RBAC in Solaris 10

Darren J Moffat

Staff Engineer, Networking & Security

Sun Microsystems, Inc.

7th October 2004

Agenda

- Least Privilege / RBAC in Solaris 10
- SMF - Service Management Framework
- Zones (N1 Grid Containers)
- Solaris Cryptographic Framework
- Other security releated features in Solaris 10

Traditional Method

- All powerful root user
- BSD/SunOS use of wheel group
  - Must be in wheel group
  - Must know the password
- Wrapper scripts & setuid

What is Role Based Admin ?

- Application of Principle of Least Privilege
- Roles ~ Job Function
  - Printer Admin / User Admin / Database Admin
- Give only the commands needed
- Give only the privileges needed

Least Privilege in Solaris 10

- Traditional UNIX is root or user
  - Kernel checks explicitly for uid = 0 or object owner
- CMW and later (expired) POSIX specifications on least privilege.
- Solaris 10 privileges evolution of 10+ years of experience in Trusted Solaris.

Solaris Privileges

- 50+ fine grained privileges instead of uid == 0
- Each process has 4 privilege sets in its' kernel creds:
- Inheritable set (I)
  - The set of privileges child processes get on exec.
- Permitted set (P)
  - The maximum set of privileges for the process
- Effective set (E)
  - Subset of P that are currently asserted as needed by the process
- Limit set (L)
  - Upper bound a process and its children can obtain (takes effect on exec)

Viewing process privileges

NFS daemon

```
# ppriv `pgrep nfsd`
357:    /usr/lib/nfs/nfsd
flags = PRIV_AWARE
        E:
basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_
session,sys_nfs
        I:
basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_
session
        P:
basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_
session,sys_nfs
        L:
basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_
session

# pcred `pgrep nfsd`
357:    e/r/suid=1  e/r/sgid=12
```

Viewing process privileges

Normal user shell

```
$ ppriv $$
2337:   ksh
flags = <none>
        E: basic
        I: basic
        P: basic
        L: all
```

What privileges do I need ?

Privilege “Debug” mode allows you to determine this:

```
$ ppriv -D $$
$ cat /etc/shadow
cat[3003]: missing privilege "file_dac_read" (euid =
35661, syscall = 225) needed at ufs_iaccess+0xd2
cat: cannot open /etc/shadow

$ cp /usr/sbin/ping /tmp
$ /tmp/ping jurassic
ping[3016]: missing privilege "net_icmpaccess" (euid = 35661,
syscall = 230) for "devpolicy" needed at so_socket+0xa7
/tmp/ping: socket Permission denied
```

Basic Privileges

- New for Solaris 10 are basic privileges.
  - Not in previous Trusted Solaris implementations.
- These are things all normal users can normally do.
  - `proc_fork, proc_exec, proc_session, proc_info, file_link_any`
- Dropping `proc_fork` and `proc_exec` from system daemons that should never fork or exec gives extra protection against buffer overflow exploits that attempt to get a shell

What is a Role in Solaris ?

- User account with “normal” attributes
- Can't be logged into directly – only su or assumed in smc
- Normally has a set of Rights Profiles
- Normally has a profile shell as `$SHELL`
  - `/bin/pfsh, /bin/pfcsh, /bin/pfksh`
  - All these are links to normal shell but use `/bin/pfexec` to run with privilege if needed.

Solaris RBAC configuration

- `exec_attr`: Execution profiles specify commands and the user, group ids and default/limit privileges
- `prof_attr`: Rights Profiles are collections of execution profiles and authorizations
- `auth_attr`: Authorizations Definition
- `user_attr`: Profiles, Authorizations, Roles (grant & define), Projects
- All tables are multi-field with extensible key-value pairs: C APIs provided.

RBAC & privileges

- RBAC profiles list the privileges the process will inherit when run.
- Examples:

```
•   Process Management:solaris:cmd:::/usr/bin/nice:privs=proc_owner,proc_priocntl
•   Process Management:solaris:cmd:::/usr/bin/kill:privs=proc_owner
•   File System Management:solaris:cmd:::/usr/sbin/umount:privs=sys_mount
•   Network Management:solaris:cmd:::/usr/sbin/ifconfig:privs=sys_net_config
```

How is RBAC used ?

- Rights profiles allow for a hierarchical definition
- Authorizations checked by privileged programs:
  - SMC – Administration Interface and internal use
  - SMF – Service Management Facility
  - Device Commands: allocate, cdrw
- Projects for “accounting” and resource management/billing.
- Admin via SMC and/or `usermod/rolemod`

SMF – Service Management Framework

- SMF – Service Management Framework
  - Dependancy based system service startup
- SMF service definitions (manifests) security attributes:
  - Assign uid/gid/default and limit privileges to services
  - Provide a Solaris RBAC authorization that is required to administer the service.
    - $ svcadm restart svc://network/lp
    - That restarts the lp service as a normal user if the user had the authorization.
- Provides distinction between configured/enabled
  - Service can be fully configured but disabled

Zones

- Multiple virtualized application environments from a single Solaris kernel
- Process containment
  - Resource usage & security isolation
- No direct access to hardware
- Zones appear as separate hosts from “outside” the Solaris instance
  - Zones have unique set of 0 or more IP addresses.

Zones

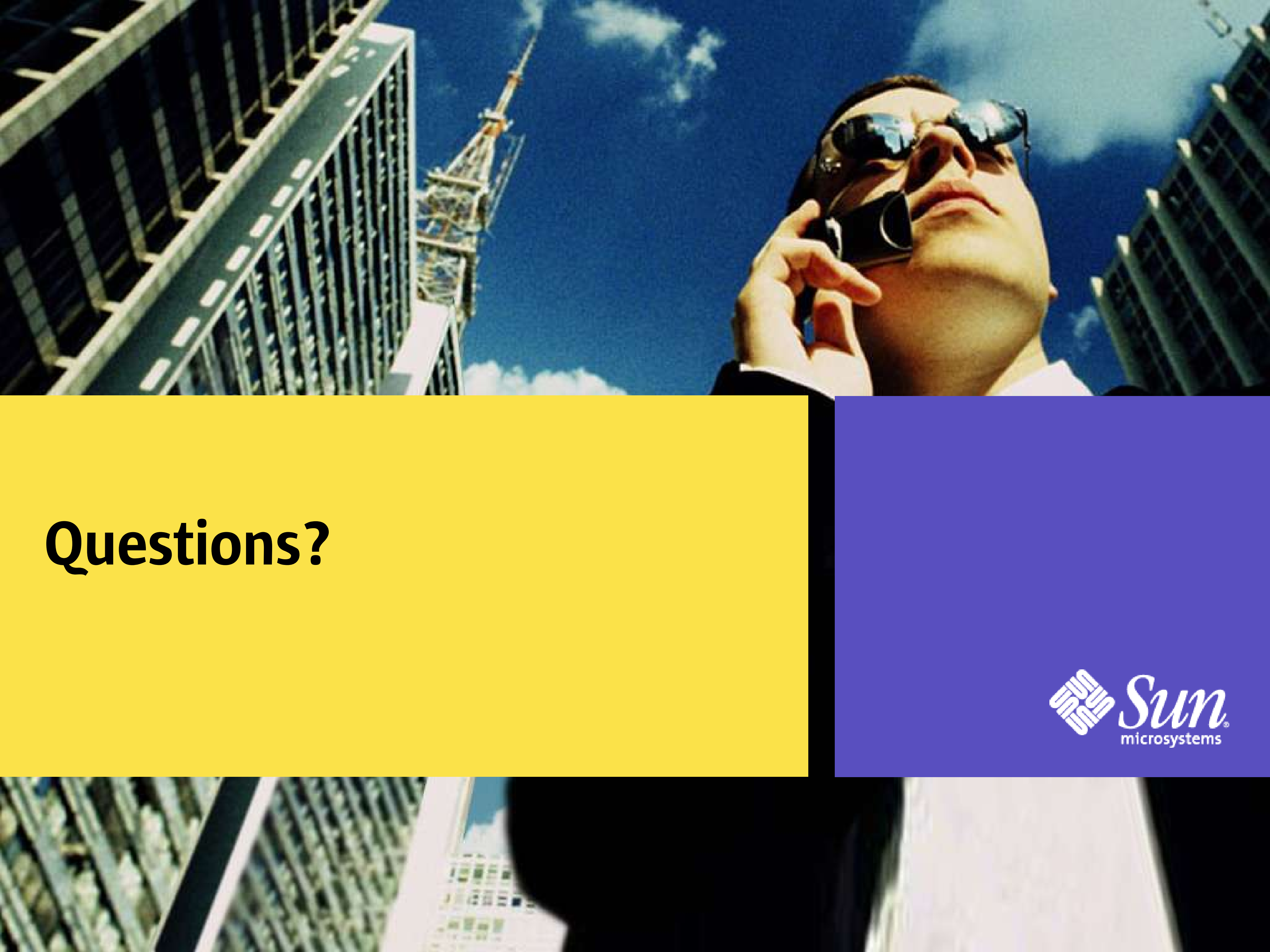
- Each Zone in Solaris 10 has a subset of the available privileges.
  - Zones don't have any of the system management privileges and are missing some of the privileges for Dtrace.
  - In addition to this processes in Zones can't send signals to other zones even if they do have proc_session or proc_owner
- Can only see processes in same Zone (except global zone)
- Separate uid/gid namespace
- Separate filesystem space

Solaris Cryptographic Framework

- User and kernel cryprographic framework.
- Userland is PKCS#11
- OpenSSL to PKCS#11 ENGINE
- Kernel support used by IPsec, Kerberos (NFS)
- Userland used by Kerberos, IKE, OpenSSL ENGINE apps
- Java 1.5 uses Solaris PKCS#11 out of the box.
- Seemless access to hardware crypto
- Kernel load balances between hardware/software
- Pluggable kernel & user interfaces.
- cryptoadm(1m) command for policy

Password enhancements

- N failed login attempts can now lock account
  - Accounts can be marked as no lock
- Password history
- Improved control over password sanity checks
  - Including cracklib support
- Support for pluggable crypt(3c) interface [Solaris 9]
  - Supports Linux/BSD MD5 & Blowfish



Questions?

Solaris Security

Darren J Moffat

darren.moffat@sun.com

sudo vs Solaris RBAC

Feature	Solaris RBAC	Sudo
Authorisations	Y	N
PAM	Y	Y
Cross Platform	N	Y
Kerberos Support	Y[6]	Y
Solaris BSM Audit	Y	N
RUID	Y	Y[9]
EUID	Y	N[9]
RGID	Y	N
EGID	Y	N
Hierarchical Profiles	Y	N[11]
Network Wide Policy	Y [1]	N [2]
Host Specific Policy	Y [3]	Y [4]
Netgroup Policy	N	Y
Require Password	N[12]	Y
Allow no Password	Y [5]	Y
Cached Authentication	N [6]	Y
Restrict Users	Y	N
Profile Shells	Y	N
Control cmd arguments	N	Y
Privileges/Capabilities Aware	Y[10]	N
Authenticate as Self	N[7]	Y
Control Sensitive Environment Variables	Y[8]	Y
Control UMASK	N	Y
Fine grained Policy Admin	Y	N
Default Profiles for OS Admin	Y	N

Notes

1 All supported Nameservices

2 Assumes “rdist”

3 Follows nsswitch: files can override remote nameservice

4 Host/network/netgroup policy in config

5 Not for NIS+ roles

6 When configured for su(1) in pam.conf(4)

7 No for Roles but Yes for just profiles

8 When used as a role su(1) rules apply

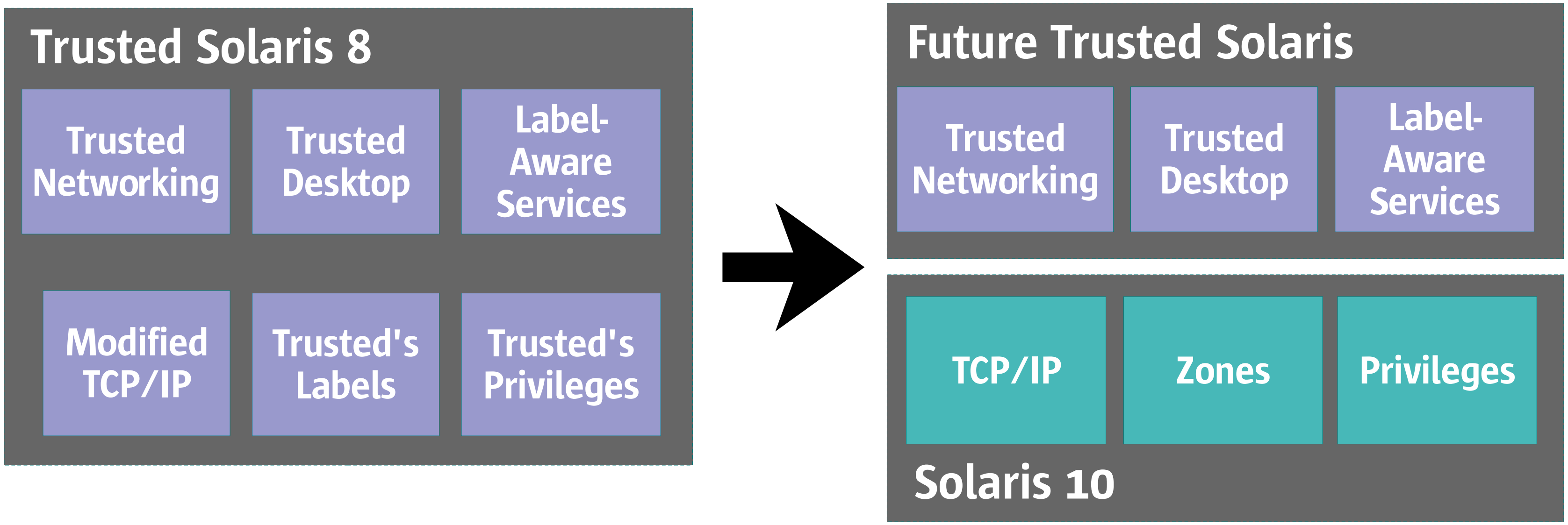
9 stay_setuid provides similar functionality

10 Only used in Trusted Solaris

11 Profiles are approximately the same as sudo Cmd_Alias

12 Roles may require a passord[5] profile shells don't

Layered Trusted Solaris™



Benefits:

- Software portability
- Patch compatibility
- Shorter release window
- More familiar