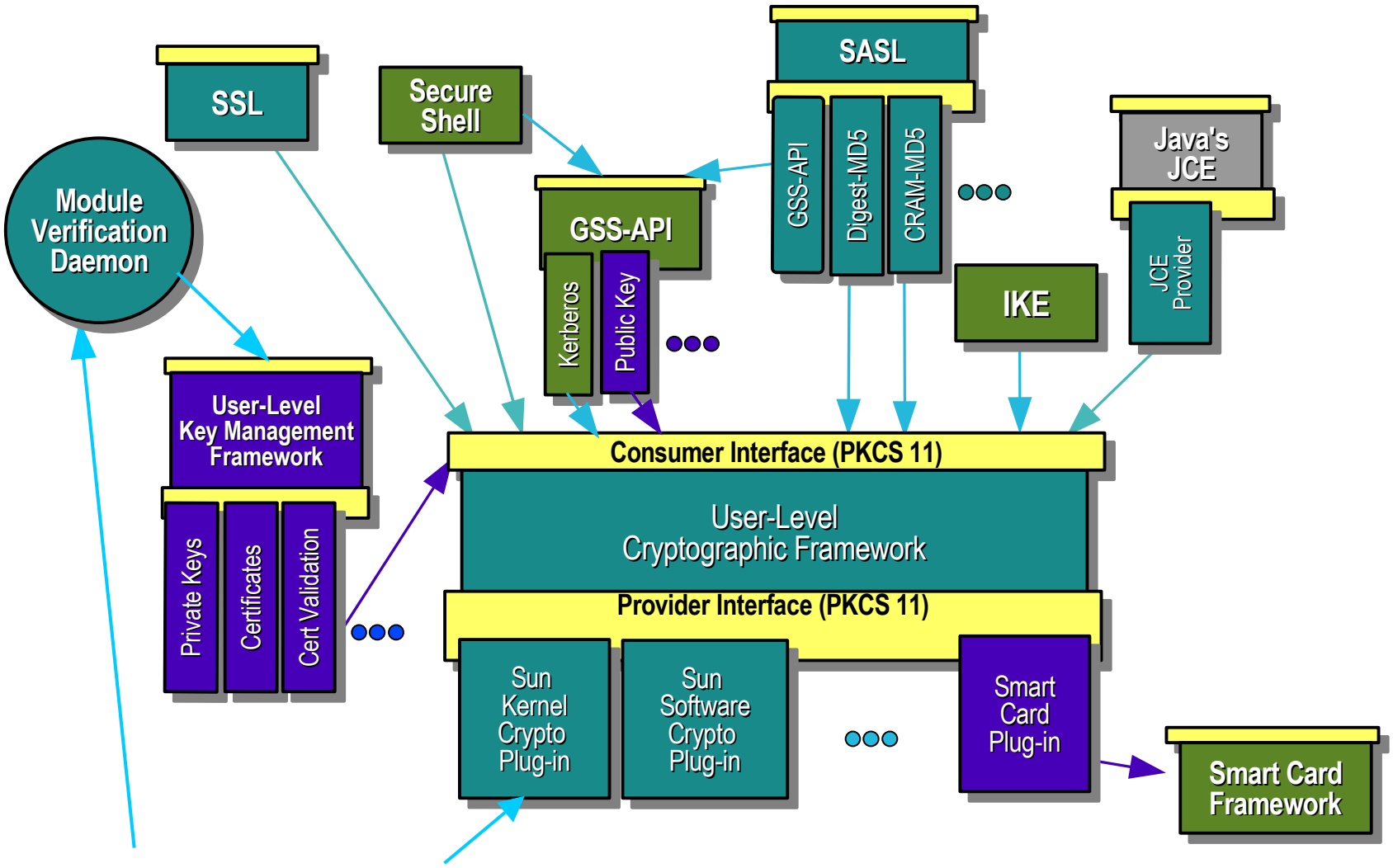
Network Security Architecture



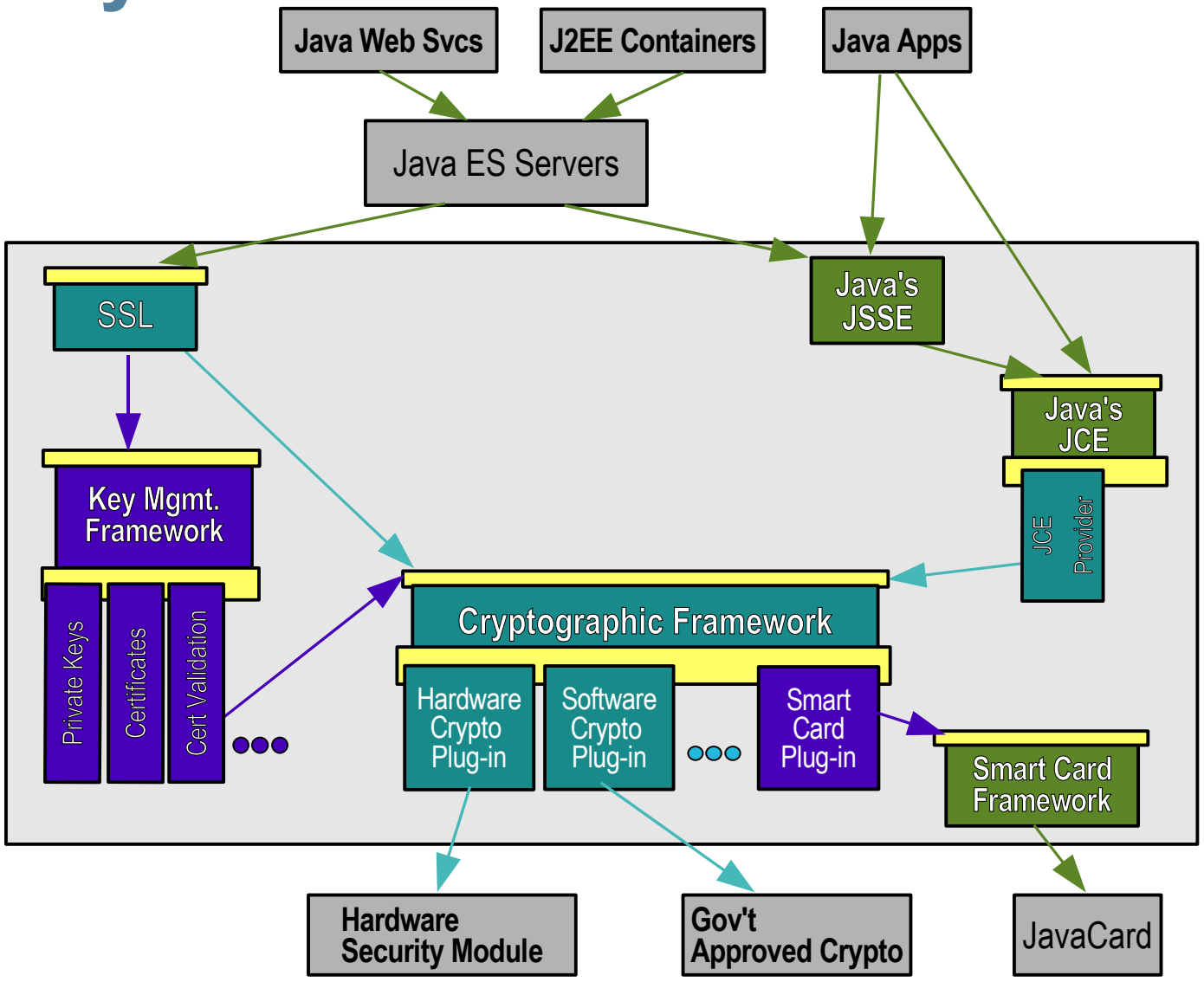
Network Security Architecture - Kernel



Network Security Architecture - User



Security Platform for Web Services



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  control:28dd3a3af2fcc103f422993de5b162f3
                 test:28893a3af2fcc103f422993de5b162f3
```

[1] See: Sun BluePrint: Automating File Integrity Checks, http://www.sun.com/blueprints/0305/819-2259.pdf

Solaris Audit

- Kernel auditing of system calls and administrative actions.
  - Can record events happening in any zone (from the global zone).
  - 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:

```
$ auditreduce -m AUE_su -r joe | praudit -s
   file,2005-04-12 07:25:06.000 -04:00,
   header,97,2,AUE_su,,10.8.31.9,2005-04-12 07:28:30.220 -04:00
   subject,joe,joe,other,joe,other,1834,3097759606,12114 22
   10.9.1.3
   text,bad auth. for user roleB
   return,failure,2
```

Example taken 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 additional software packages added to a standard Solaris 10 system.
- A set of label-aware services which implement multilevel security.

Extending Solaris 10 Security Features

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

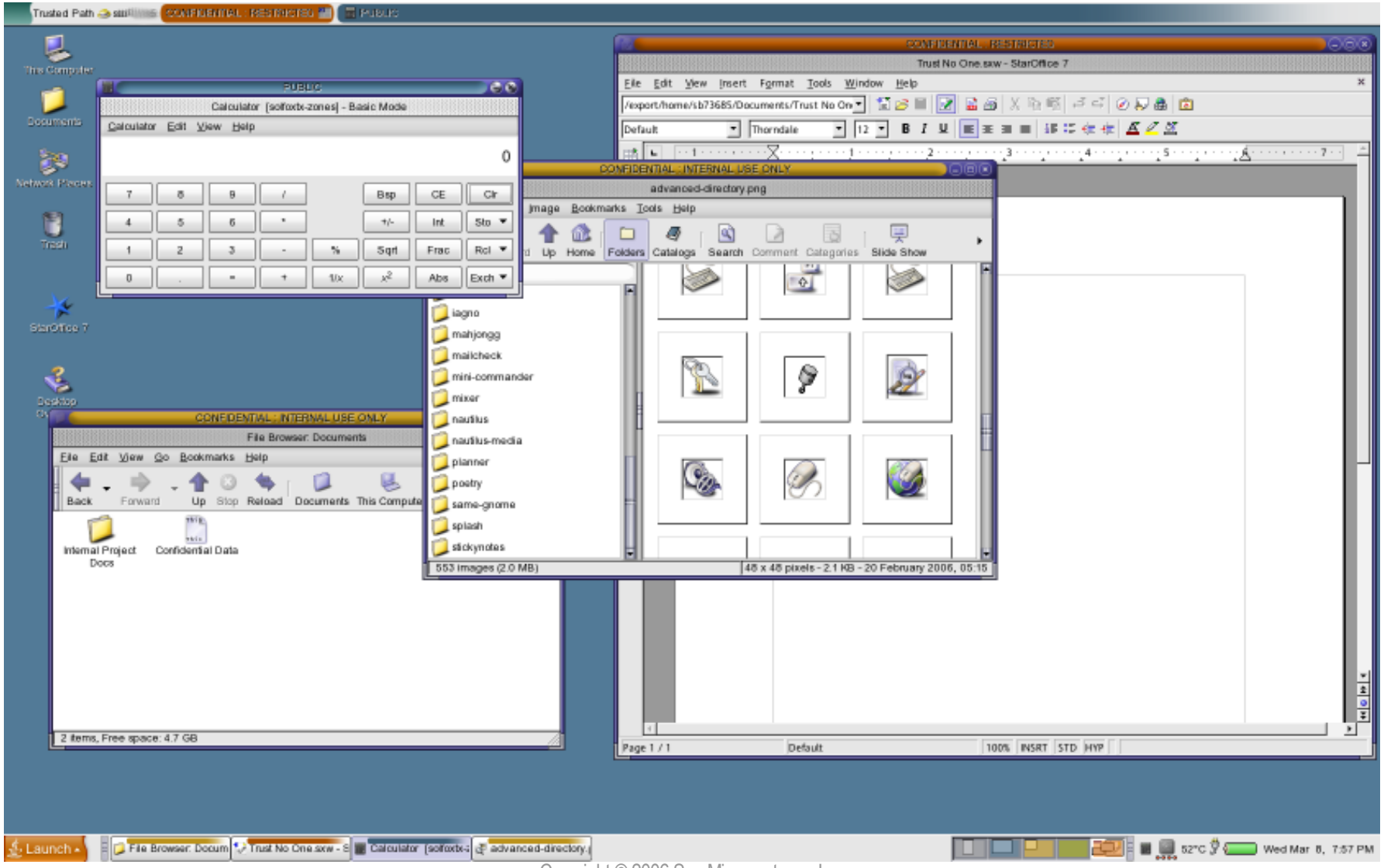
Trusted Extensions in a Nutshell

- 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.
- Administrators utilize Solaris Roles for duty separation.
  - Installation, System Admin., Security Admin., etc.

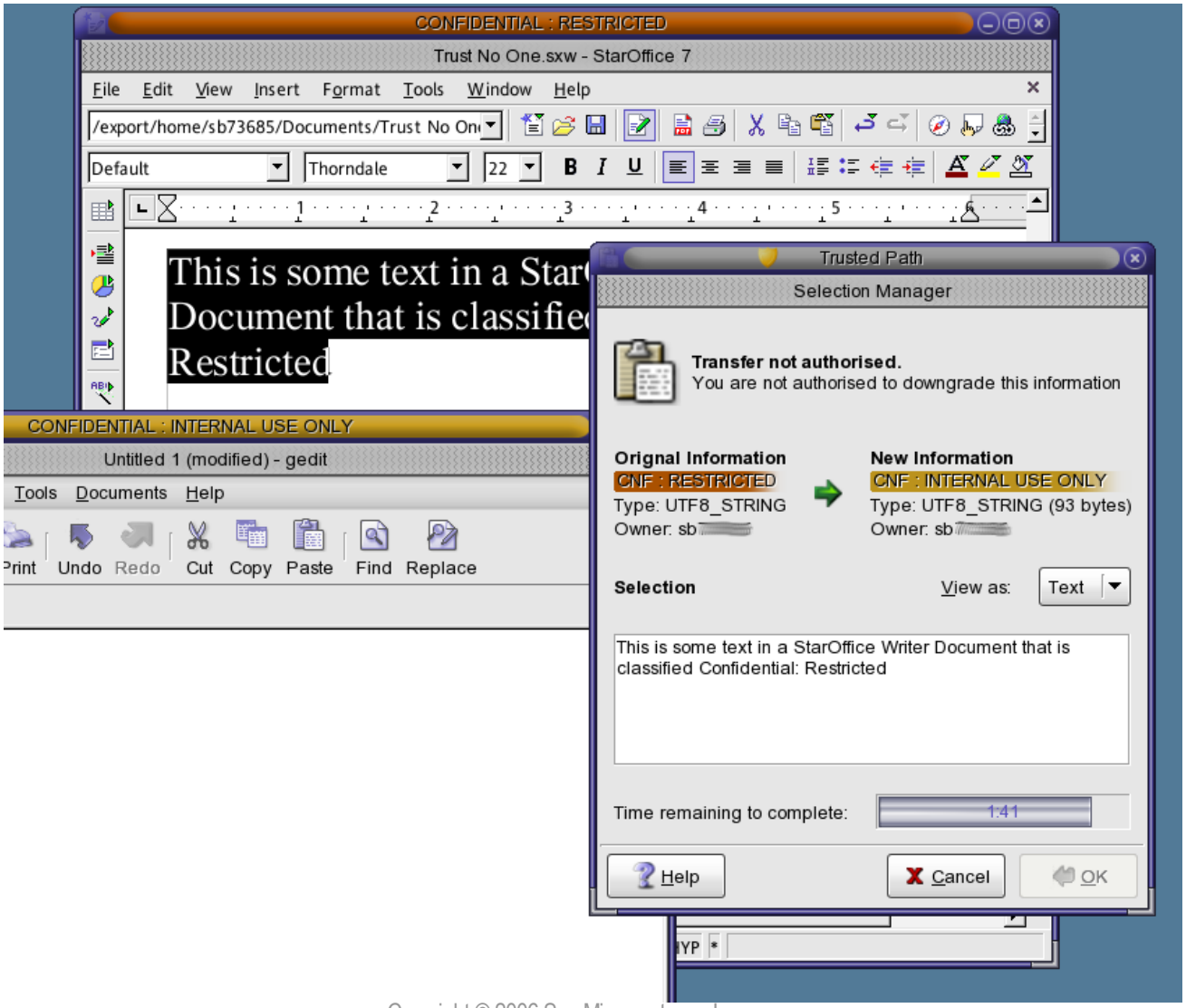
What are Label-Aware Services?

- 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



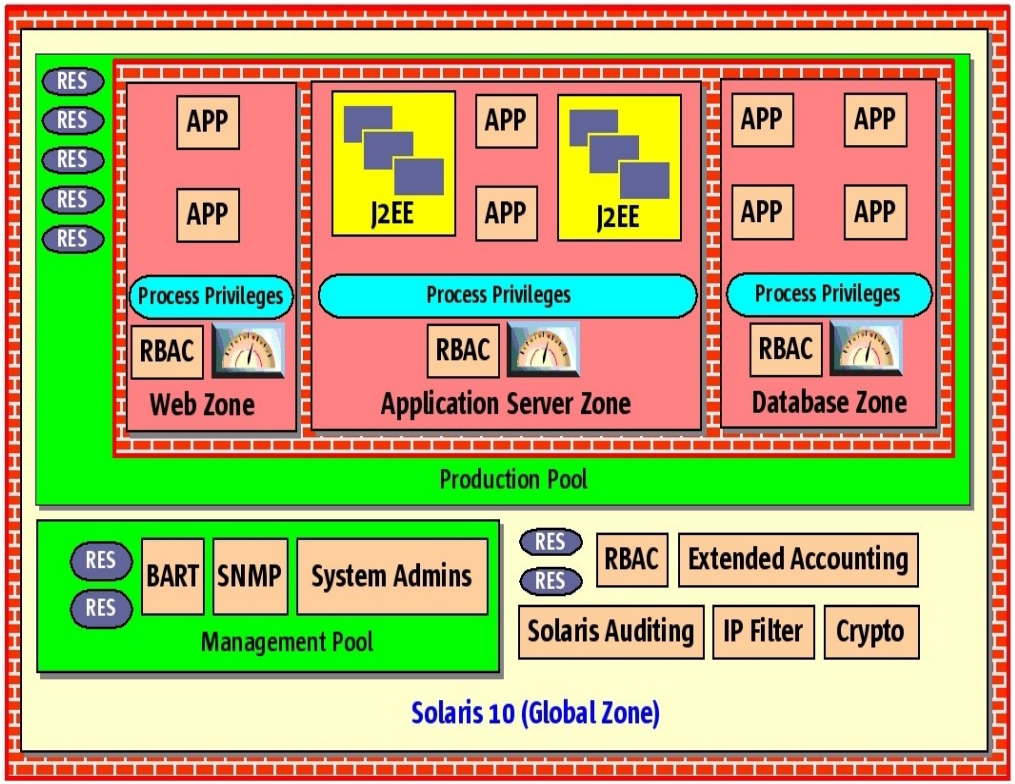
Mandatory Access Control



Putting It All Together

Solaris 10 Security – A Secure Foundation for Success:

- Reduced Networking Meta Cluster
- Signed Binary Execution
- Solaris Security Toolkit
- Secure Service Management
- User Rights Management
- Process Rights Management
- Resource Management
- Kerberos, SSH, IPsec
- Cryptographic Framework
- Containers / Zones
- IP Filter, TCP Wrappers
- Auditing, BART
- Trusted Extensions



But wait! There's more!

- Auditing Improvements
  - Audit Trail Noise Reduction
  - Audit Event Reclassification
- Enhanced TCP Wrappers Support
  - Now integrated with rpcbind and sendmail
- New Mount Options
  - noexec, nodevices
- User Process Visibility Restrictions
- vacation(1) Mail Filtering

and more...

- “root” GID is now “0” (root) not “1” (other)
- IPsec NAT Traversal
- RIPv2 Protocol Support
- 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...

and more...

- NFSv4
  - Support for GSS_API, ACLs, etc.
- Sendmail 8.13 (8.13.8 in OpenSolaris)
  - Support for rate limiting and milters.
- BIND 9
  - DNSSEC, Views, IPv6 Support
- Java 1.5 Security
  - 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...

OpenSolaris Contributions

- ZFS (S10U2)
- Cryptographic Framework Metaslot (S10U1)
- Kernel SSL Proxy (S10U2)
- IKE Support for NAT-T (RFC 3947 and RFC 3948) (S10U1)
- Randomized TCP/UDP Ephemeral Port Selection
- Persistent Static Routes
- Sendmail TLS Support (S10U1)
- elfsign(1) Token Support
- Kerberos 1.4 Resync
- Java 1.6

Actions...

1. Evaluate, pilot and use Solaris 10 Today!
2. Join the OpenSolaris Community!
3. 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
- Mark Thacker
- Gary Winiger

Solaris 10 Security
Technical Deep Dive

Glenn Brunette

Sun Microsystems, Inc.

glenn.brunette@sun.com

http://blogs.sun.com/gbrunett