

Practical Solaris 10 Security

Franz Haberhauer

Technical Director

Global Systems Engineering

Sun Microsystems GmbH



Security Goals - Defensive

- Provide strong assurance of **system integrity**
 - > Simplify building and deploying of secure solutions
 - > Monitor system state for unexpected change
 - > Audit security relevant changes
- **Defend system** from unauthorized access
 - > Contain damage caused by unauthorized access
 - > Minimize privileges given to people and processes
 - > Filter inbound communications into the system

Security Goals - Enabling

- **Secure authentication** of all active subjects
 - > Use strong user and host level authentication
 - > Integrate authentication mechanisms
 - > Leverage a unified authentication infrastructure
- **Protect communications** between endpoints
 - > Provide private data transmissions
 - > Verify integrity of received data
 - > Securely establish and protect keys

Security Goals - Deployable

- Emphasize **integratable stack** architecture
 - > Enable pluggable use of 3rd party security providers
 - > Provide abstracted APIs for customers
 - > Offer robust security platform for Sun's products
- **Interoperable** with other security architectures
- **Ease management** and use of security features
 - > Transparently maintain security infrastructure
 - > Simplify and centralize security policy definition
 - > Minimize visibility of secure features to end users
- Receive **independent assessment** of security

Agenda

- Attacker Goals
- Attack Scenario Background
- Attack Defense Scenario
- Attack Detection Scenario

Attacker Goals

- Local System Access
- Administrative Privileges
- Access Privileged Information
- Conceal Attack and Avoid Detection
- Inject, Modify or Destroy Local Content
- Staging Platform for Further Attacks

Attack Scenario Background

- While operating from the network:
 - > Attack originates from a local or remote network.
 - > Attacker does not have local system access.
- While operating from the local system:
 - > Attacker has obtained command line access (unprivileged account).
- In Both Cases:
 - > Attack takes place against a Solaris 10 non-global zone.
 - > Solaris 10 global zone == “service processor”

Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)
- Process Rights Management
- Non-Executable Stack
- Pluggable Authentication Mechanism (PAM)
- Reduced Installation Profile
- Solaris Zones
- Solaris Cryptographic Framework
- User Rights Management

Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.

IPFilter

IPFilter

- Open Source Kernel Packet Filter in Solaris 10
 - > ressourcensparend implementiert
 - > begrenzt als End-point-FireWall konfigurierbar
 - > kein HA, kein stealth Mode, keine Verschlüsselung möglich
 - > basiert auf Open Source IP Filter
 - > Darren Reed (nun Sun Mitarbeiter)
 - > <http://coombs.anu.edu.au/~avalon/ip-filter.html>
- Zur Absicherung von Systemen
 - > z.Zt. nicht zwischen lokalen Zonen anwendbar (Shared IP Stack – Crossbow: Stack Instances in NV!)
- Built in Netzwerk und Port Address Translation (NAT/PAT)

IPFilter Komponenten

- Kernel modules
 - > ipf – packet filtering
 - > pfil – STREAMS driver
- Konfiguration
 - > Text-basierte Konfiguration (last match)
 - > /etc/ipf/pfil.ap
 - > /etc/ipf/ipf.conf
 - > /etc/ipf/ipnat.conf
 - > /etc/ipf/ippool.conf
 - > Filter by: IP Addr (src,dst), Port, Direction, ...
 - > Regeln: block, pass, logging von Paketen

IPFilter Administration

- ipf(1M) – Packet Filter Konfiguration laden
- ipnat(1M) – NAT Konfiguration
- ipmon(1M) – Monitor logged packets
- ipfstat(1M) – Anzeige von Statistiken und Filtern
- ippool(1M) – Laden von ippools

IPFilter benutzen (1)

- Standard: IPFilter abgeschaltet
- Auf welchem Interface soll IPFilter filtern ?
 - > /etc/ipf/pfil.ap eintragen
- Packet Filter Module laden
 - > svcadm restart network/pfil
- FireWall Regeln eintragen
 - > /etc/ipf/ipf.conf
 - > z.B. block in any
 - > z.B. pass in any
 - > Beispiele siehe /usr/share/ipfilter/examples

IP Filter Beispiel

- /etc/ipf/ipf.conf

```
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
```

IPFilter benutzen (2)

- IPfilter einschalten
 - > `svcadm enable network/ipfilter`
- Interface neu initialisieren
 - > `ifconfig bge0 unplumb`
 - > `ifconfig bge0 plumb 1.2.3.4 netmask 255.255.255.0 up`
- Nach Änderungen die Regeln neu laden
 - > `ipf -FA -f /etc/ipf/ipf.conf`
 - > `ipnat -CF -f /etc/ipf/ipnat.conf`

Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)

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)
 - `/var/svc/profile`
- 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.

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

```
# net services
```

```
net services: usage: net services [ open | limited ]
```

```
# net services 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
```

Secure By Default

Disabled Services

Service	FMRI
dtprintinfo	svc:/application/cde-printinfo
CDE subprocess control	svc:/network/cde-spc
DMI	svc:/application/management/dmi
SNMP	svc:/application/management/sma
Solstice Enterprise Agent	svc:/application/management/snmpdx
Seaport	svc:/application/management/seaport
X font server	svc:/application/x11/xfps
Internet print protocol	svc:/application/print/ipp-listener:default
SVM remote metaset	svc:/network/rpc/meta
SVM remote mediator	svc:/network/rpc/metamed
SVM remote multihost disk	svc:/network/rpc/metamh
SVM communication	svc:/network/rpc/mdcomm
rstatd	svc:/network/rpc/rstat:default
rusersd	svc:/network/rpc/rusers:default
telnetd	svc:/network/telnet:default
statd	svc:/network/nfs/status
lockd	svc:/network/nfs/nlockmgr
NFS client	svc:/network/nfs/client
NFS server	svc:/network/nfs/server
rquotad	svc:/network/nfs/rquota
NFS v4 callback daemon	svc:/network/nfs/cbd
NFS id mapping	svc:/network/nfs/mapid
ftpd	svc:/network/ftp:default
fingerd	svc:/network/finger:default
rlogind	svc:/network/login:rlogin
rshd	svc:/network/shell:default

Secure By Default

Restricting Properties

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

- > `svcadm enable ftpd`
- > `svccfg -s sendmail setprop config/local_only = false`

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 Execution Context Example

```
# svccprop -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_privad
dr
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>

Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)
- Process Rights Management

**Process
Privileges
=
Least
Privileges**

Process Rights Management

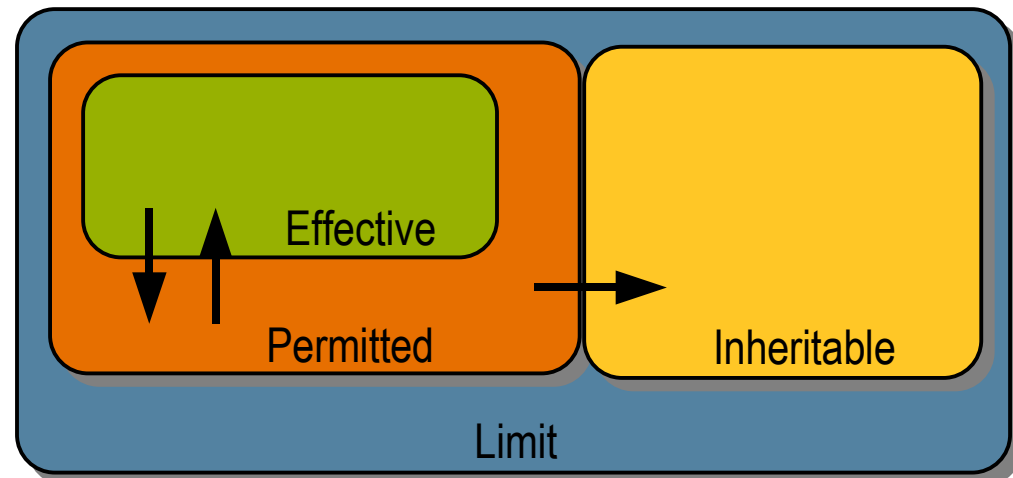
- = Least Privileges
- minimale Privilegien für Prozesse
 - > Aufgabe von "alles oder nichts" Rechtevergabe
 - > root vs. Rest der Nutzer
 - > meist wird nur ein Bruchteil benötigt
 - > Device Zugriff
 - > reservierte Netzwerkports
 - > RT Priorität

Process Rights Management (2)

- Aufspaltung in verschiedene Privilegien
 - > zur Zeit 67 Privilegien
- Abfrage von Privilegien, nicht UID=0
- Individuelle Privilegien können ein- und ausgeschaltet werden
- Integration mit RBAC
- Seit Solaris Express 02/2003
- Herausforderung:
 - > Rückwärtskompatibilität

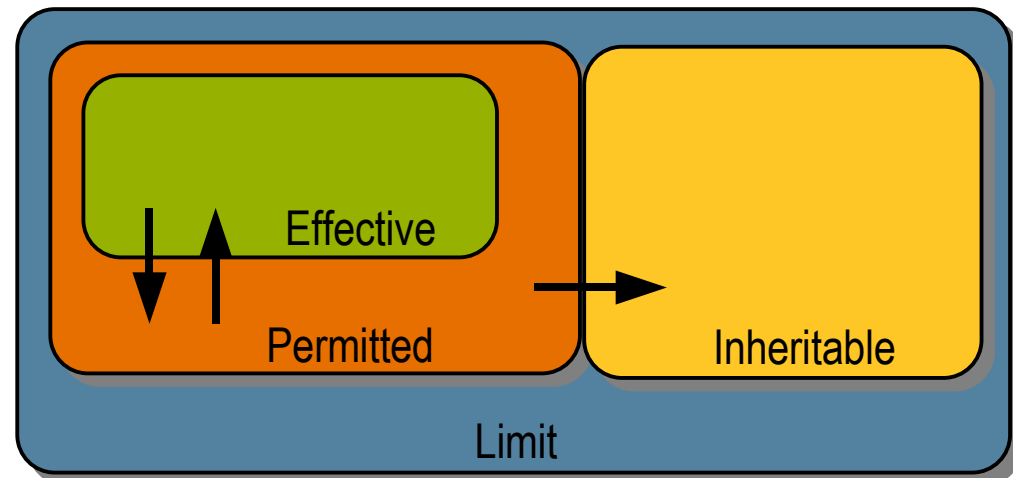
Privilegien Sets

- 4 Privilegien Sets je Prozess
 - > E - ffective set
 - > in Benutzung
 - > P - etermined set
 - > Privilegien, die benutzbar wären
 - > Maximumangabe für E
 - > I - nheritable set
 - > vererbare Privilegien
 - > L - imited set
 - > benutzbare Privilegien



Privilegien Sets (2)

- Privilegien zu P, E, I hinzufügbbar
- löschen aus L,I,E,P
 - > löschen aus P löscht aus E
 - > löschen aus L wird erst mit exec aktiv
- `privileges(5)`
- `ppriv(1)`
- `ppriv -l -v`
zeigt alle Privilegien



Privilegien anzeigen

```
$ 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
```

Privilegien anzeigen

- `ppriv -v $$`
 - > zeigt die Privilegien eines Prozesses

```
ppriv -v 470
470: /usr/lib/sendmail -Ac -q15m
flags = <none>
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,proc_exec,proc_fork,proc_info,proc_session
L: 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_link_any,file_owner,
file_setid,ipc_dac_read,ipc_dac_write,ipc_owner,net_icmpaccess,net_privaddr,net_rawaccess,proc_audit,
proc_chroot,proc_clock_highres,proc_exec,proc_fork,proc_info,proc_lock_memory,proc_owner,proc_prioctl,
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
```

Arbeit mit Privilegien

- `ppriv -e -D cat /etc/shadow`

```
ppriv -e -D cat /etc/shadow
```

```
cat[586]: missing privilege "file_dac_read" (euid = 100, syscall = 225) needed at ufs_iaccess+0xd2
```

```
cat: cannot open /etc/shadow
```

- Modifikation von Privilegien
 - > `ppriv -s EIP=basic,file_dac_read <pid>`
 - > PE oder I oder PEI
- Ausführung von Kommandos mit Privilegien - z.B. via RBAC
 - > `pfexec -P <privs> <cmd>`
- Modifikation von user Standardset
 - > `usermod -K defaultpriv=basic,file_dac_read <user>`

Privilegien und Zonen

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_prioctl
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	

Legende

a = mandatory
a = optional
a = nur global
a = Standard
a = TX

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]
```

<u>STAT</u>	<u>TIMESTAMP</u>	<u>PPID</u>	<u>PID</u>	<u>PRIV</u>	<u>CMD</u>
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://opensolaris.org/os/community/security/projects/privdebug/>

Sun BluePrints Onlinep - February 2006

PRIVILEGE DEBUGGING IN THE SOLARIS™ 10 OPERATING SYSTEM

<http://www.sun.com/blueprints/0206/819-5507.pdf>

Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)
- Process Rights Management
- Non-Executable Stack

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
```

Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)
- Process Rights Management
- Non-Executable Stack
- Pluggable Authentication Mechanism (PAM)

Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)
- Process Rights Management
- Non-Executable Stack
- Pluggable Authentication Mechanism (PAM)
- **Reduced Installation Profile**

Platform Minimization

- Process of installing a system with only the software necessary to support its designated business function.
- How is it done?
 - Install smaller Solaris OE meta-cluster.
 - Add packages that you need.
 - Remove packages that you don't (optional)*.

Note: Removing packages may impact the supportability of the system. Always exercise care!

SunSolve Infodoc 86177 Support of mimized Systems

Platform Minimization

- Pros:
 - Fewer software packages to secure/maintain.
 - Fewer software packages to patch.
 - Fewer software packages to exploit.
- Cons:
 - Lack of mature minimization tools.
 - Lack of documented software dependencies.
 - Lack of QA'ed minimized configurations.
 - Lack of support for minimized builds.

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

Platform Hardening

- Process of configuring a system to improve its overall security posture.
- How is it done?
 - Apply security patches or updates.
 - Disable unnecessary services.
 - Enable additional security functions
 - logging, auditing, password aging, etc.
 - Install additional software components
 - Secure Shell (pre-S9), Fix-Modes, etc.
 - Built into Solaris 9 and 10

Platform Hardening

- Pros:
 - Greater security due to fewer attack vectors.
 - Greater accountability through auditing.
 - Greater compliance with commonly accepted platform security practices.
- Cons:
 - Lack of documented software dependencies.
 - Lack of QA'ed hardened configurations.
 - Lack of consistent support for hardened builds.
 - Log generation impact on system resources.

Solaris Security Toolkit (SST)

- Codification of the security recommendations as documented by the Sun BluePrints program.
- Collection of Shell scripts used to harden/audit the configuration of systems running Solaris OE.
- Flexible and extensible framework for rapidly hardening/auditing platforms in accordance with a defined security policy.
- Mechanism for hardening/auditing platforms in a repeatable, reliable manner for one system or for thousands.

Solaris Security Toolkit History

- Originally called the JumpStart Architecture and Security Scripts (JASS) Toolkit.
- Based on the Sun BluePrints:
 - Solaris OE Security
 - Solaris OE Network Settings for Security
 - Auditing in the Solaris 8 OE
- Supports Solaris 2.5.1 and later.
- Supports SPARC and Intel.
- Supports Trusted Solaris 8.
- SST 4.2 fully supported on Solaris 8, 9, and 10
- <http://www.sun.com/software/security/jass/>



Solaris Security Toolkit

- Download
- Verify Fingerprint
- Installation
- Usage: Hardening (Standalone)
- Usage: Hardening (JumpStart)
- Usage: Undo
- Usage: Auditing
- Usage: Bundled Utilities

Hardening (Standalone)

1. Reboot the system*
 2. Test and validate the system.
 3. Create or select a Security Profile
 4. Customize the Security Profile
 5. Apply the Security Profile
 6. Reboot the system.
 7. Test and validate the system.
- * The system should reboot cleanly without any errors. Resolve issues before continuing.

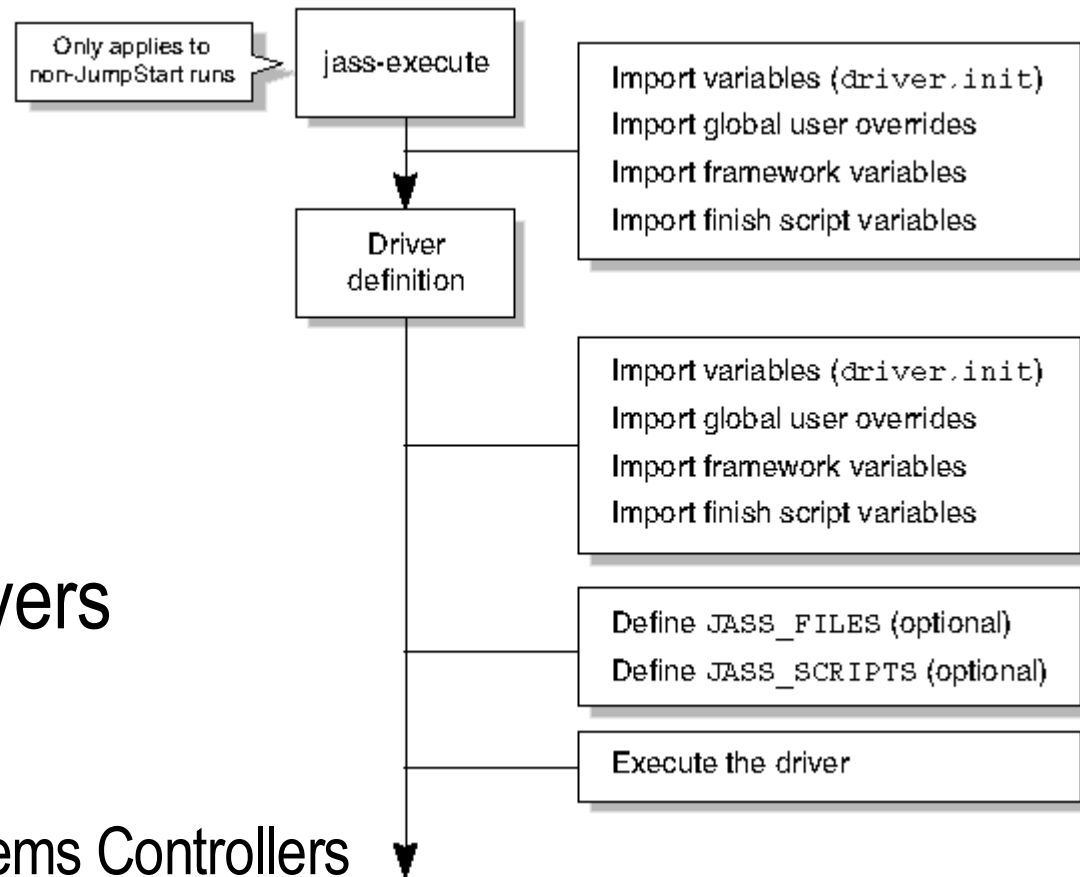
A Customizable Framework

• Customizable Driver Scripts

- # disable-ab2.fin
- # disable-apache.fin
- # disable-apache2.fin
- # disable-appserv.fin
- # disable-asppp.fin
- # disable-autoinst.fin
- # disable-automount.fin
- # disable-dhcpd.fin
- ...

• Product-Specific Drivers

- Server Systems
- Sun Cluster 3.x
- SunFire High-End Systems Controllers



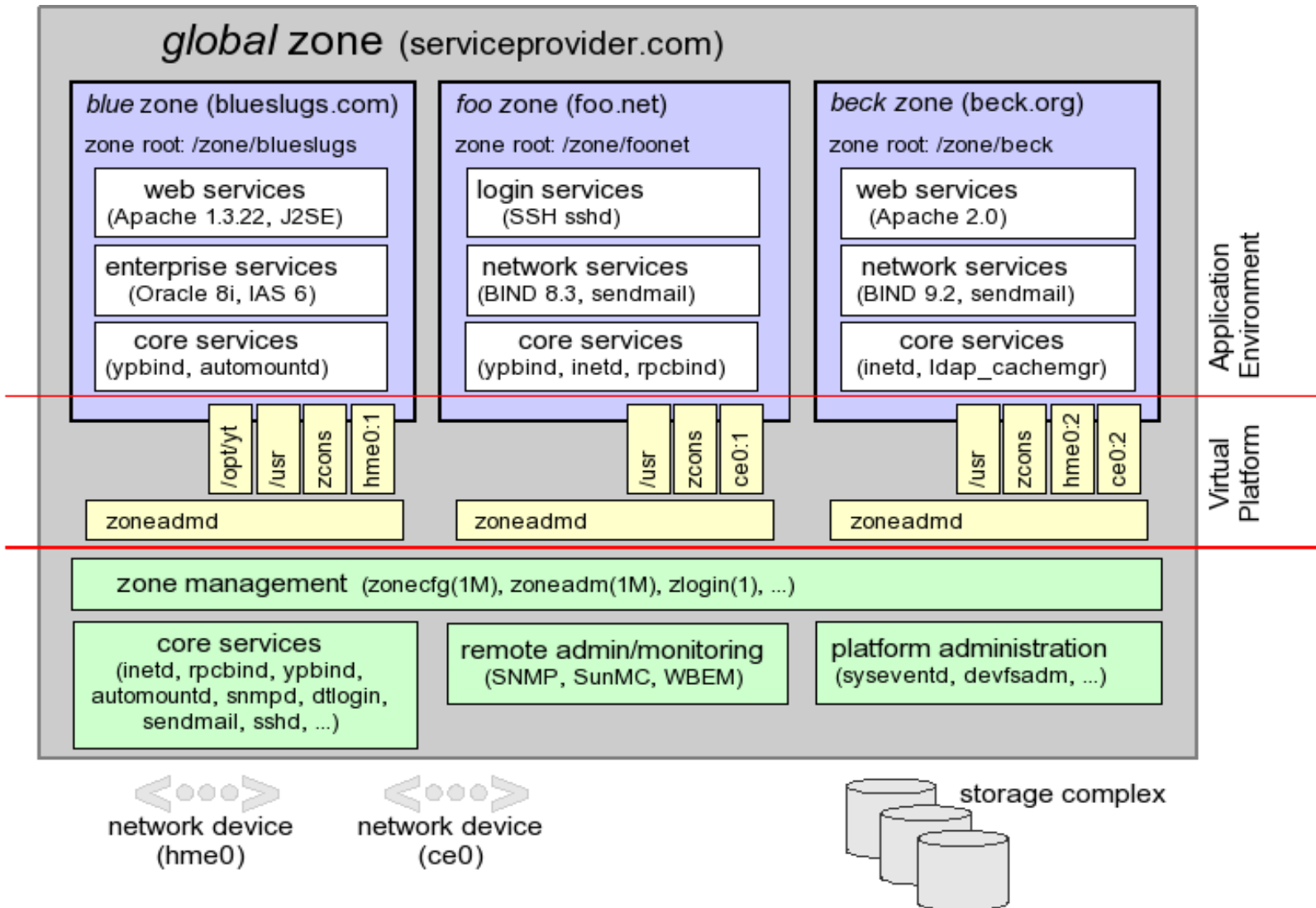
Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)
- Process Rights Management
- Non-Executable Stack
- Pluggable Authentication Mechanism (PAM)
- Reduced Installation Profile
- Solaris Zones

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).
- Zones operate with fewer privileges (default).

Zones



Zones Security #2

- By default, global zone “root” can see and do anything.
- Local zones are restricted in order to protect the security of the system:
 - > System Calls
 - > Device Manipulation
 - > Privileges
 - > Resources

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 *nodedevices* 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_prioctl
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

```
zonecfg -z myzone
> set limitpriv=
    "default,sys_time"
> exit
```

Zones Security 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 Security Example #2

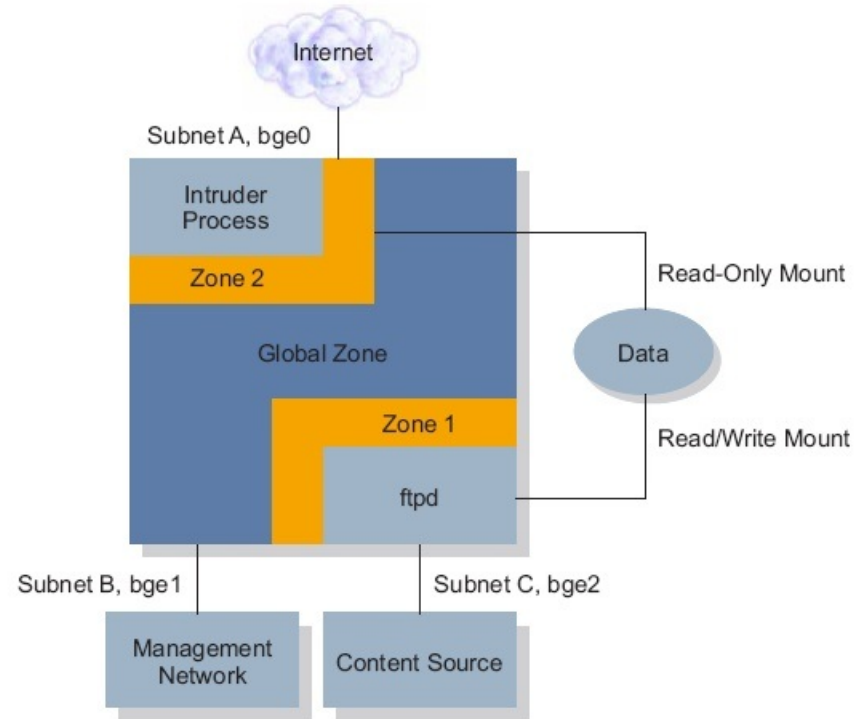
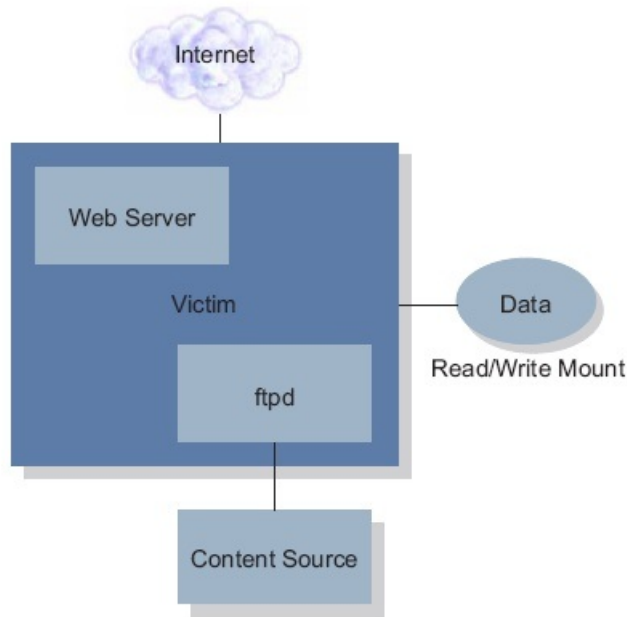
```
# mount -p
/          -    /          zfs      - no rw,devices,setuid,exec,atime
/dev       -    /dev       lofs     - no zonedevfs
/lib       -    /lib       lofs     - no ro,nodevices,nosub
/platform  -    /platform lofs     - no ro,nodevices,nosub
/sbin     -    /sbin     lofs     - no ro,nodevices,nosub
/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
```

Unique Use Cases for Containers

Content-Upload for a Webserver



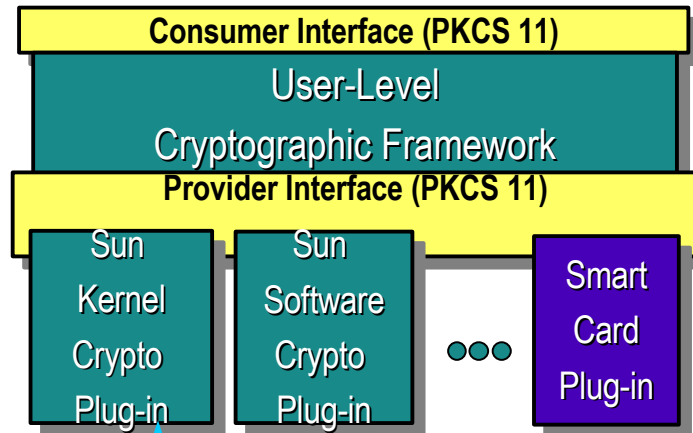
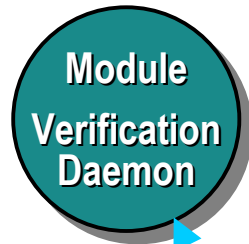
Attack Defense Scenario

- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)
- Process Rights Management
- Non-Executable Stack
- Pluggable Authentication Mechanism (PAM)
- Reduced Installation Profile
- Solaris Zones
- Solaris Cryptographic Framework

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, 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 and the Sun CryptoAccelerator 6000

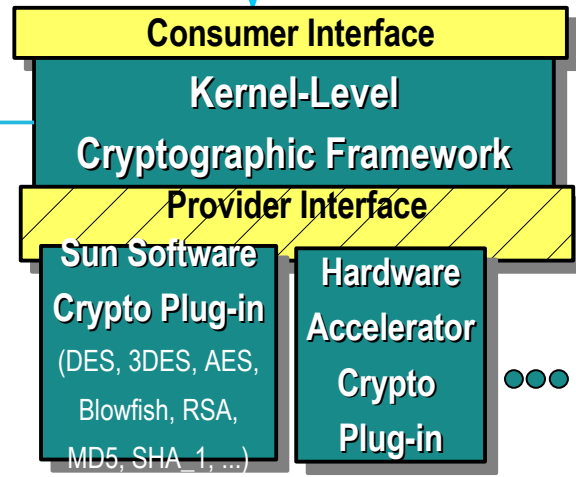
Crypto Framework Architecture



`/dev/cryptoadm`

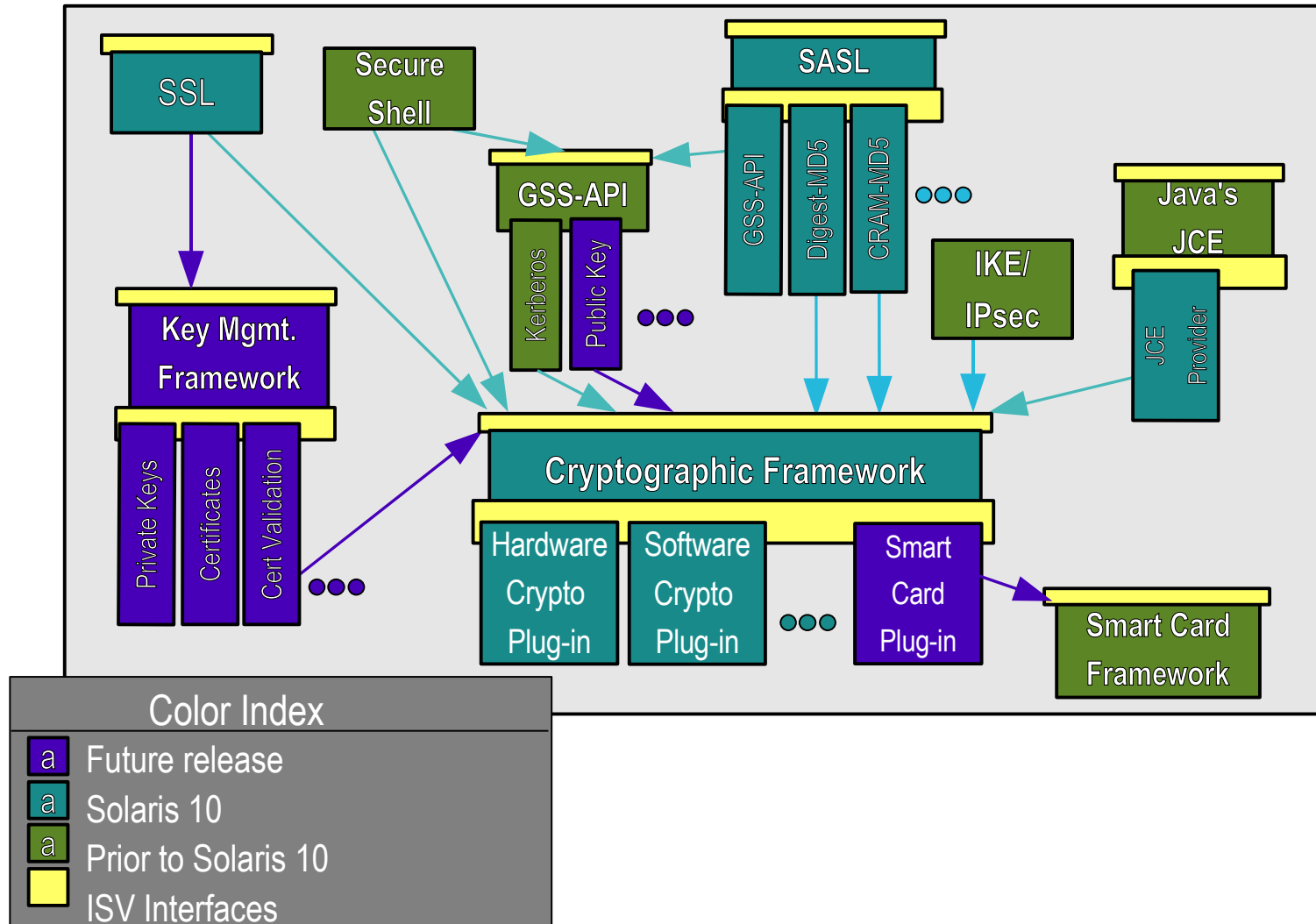
`/dev/crypto`

Kernel-Level

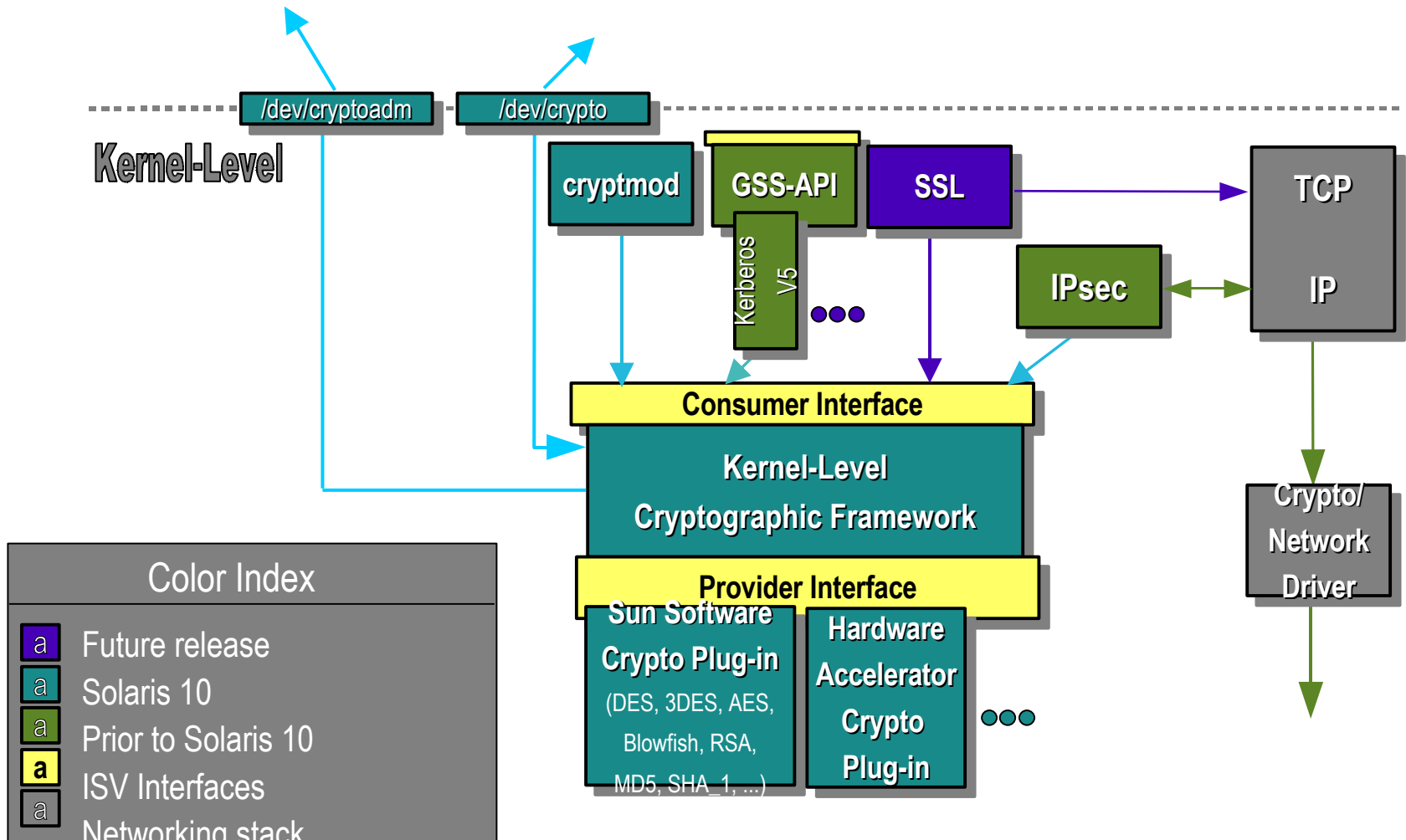


Color Index	
a	Future release
a	Solaris 10
a	Prior to Solaris 10
a	ISV Interfaces

Network Security Architecture

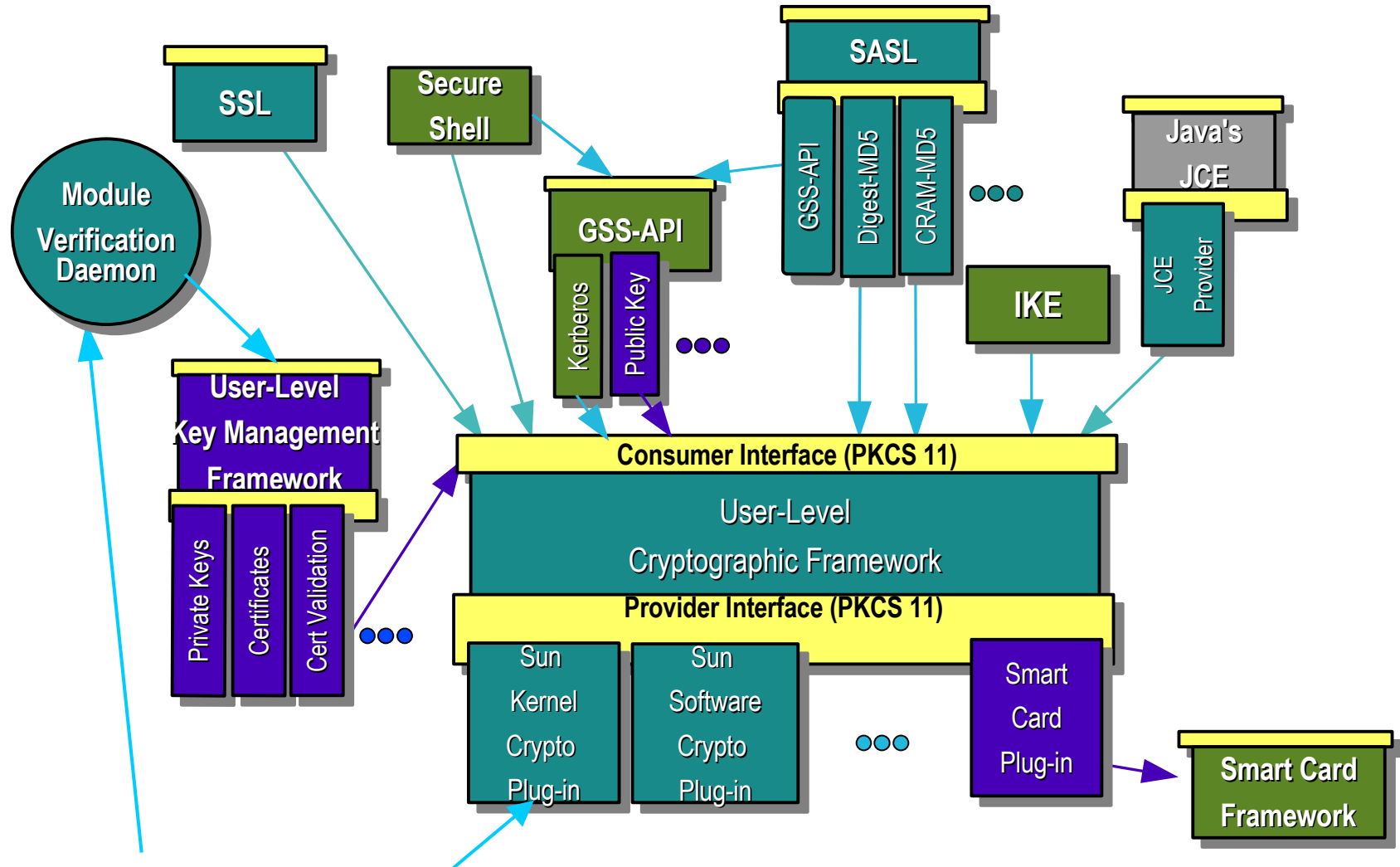


Network Security Architecture - Kernel

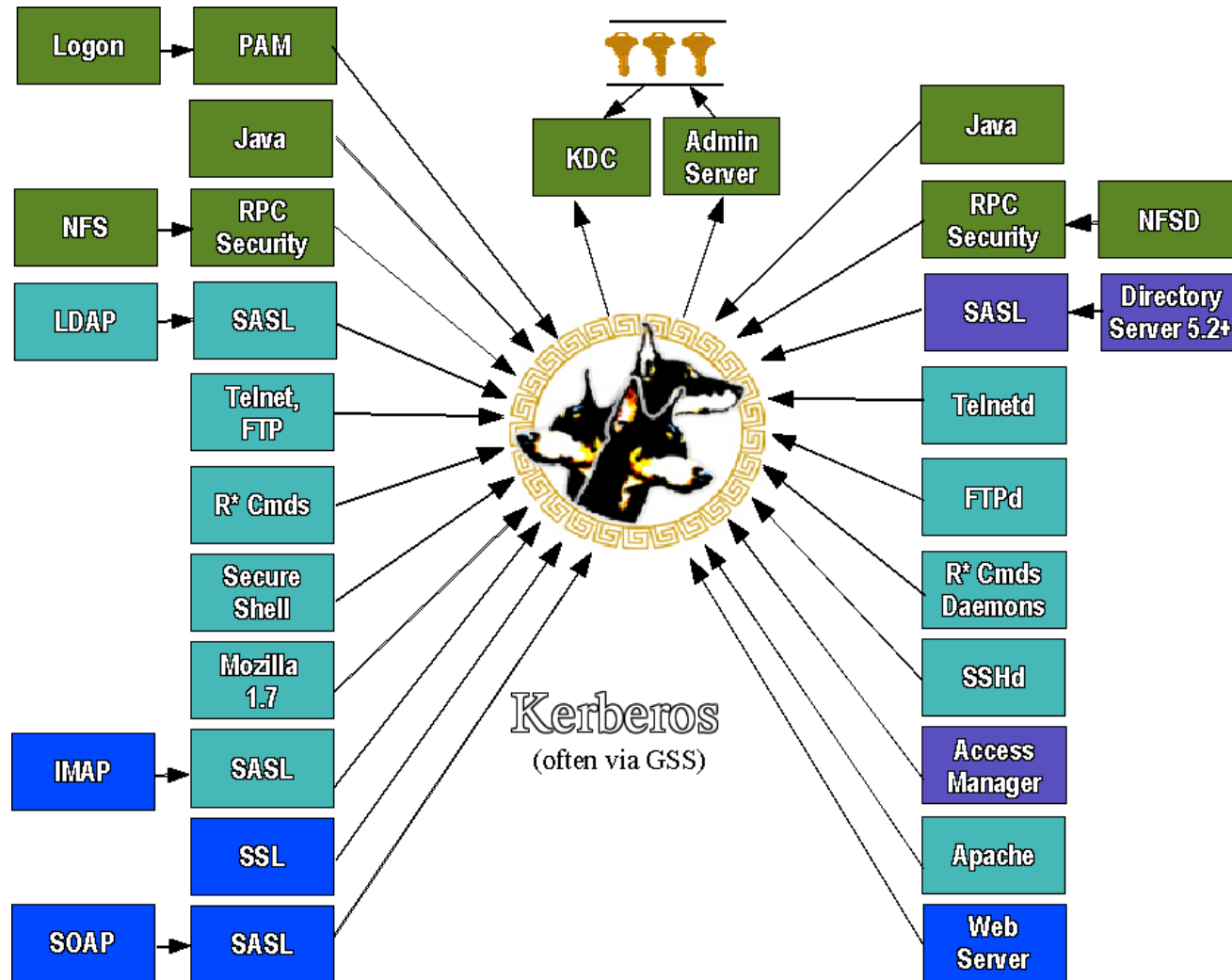


Color Index	
a	Future release
a	Solaris 10
a	Prior to Solaris 10
a	ISV Interfaces
a	Networking stack

Network Security Architecture - User



Kerberos Ecosystem Progress



Attack Defense Scenario

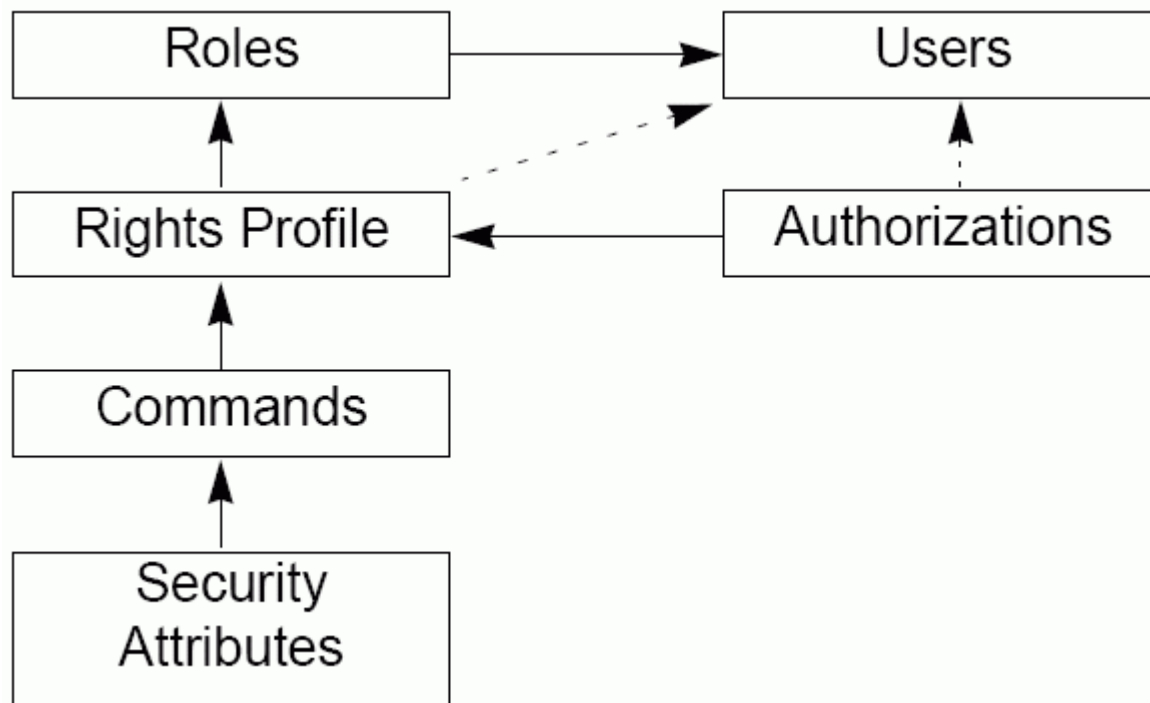
- IP Filter, TCP Wrappers, IPsec, etc.
- Service Management Facility (SMF)
- Process Rights Management
- Non-Executable Stack
- Pluggable Authentication Mechanism (PAM)
- Reduced Installation Profile
- Solaris Zones
- Solaris Cryptographic Framework
- **User Rights Management**

Role Based Access Control (RBAC)

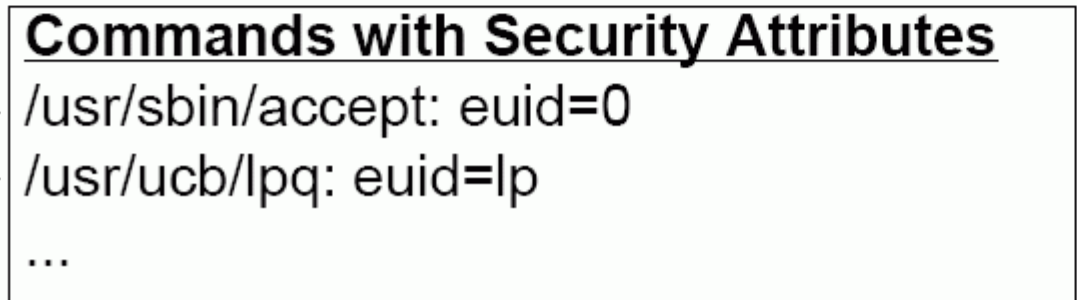
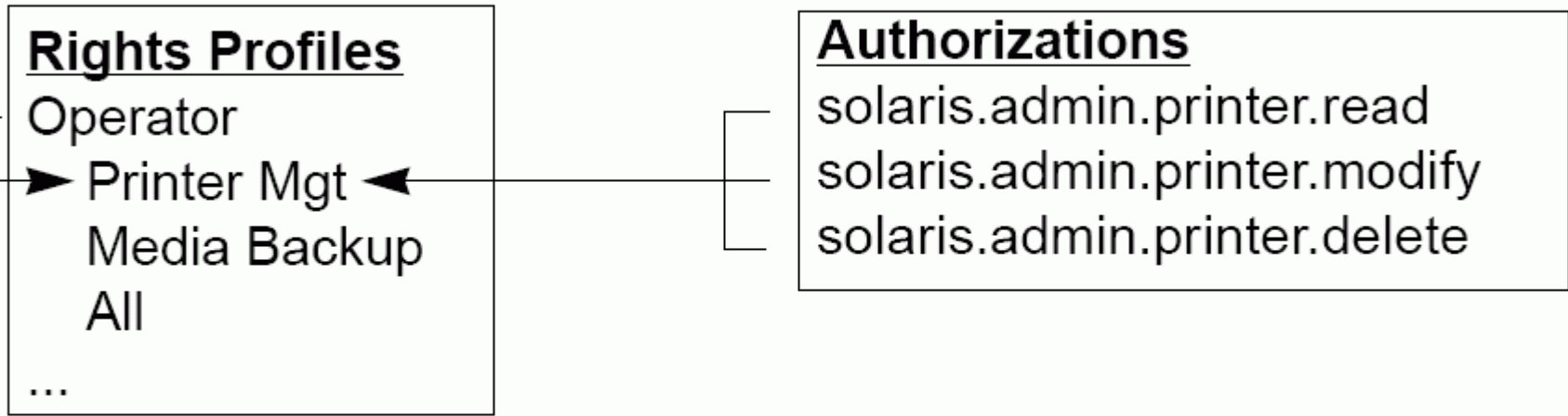
RBAC - Überblick

- RBAC realisiert ein Framework zum Delegieren von Sysadmin Aufgaben an Nutzer durch Festlegung von Rollen
 - > ohne Kenntnis des root Passworts
 - > für begrenzten Personenkreis
 - > Mehrstufige Logins erzwingen
 - > z.B. erfordere 4-Augen Prinzip für Nutzeranmeldung
- Seit Solaris 8

Role Based Access Control (RBAC)



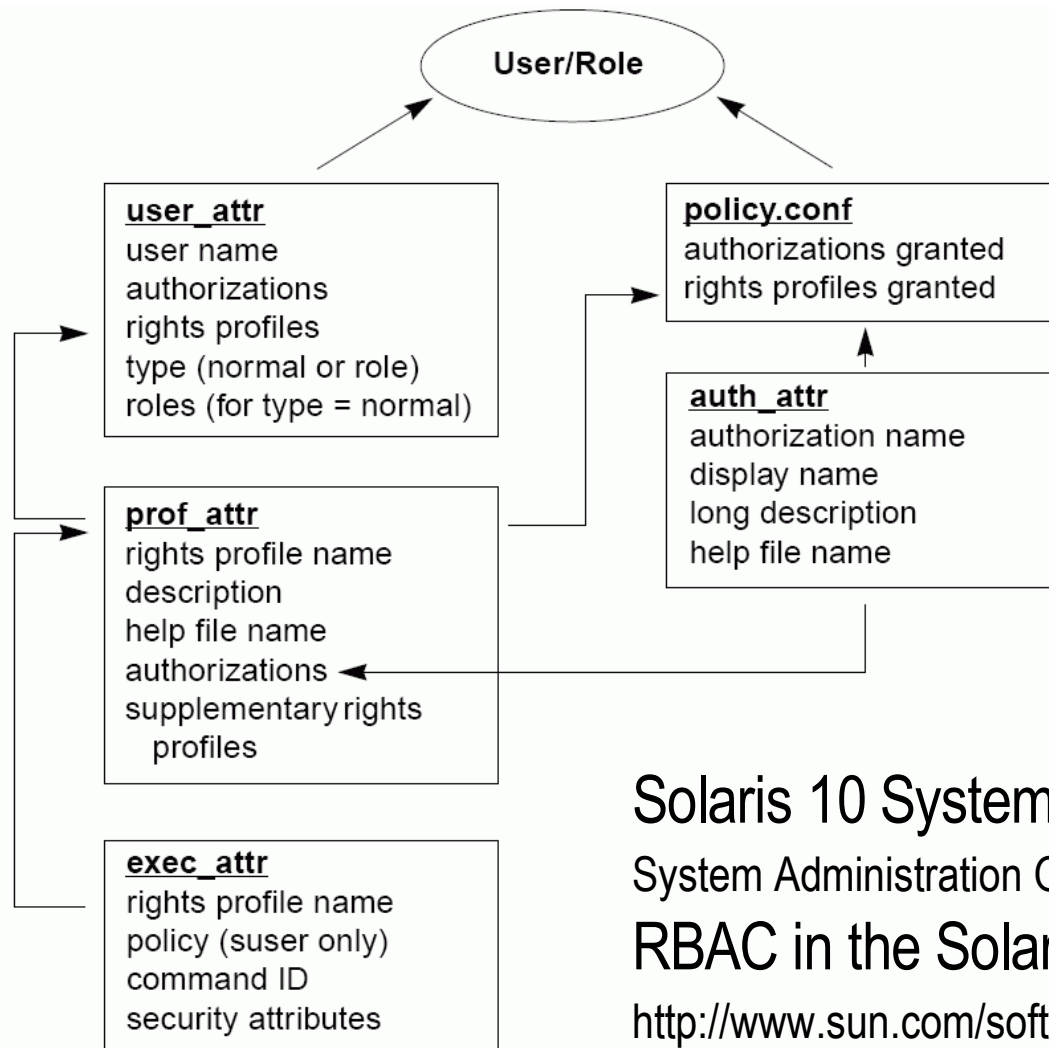
Role Based Access Control (RBAC)



RBAC Begriffe (1)

- Execution Profile
 - > /etc/security/prof_attr legt Profiles mit Namen fest
 - > /etc/security/exec_attr ordnet Attribute und Anwendungen den Profilen zu
- Privileg (Least Privileges)
 - > erlaubt die Ausführung von System-Operationen
 - > kann in Profiles Programmen zugeordnet werden
 - > Abprüfung durch den Kernel
- Autorisierung (authorization)
 - > erlaubt die Ausführung von privilegierten Aktionen in Programmen
 - > Abprüfung durch Anwendung selbst

Role Based Access Control (RBAC)



Solaris 10 System Administration Collection
 System Administration Guide: Security Services
RBAC in the Solaris Operating Environment
<http://www.sun.com/software/whitepapers/wp-rbac/wp-rbac.pdf>

RBAC Begriffe (2)

- User Profile
 - > /etc/user_attr bündelt alle zugeteilten Profile und Autorisierungen für Benutzer und legt Rollen fest
 - > Einstellungen werden in die User-Session übernommen
- Rolle
 - > account mit UID, Home, Passwort
 - > login in role nur durch bestimmte Nutzer durch su(1M)
 - > wird für eine spezielle Funktion vergeben, z.B. DB-admin
 - > z.B. root als Rolle einführen, damit ist nur noch root-login per su möglich

RBAC - Funktionsweise

/etc/user_attr

dettlefd:::auths=solaris.smf.manage.ssh;type=normal;profiles=Zone Management

/etc/security/prof_attr

...
 Zone Management::
 Zones Virtual Application Environment Administration:
 help=RtZoneMngmnt.html
 ...

/etc/security/auths_attr

...
 solaris.smf.manage.ssh::
 Manage Secure Shell Service States::
 help=SmfSshStates.html
 ...

... /etc/security/exec_attr

Zone Management:solaris:cmd:::/usr/sbin/zlogin:uid=0
 Zone Management:solaris:cmd:::/usr/sbin/zoneadm:uid=0
 Zone Management:solaris:cmd:::/usr/sbin/zonecfg:uid=0

RBAC oder sudo ?

- Warum sudo anstelle von RBAC ?
 - > Cross platform
 - > CLI Argumente kontrollierbar
 - > per Kommando Environments möglich
- Warum RBAC anstelle von sudo ?
 - > seit S8 in Solaris supported
 - > dokumentiert, integriert (u.a. SMF und Least Privileges) und maintained
 - > unterstützt durch SMC und Webmin GUI's
 - > wird mit der Common Criteria Evaluation evaluiert
 - > realisiert Rollen und profiled Shell ähnlich sudo
 - > wird von audit mit ausgewertet

Rolle oder Profile ?

- Rolle
 - > shared Account ist möglich
 - > zweites login ist gewünscht
 - > für bestimmte Funktionen
- Profile
 - > isolierte Festlegung von Aktionen mit suid oder Privilegien
 - > Profile shells
 - > /usr/bin/pfsh, /usr/bin/pfcsh, /usr/bin/pfksh
 - > prüft bei Ausführung Profiles

User Rights Management (Roles)

Solaris Users versus Roles

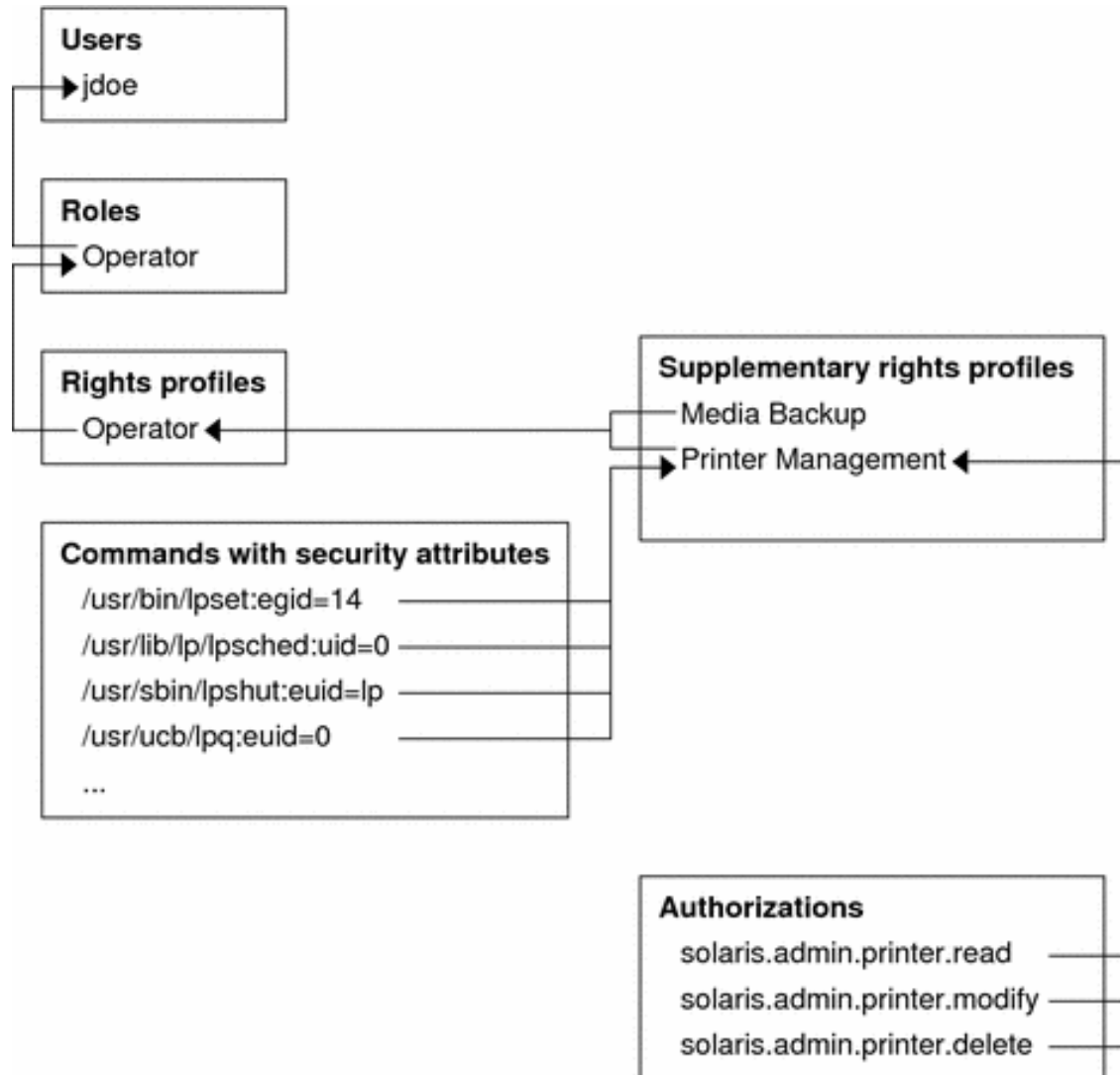
- > Roles can only be accessed by users already logged in.
- > Users cannot assume a role unless authorized.

```
$ 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
```

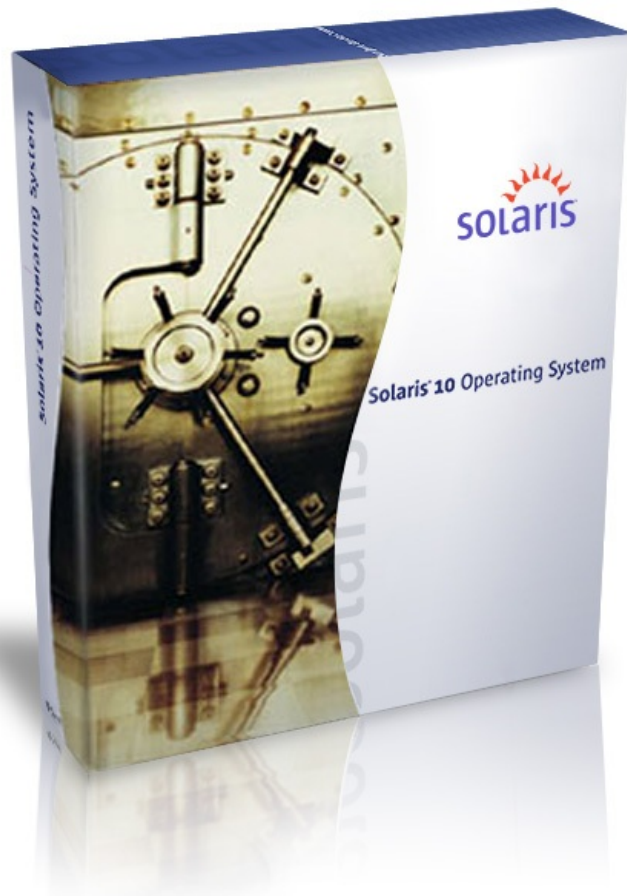
```
$ svcadm restart apache2
```

```
$ 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
```

Solaris Trusted Extensions with Solaris 10 11/06



Labeled Security for Solaris 10

Multi-Level Desktop, Networking
and Printing

Labeled Filesystems and Devices

Compatible with all Solaris
hardware and applications

Common Criteria Target:

CAPP, RBACPP, LSPP @ EAL 4+

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 (objects) and clearances (subjects)
 - > Additional desktop policies
- Solaris Containers (Zones)
 - > Unique Sensitivity Labels
 - > Trusted (label-based) Networking

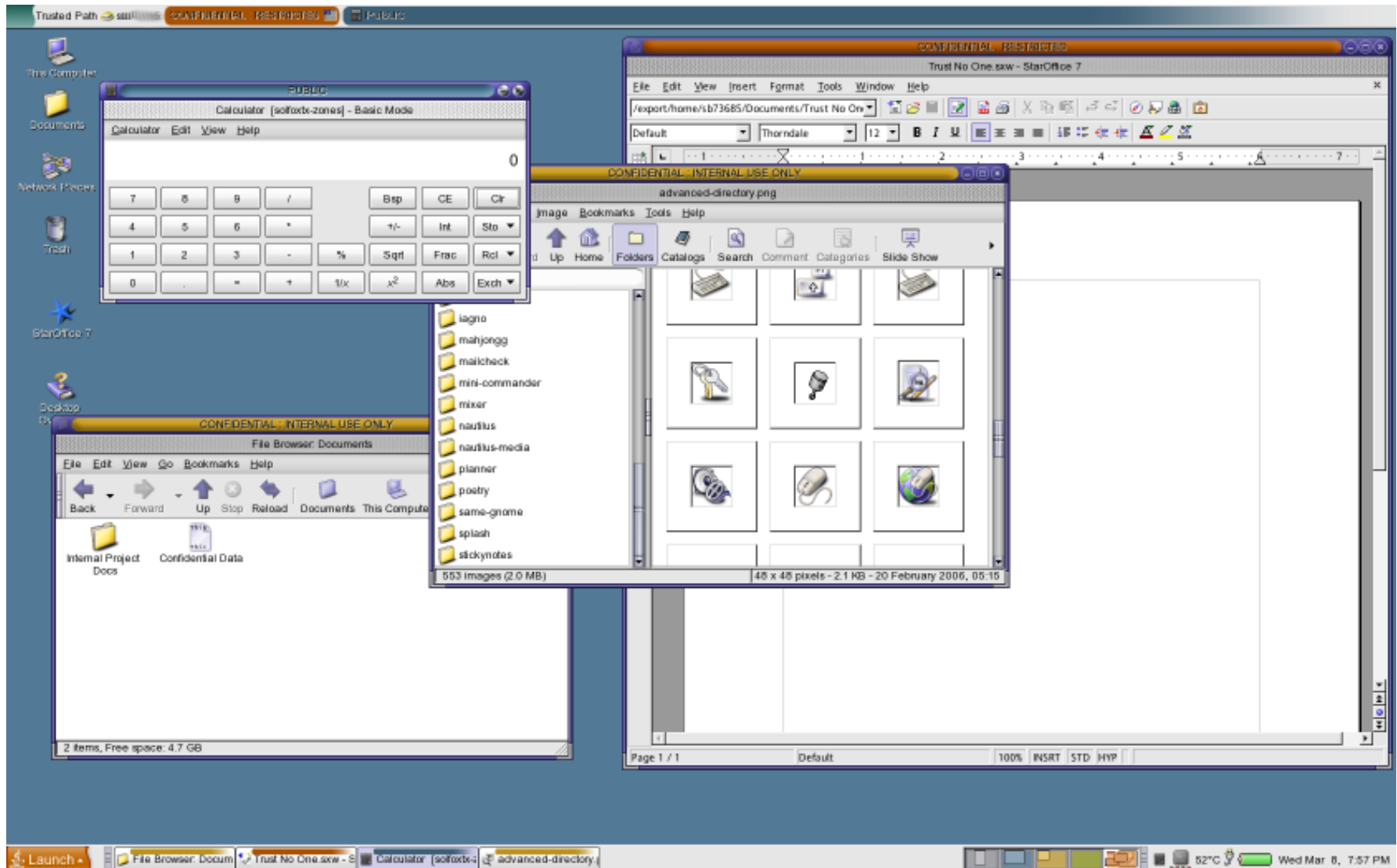
Trusted Extensions in a Nutshell

- 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.

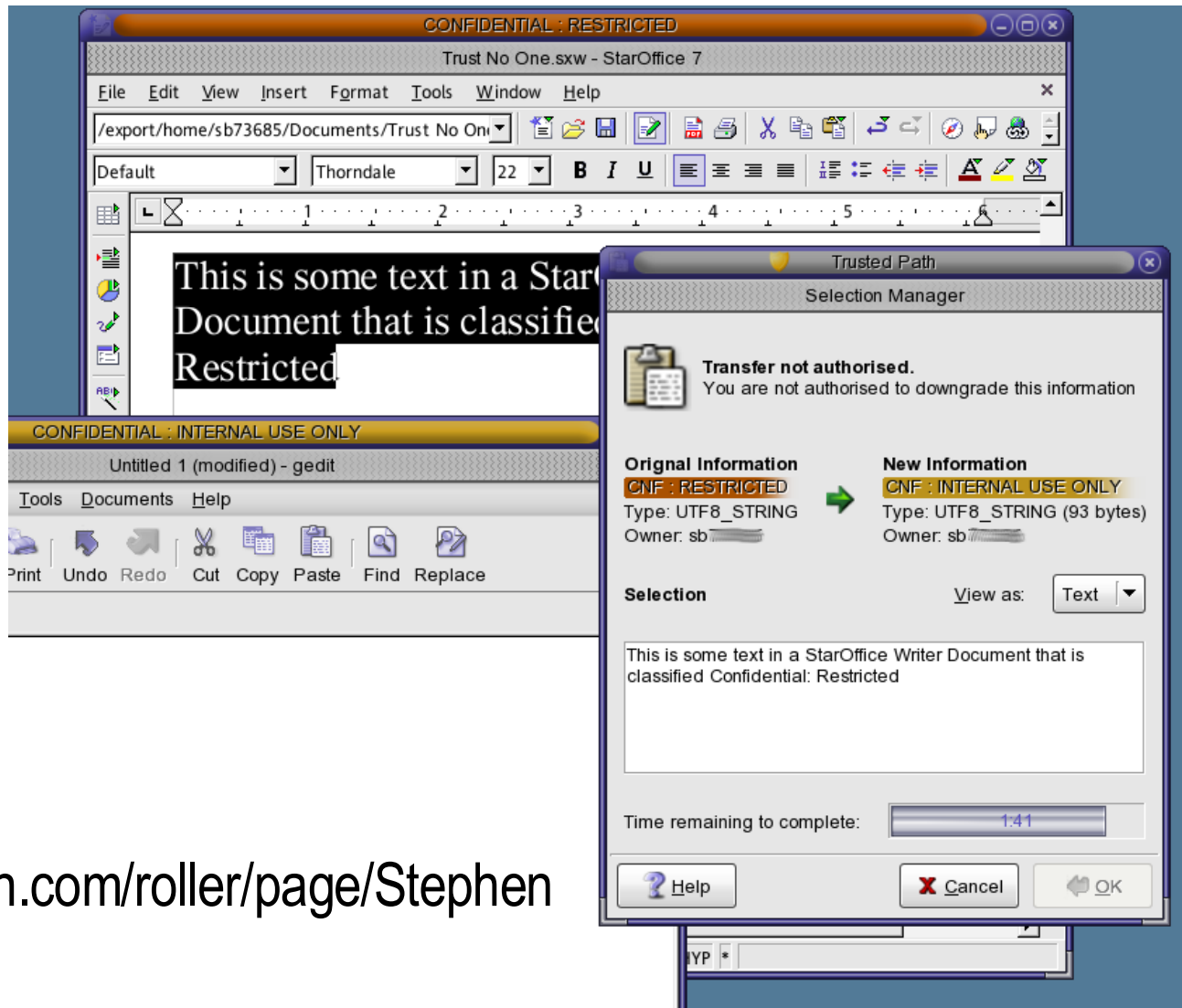
What are Label-Aware Services?

- 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



blogs.sun.com/roller/page/Stephen

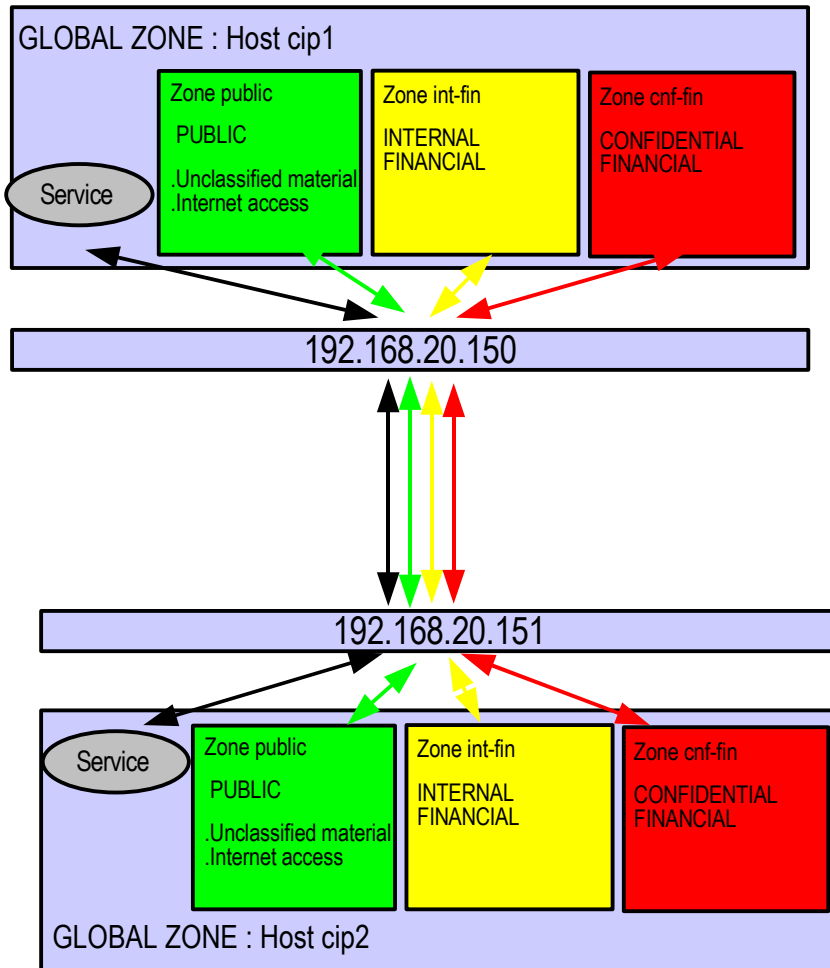
Key TX functionalities : Trusted Networking

- On a TX host, networking is labelled
 - > CIPSO is used for inter-TX hosts communications.
 - > Commercial IP Security Option
 - > The label is inserted in the option header of the packets.
 - > Label is checked for establishing connection.
- Any unlabeled packet is either ignored or matched from source.
 - > It is possible to integrate with non label aware resources.
- Non cipso aware “utility” resources (routers, DNS, ...) have to be declared `admin_low`.

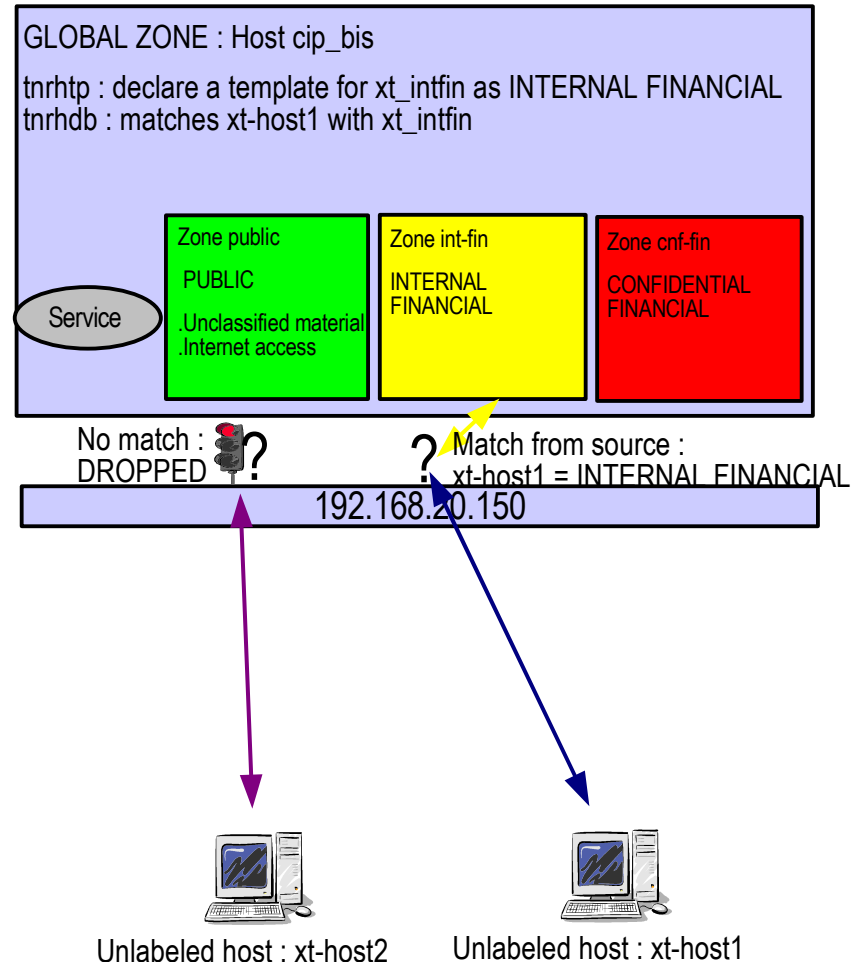
Trusted Networking : Simple cases...

Basic example : CIPSO and unlabeled communication

CIPSO communication



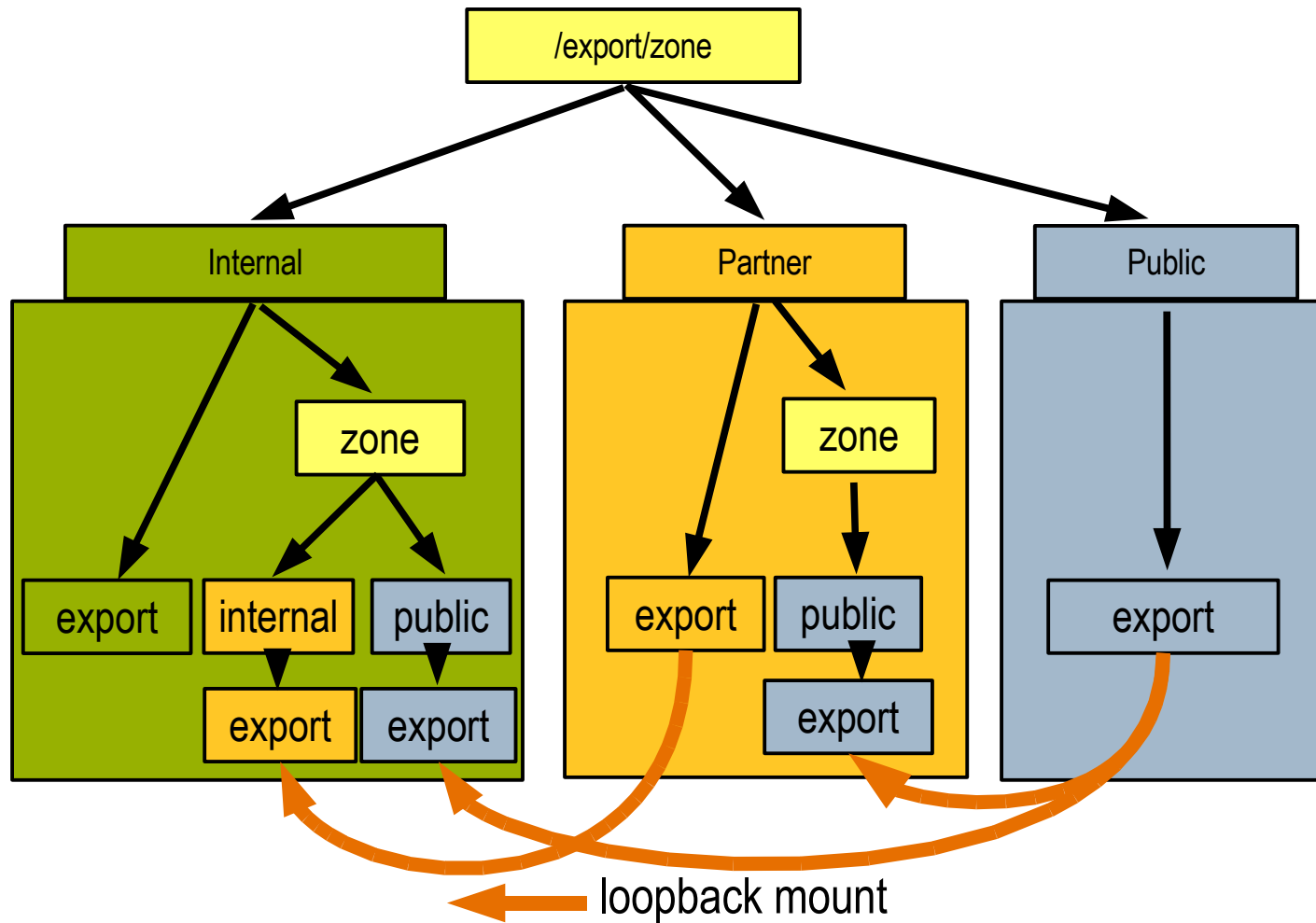
Unlabeled communication



Zonen für Trusted Extensions

- Jede Zone hat einen Label
 - > Labels sind durch Zone IDs der Prozesse realisiert
 - > Prozesse sind durch Label (und Zone ID) isoliert
 - > Dateien einer Zone nehmen den Zonen Label an
- Besonderheit der globalen Zone
 - > Administration aller Zonen
 - > keine Labeling Policies in der globalen Zone
 - > keine Nutzerprozesse, nur Trusted Computing Base (TCB)
 - > Trusted Path Attribute wird gesetzt
 - > stellt Services für andere Zonen zur Verfügung
- Zentraler Name Service für alle Zonen
- Gerätezuordnung pro Zone bzw. pro Label

Lower-Labeled Files Lesen



Why Labeled Security?

Common IS and DAC

“an hard limit is up to be reached...”

- On standard systems, access to an object is checked vs. the subject identity basis only.
 - > This is DAC : Discretionary Access Control.
 - > Based on the UID/GID on the subject side.
 - > Based on the “owners” and linked rights (i.e. rwx) on object side.
- Different sensitivity level objects could be owned at same ID
 - > this sensitivity is not handled by the DAC systems.
 - > A subject can handle freely any object he have the right to access.

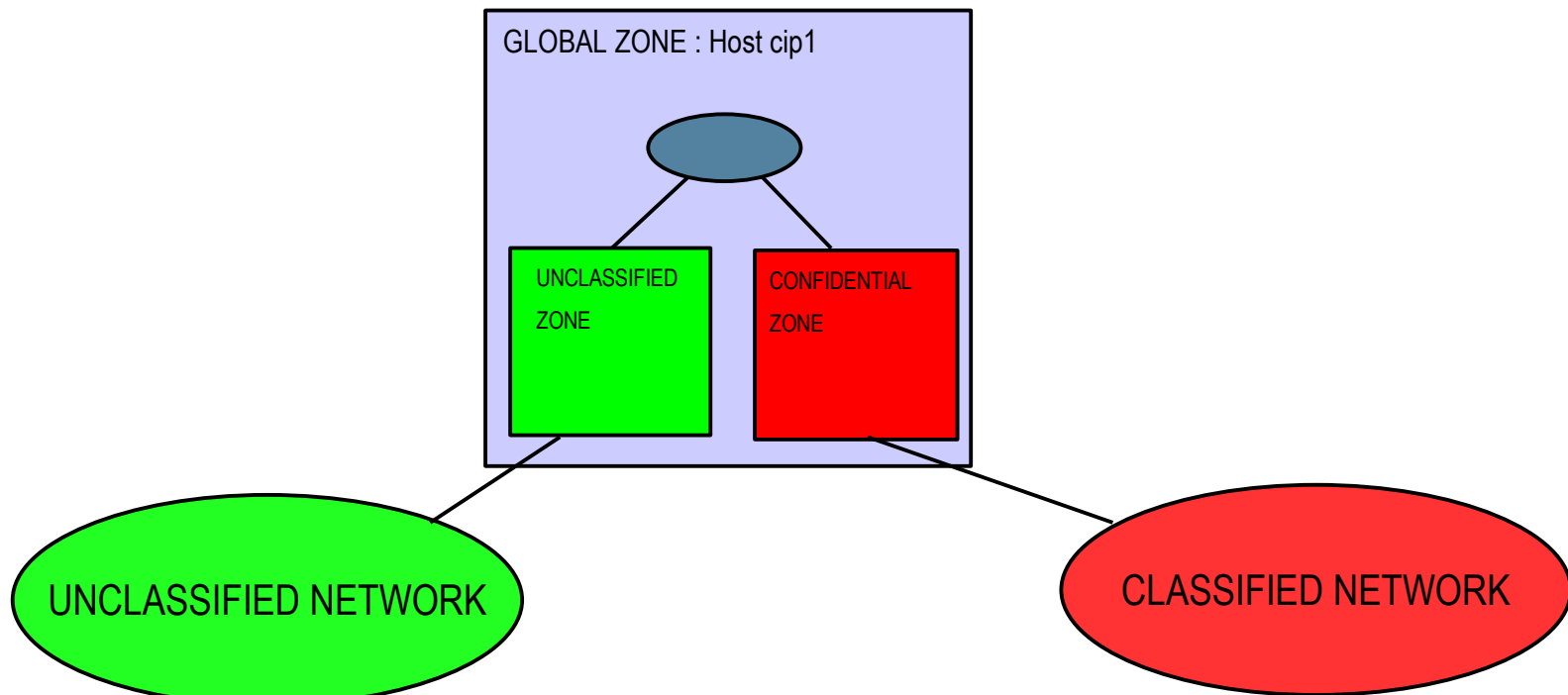
When the cure can be worst than the disease...

- Trying to solve that issue within conventional systems often leads to physical dissociation and compartmentation, or “leaking patches”.
- This is not acceptable in most cases because of :
 - > Costs vs. need for competitiveness.
 - > Static model vs. need for IS dynamicity to stick to business evolution constraints.
 - > Closed model vs. opening of production tools.
 - > Heavy human policies & procedures vs. users productivity.
 - > Infrastructure complexity leading to :
 - > Maintenance complexity and costs.
 - > Door fully opened to human mistake or policies “turnaround”.



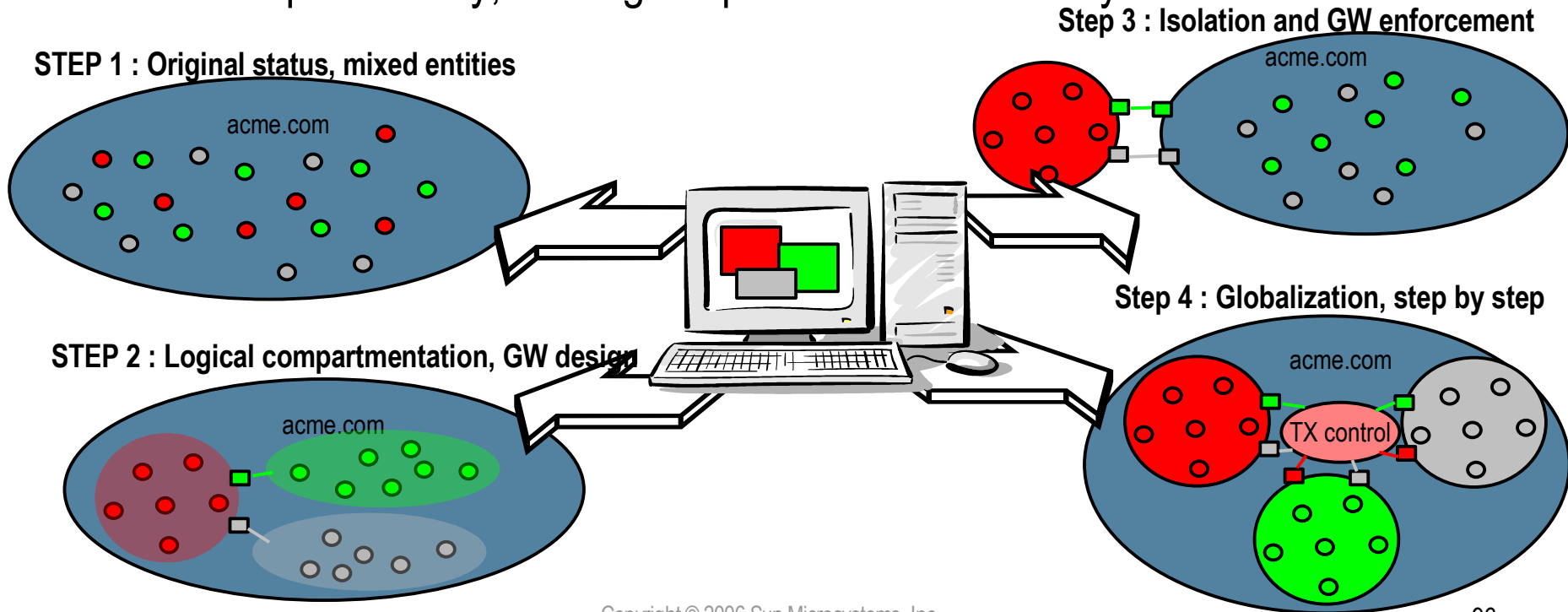
Trusted gateways...

- TX Gateways focuses on
 - > Standard services (proxies, MTAs, etc...)



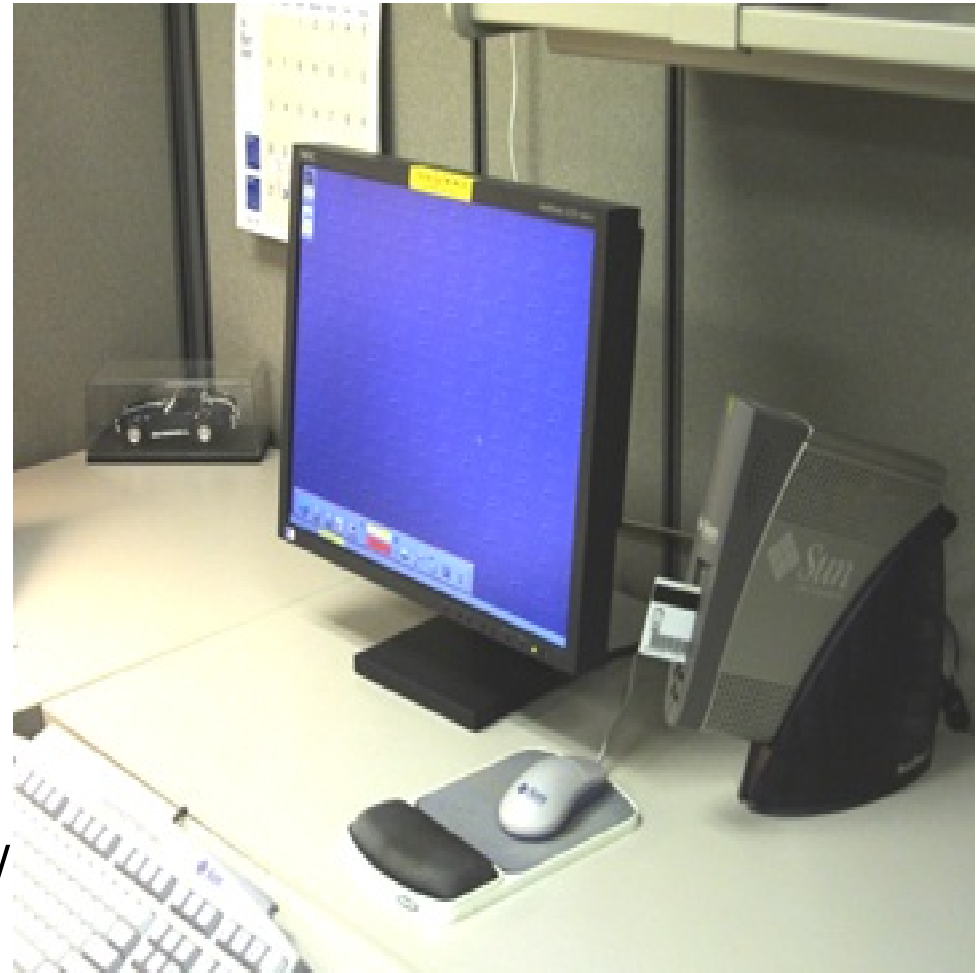
Group compartmentation...

- Compartmentation of “sensible groups” within the enterprise
 - > Potentially address any company owning financial, R&D or HR departments they consider sensible. So... any company...
 - > Can be done on a step by step phasing, with low intrusion to IS and users productivity, and high improvement on security...



GSNAP, SunRays and SSGD...

- SunRays are of HUGE added value
 - > User friendly terminal.
 - > No state.
 - > Low maintenance cost.
- *SNAP
 - > SunRay TX integration
- SSGD
 - > “userland” is mostly windows.
 - > Do they care about it ?
 - > They care about the tools.
 - > SSGD deliver the tools, as they are used to them : productivity.



Attack Detection Scenario

- Solaris Audit
- Basic Audit and Reporting Tool (BART)
- Cryptographically Signed ELF Objects
- Solaris Fingerprint Database
- Solaris Security Toolkit

Solaris Audit

- Kernel auditing of system calls and administrative actions.
 - > Can record events happening in any zone (from the global zone).
- 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,1834,3097759606,12
114 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>

Solaris Auditing

- Overview
- Audit configuration
- Parsing the audit trail

Overview of Solaris auditing

- An old Solaris (SunOS) feature
 - > But unfortunately not very well known
- Comes from the C2 requirement for accountability
- “Solaris” auditing is also available on
 - > FreeBSD
 - > TrustedBSD
 - > Darwin (Mac OS X)
- Most of this presentation applies to the above
- More OSes are adopting OpenBSM
 - > There is a Linux port coming...

Overview of Solaris auditing

- Every system call can generate an audit record
 - > And many administrative commands too, like
 - > `poweroff`
 - > `useradd`
 - > `passwd`
- Audit records are generated in the kernel
 - > And put on a ring buffer
- The user land daemon `auditd` grabs the records from the ring buffer
 - > And writes them to the audit trail

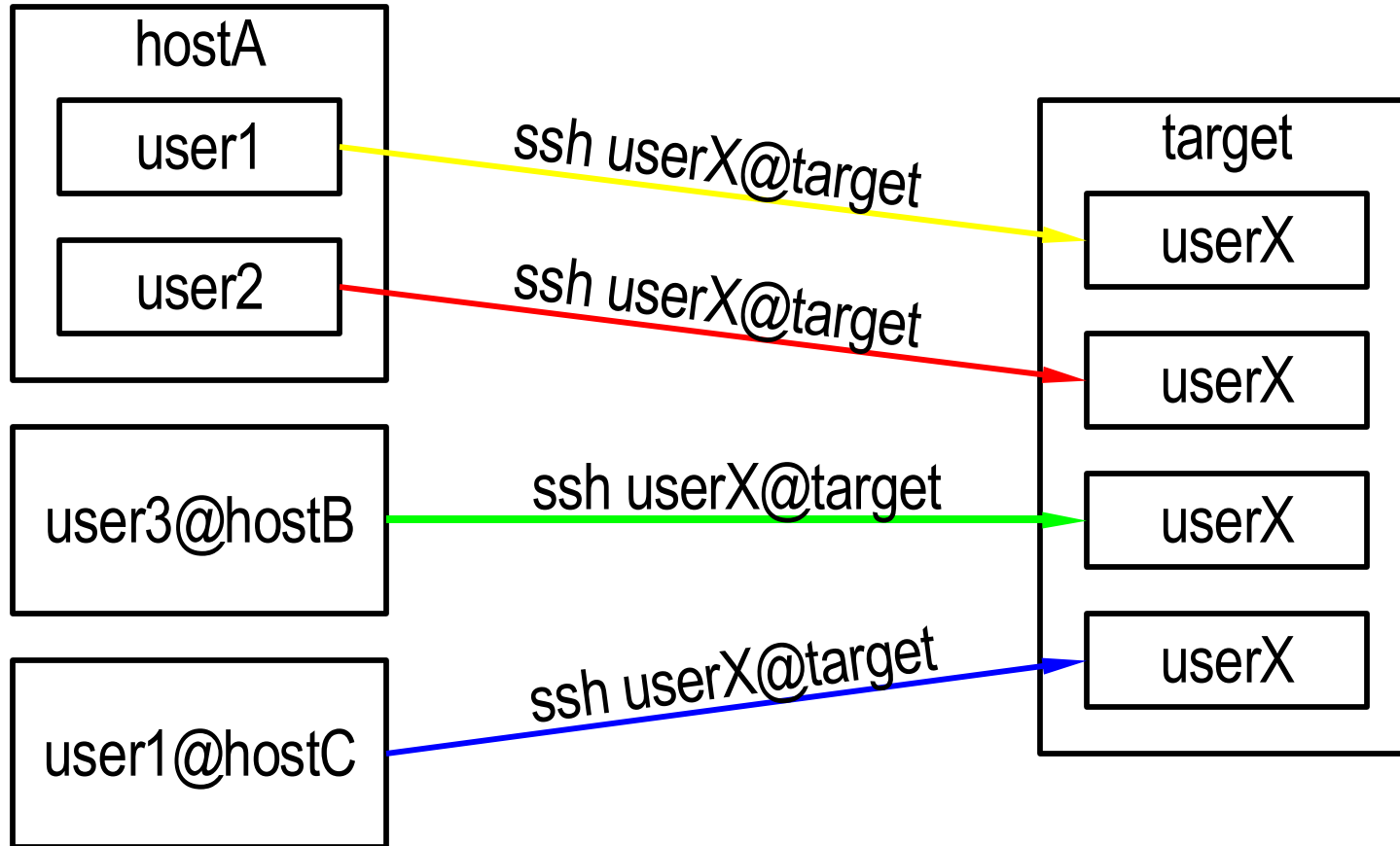
Overview of Solaris auditing

- Privileges
 - > `proc_audit`
 - > Allows a process to generate audit records
 - > `sys_audit`
 - > Allows a process to manipulate the audit system
 - > For more information see `privileges(5)`
- Profiles
 - > Audit Review
 - > Lets you review the audit trail, which by default is only accessible by root
 - > Audit Control
 - > Lets you perform all audit administrative tasks

Audit terminology

- Audit trail
 - > A collection of one or more audit files that store the audit data from all systems that run the auditing service.
- Audit record
 - > Audit data that is stored in audit files. An audit record describes a single audit event. Each audit record is composed of audit tokens.
- Audit token
 - > A field of an audit record or event. Each audit token describes an attribute of an audit event, such as a user, a program, or other object.
 - > See `audit.log(4)`

How to keep things apart?



And now everybody runs: `su - userY -c "ls /foo"`

Audit terminology

- Audit id
 - > Id tied to the “initial” user, stays the same even after a “su -” i.e. an `exec(2)` of a setuid program
- Session id
 - > Id tied to the login session. Unique session id for each login

Solaris Auditing

- Overview
- Audit configuration
- Parsing the audit trail

Audit configuration

- Enabling auditing
 - > `bsmconv, bsmunconv`
 - > Note that the automounter gets disabled when you enable auditing, but can be re-enabled with:
`svcadm enable autofs`
- Configuration files
 - > `audit_control, audit_user,`
`audit_class, audit_event,`
`audit_startup`
- Audit commands
 - > `audit, auditconfig`

The audit_event file

- Maps audit events to audit classes

> Format: number:name:desc:class(es)

```

1:AUE_EXIT:exit(2):ps
2:AUE_FORK:fork(2):ps
4:AUE_CREAT:creat(2):fc
5:AUE_LINK:link(2):fc
6:AUE_UNLINK:unlink(2):fd
7:AUE_EXEC:exec(2):ps,ex
8:AUE_CHDIR:chdir(2):pm
9:AUE_MKNOD:mknod(2):fc
10:AUE_CHMOD:chmod(2):fm
11:AUE_CHOWN:chown(2):fm
6172:AUE_ssh:login - ssh:lo
6169:AUE_poweroff_solaris:poweroff(1m):ss
6227:AUE_zlogin:login - zlogin:lo
  
```

The audit_class file

- Defines the audit classes
 - > Format: mask:name:description

```

0x00000000:no:invalid class
0x00000001:fr:file read
0x00000002:fw:file write
0x00000004:fa:file attribute access
0x00000008:fm:file attribute modify
0x00000010:fc:file create
0x00000020:fd:file delete

...

0x10000000:ct:Custom trace events
0x20000000:io:ioctl
0x40000000:ex:exec
0x80000000:ot:other
0xffffffff:all:all classes (meta-class)

```

Using audit flags

- Audit flags (for a class abbreviated “cl”)
 - > All cl events
 - > cl
 - > All failed cl events
 - > -cl
 - > All successful cl events
 - > +cl
 - > All events, except cl
 - > all, ^cl
 - > All events, except successful cl events
 - > all, ^+cl

The `audit_control` file

- Defines overall audit settings
 - > Location and max disk usage of audit files
 - > `dir:/var/audit`
 - > `minfree:20`
 - > System wide audit flags for users
 - > `flags:lo,ex`
 - > Non attributable audit flags
 - > `naflags:na`

The `audit_control` file

- To send `lo` class events to syslog, add
 - > `plugin:name=audit_syslog.so;`
`p_flags=lo`
 - > And it will generate messages with facility `audit` and level `notice`
 - > Looking like this:

```
Oct 11 09:39:38 ferrari audit: [ID 702911  
audit.notice] su ok session 3402238775 by  
martin as root:root in global from ferrari
```

The `audit_user` file

- Lets you override system wide settings for an individual user
 - > Format `user:always:never`
 - > E.g. `martin:all,^+fr:-nt`
 - > Means for the user martin always audit all events except successful file reads, and never audit failed network events

The `audit` command

- Three modes of operation
 - > Log switching, i.e. start a new audit file
 - > `audit -n`
 - > Normally added to a cron job to start a new log weekly, daily or hourly – depending on amount of data generated
 - > Signal `auditd` to re-read the configuration files
 - > `audit -s`
 - > Terminate `auditd`
 - > `audit -t`

The `auditconfig` command

- The “swiss knife” audit command
- Used to check if audit is enabled
 - > `auditconfig -getcond`
- Checking the audit info for a process
 - > `auditconfig -getpinfo pid`
- Get/set the audit policy
 - > `auditconfig -getpolicy`
 - > `auditconfig -setpolicy +policy`
 - > **See** `auditconfig(1M)` for policy options

The audit_startup file

- Setting the starting audit policy
 - > Used to make policy changes effective after reboot

```
#!/bin/sh
/usr/bin/echo "Starting BSM services."
/usr/sbin/auditconfig -setpolicy +cnt
/usr/sbin/auditconfig -setpolicy +argv
/usr/sbin/auditconfig -setpolicy +zonename
/usr/sbin/auditconfig -setpolicy +seq
/usr/sbin/auditconfig -conf
/usr/sbin/auditconfig -aconf
```

Solaris Auditing

- Overview
- Audit configuration
- Parsing the audit trail

Parsing the audit trail

- Audit commands
 - > `auditreduce, praudit`
- Audit trails
 - > Default location `/var/audit`
 - > File name format:
 - > `yyyymmddhhmss.yyyymmddhhmss.host`
 - > `yyyymmddhhmss.not_terminated.host`
 - > Default file permissions are
 - > `user == root`
 - > `group == root`
 - > `mode == 0640`

Audit command: `auditreduce`

- Used to select a subset of records from logfiles(s)
 - > `-a` after date (format: `yyyymmddhhmmss`)
 - > `-b` before date
 - > `-c` class
 - > `-u` audit user
 - > `-e` effective user
 - > `-r` real user
 - > `-s` session
 - > `-m` event
 - > `-z` zone
 - > `-o path=/path/to/file`

Audit command: `praudit`

- Used to convert audit records into human readable format
 - > `-l` produce one audit record per line (for plain text)
 - > Useful when piping the output to a (perl) script
 - > `-r` produce raw output
 - > i.e. no uid to user name, IP to hostname lookups
 - > `-x` produce xml output

Example

- Listing logins/logouts (the `lo` class)

```
# auditreduce -c lo *.not_terminated.ferrari | praudit
```

```
header,100,2,login - local,,localhost,2006-10-09 21:07:14.010 +02:00
subject,martin,martin,martin,martin,martin,495,3090409780,0 0 localhost
return,success,0
zone,global
sequence,2
```

```
header,100,2,su logout,,localhost,2006-10-09 21:10:06.564 +02:00
subject,martin,audit,audit,audit,audit,756,3090409780,0 0 localhost
return,success,0
zone,global
sequence,912
```

```
header,84,2,login - ssh,,localhost,2006-10-09 21:10:22.577 +02:00
subject,audit,audit,audit,audit,audit,769,1262240463,1386 136704 localhost
return,success,0
zone,global
sequence,925
```

...

Example

- Listing all events belonging to a session

```
# auditreduce -s 3777453003 *.not_terminated.ferrari | praudit

header,100,2,login - local,,localhost,2006-10-11 15:31:52.260 +02:00
subject,martin,martin,martin,martin,martin,496,3777453003,0 0 localhost
return,success,0
zone,global
sequence,2

header,218,2,execve(2),,,localhost,2006-10-11 15:31:52.370 +02:00
path,/usr/dt/config/Xsession.jds
attribute,100755,root,bin,0,27031,0
exec_args,2,/bin/ksh,/usr/dt/config/Xsession.jds
subject,martin,martin,martin,martin,martin,497,3777453003,0 0 localhost
return,success,0
zone,global
sequence,3

header,204,2,execve(2),,,localhost,2006-10-11 15:31:52.392 +02:00
path,/usr/dt/bin/Xsession
attribute,100555,root,bin,0,8770,0
exec_args,2,/bin/ksh,/usr/dt/bin/Xsession
subject,martin,martin,martin,martin,martin,497,3777453003,0 0 localhost
return,success,0
zone,global
sequence,4
```

...

Example

- Listing outgoing network traffic

```
# auditreduce -m 32 *.not_terminated.ferrari | praudit
```

```
header,142,2,connect(2),,localhost,2006-10-11 16:45:21.272 +02:00
argument,1,0x4,so
socket,0x0200,0x0200,0xf488,192.168.2.34,0x0016,192.168.2.36
subject,martin,martin,martin,martin,martin,1118,3777453003,0 0 localhost
return,success,0
zone,global
sequence,1299
```

- Decoding the socket token:
 - > From:
 - > IP: 192.168.2.34 Port: 32600
 - > To:
 - > IP: 192.168.2.36 Port: 22

Example

- Listing outgoing network traffic using XML
 - > `audit_event` lets you lookup the system call `connect(2)` to the event id 32

```
# auditreduce -m 32 *.not_terminated.ferrari | praudit -x
```

```
<record version="2" event="connect(2)" host="localhost" iso8601="2006-10-11
19:57:31.921 +02:00">
<argument arg-num="1" value="0x4" desc="so"/>
<socket sock domain="0x0200" sock type="0x0200" lport="0xcd10"
laddr="192.168.2.34" fport="0x0016" faddr="192.168.2.36"/>
<subject audit-uid="martin" uid="martin" gid="martin" ruid="martin"
rgid="martin" pid="1236" sid="2239025147" tid="0 0 localhost"/>
<return errval="success" retval="0"/>
<zone name="global"/>
<sequence seq-num="990"/>
</record>
```

Basic Auditing and Reporting Tool

- File-level integrity validation tool.
 - > Operates in either “create” or “compare” mode.
 - > “rules” files define what should be evaluated and how.
 - > “manifest” files contain the results.
- Flexible operational methods.
 - > Allows “BART” input and output to be stored locally, piped to another process (transmission, compression, encryption, signing, etc.)
 - > Processing can easily be automated using SSH and RBAC¹.
- Very small footprint (1 binary)

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

BART Examples

- BART rules (bart_rules(4))

```
/usr/sbin
CHECK all
```

- BART manifest (bart_manifest(4))

```
/usr/sbin/acctadm F 28356 100555 user::r-x,group::r-x,mask:r-x,other:r-x 414f3bb4
0 2 ece9d92d00b0c13ed2d56580e3856df7
```

- BART Create Operation:

```
# bart create -r rules > manifest
# find /usr/lib/nis | bart create -l > manifest
```

- BART Compare Operation:

```
# bart compare ./manifestA ./manifestB
/usr/sbin/auditd:
  acl control:user::r-x,group::r-x,mask:r-x,other:r-x
    test:user::r-x,group::r-x,mask:r-x,other:rwX
  contents control:28dd3a3af2fcc103f422993de5b162f3
    test:28893a3af2fcc103f422993de5b162f3
```

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.

Solaris Fingerprint Database

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

Sunsolve -> Support Resources -> Security Resources

```
# digest -v -a md5 /usr/lib/ssh/sshd  
md5 (/usr/lib/ssh/sshd) =  
b94b091a2d33dd4d6481df fa784ba632
```

[Process fingerprint using the Solaris Fingerprint DB]

```
b94b091a2d33dd4d6481df fa784ba632 - (/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 Security Toolkit

Configurable (and pluggable) security tool used to configure or assess the security posture of a Solaris system.

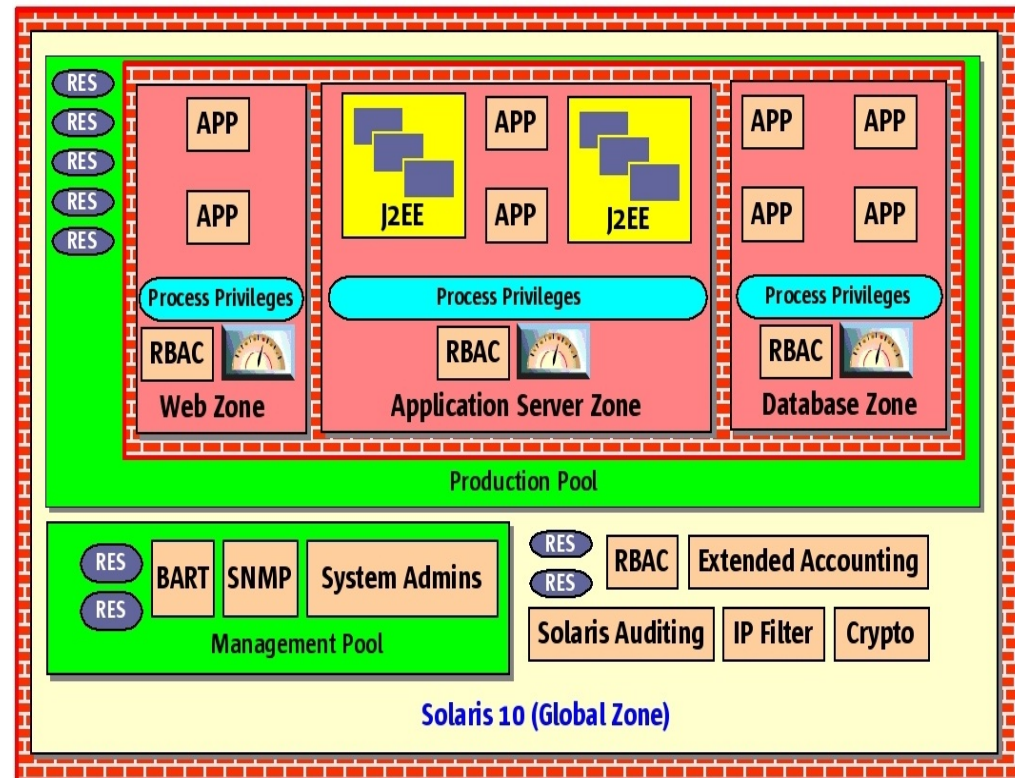
```
# jass-execute -a hardening.driver -V 2
[...]
disable-spc [FAIL]
    Service svc:/application/print/cleanup:default
    was enabled.
disable-spc [FAIL]
    Service svc:/application/print/cleanup:default
    was running.
disable-spc [FAIL] Script Total: 2 Errors

disable-ssh-root-login [PASS]
    Service svc:/network/ssh:default was installed.
disable-ssh-root-login [PASS]
    PermitRootLogin parameter is set to "no" in
    /etc/ssh/sshd_config.
disable-ssh-root-login [PASS] Script Total: 0 Errors
```

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



For More Information

- Sun Security Home
 - > <http://www.sun.com/security>
- SunSolve Security Resources
 - > <http://sunsolve.sun.de>
- 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>



Practical Solaris 10 Security

Franz Haberhauer

Sun Microsystems GmbH

Franz.Haberhauer@Sun.com

blogs.sun.com/FranzHaberhauer

