

Security-Features in Solaris 10

best Systeme GmbH, Unterföhring

Wolfgang Stief

stief@best.de

Dipl.-Ing. (FH)
Senior Systemingenieur Unix

2005-04-20
Open Systems Day '05

Agenda

Solaris 10 Security Architecture

User Rights Management

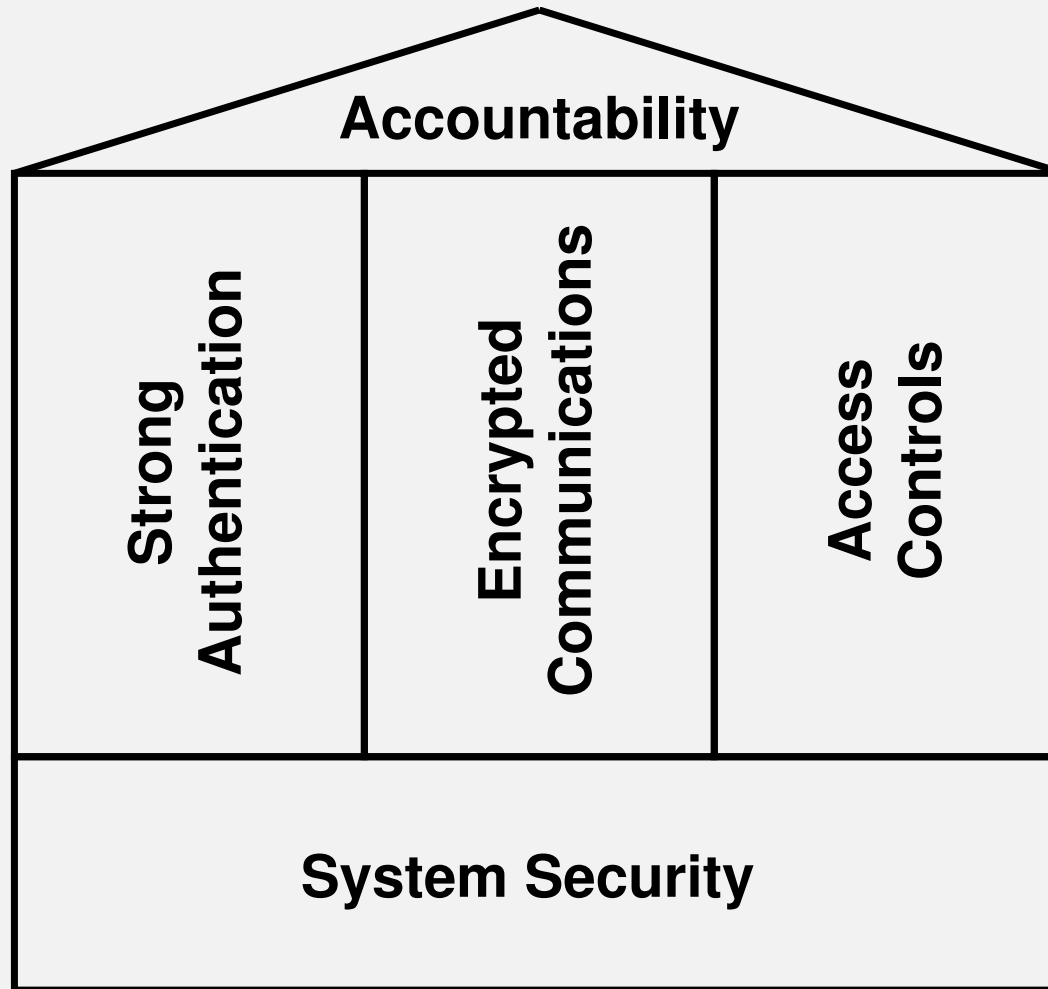
Process Rights Management

Solaris Container (aka Zones)

Layered Trusted Solaris

some more features

Solaris 10 Security Architecture



Solaris 10 Security Architecture

System Security

- sichere Netzinstallation
 - Vermeidung von Buffer Overflows
 - einfach deaktivierbare Netzwerkdienste
 - Solaris Security Toolkit
 - Security Best Practices (*Sun Blueprints*)
 - automatisiertes Patch Management
 - *Common Criteria Evaluation*
- ◇ Solaris Containers
 - ◇ *Strong Security Posture* zur Installationszeit
 - ◇ Vereinfachte Netzwerkinstallation
 - ◇ Integritätscheck für Dateien (BART)
 - ◇ Service Manager
 - ◇ verschlüsseltes Dateisystem

Solaris 10 Security Architecture

Strong Authentication

- sicheres Single Sign On (*Kerberos v5*)
 - ◇ LDAP über Kerberos
 - ◇ Secure Shell über Kerberos
 - ◇ Password Aging und Password History
- Solaris Secure Shell
- LDAP Authentifizierung über SSL
- Pluggable Authentication Modules – PAM
- Authentifizierung über Smartcard
- Java VM für Smartcards
- Gateways von NIS und NIS+ nach LDAP
- starke Passwortverschlüsselung

Solaris 10 Security Architecture

Encrypted Communications

- Solaris Secure Shell
 - ◊ Solaris Cryptographic Framework
 - ◊ OpenSSL
 - ◊ Kerberos mit DES, 3DES und AES
- Single Sign On über Kerberos v5, verschlüsselt
- verschlüsseltes NFS über Kerberos
- integrierter Support für Crypto-Beschleuniger
- IPsec/IKE
- Default-Verschlüsselung mit 128bit, 256bit AES, 448bit Blowfish

Solaris 10 Security Architecture

Access Controls

- Security Labels (nur bei *Trusted Solaris*)
- Zugriffsrechte auf Dateien, POSIX Access Control Lists
- User Rights Management (Role Based Access, RBAC)
- TCP Wrapper
- IP Paketfilter (SunScreen)
- ◇ erweitertes User Rights Management
- ◇ Process Rights Management
- ◇ IP Paketfilter (IP Filter, ipf)
- ◇ Solaris Container
- ◇ PAM auch zum Sperren von Accounts und für Passwort-Checks

Solaris 10 Security Architecture

Accountability

- erweitertes, skalierbares kernelbasiertes Auditing (System)
 - applikationsbasiertes Auditing mit XML-Output (über Syslog)
 - rollenbasierte Logeinträge für Accounts
 - Solaris Fingerprint Datenbank
 - Common Criteria EAL 4+ (ISO/IEC 15408) (Trusted Solaris)
- ◇ Audit Records je Solaris Container
 - ◇ signierte Patches und automatisierbare Patch-Prozesse
 - ◇ User Rights Management
 - ◇ Process Rights Management

Solaris 10 Security Architecture

User Rights Management

Process Rights Management

Solaris Container (aka Zones)

Layered Trusted Solaris

some more features

User Rights Management

- Alternative zum herkömmlichen Superuser-Modell
- Sog. *least privilege model*: jeder User bekommt – über Rollen – genau die Rechte, die er für seine Aufgaben braucht
- vergleichbar mit sudo
- per default sind **keine** Rollen definiert
- kommt ursprünglich von *Trusted Solaris*
- Konfiguration über `roles(1)`, `roleadd(1m)`, `smrole(1m)`, `usermod(1m)` oder per Maus im *Solaris Management Center*

User Rights Management (cont'd)

Role Based Access besteht aus:

- Authorization
(`/etc/security/auth_attr`)

- Privileges

- Security Attributes

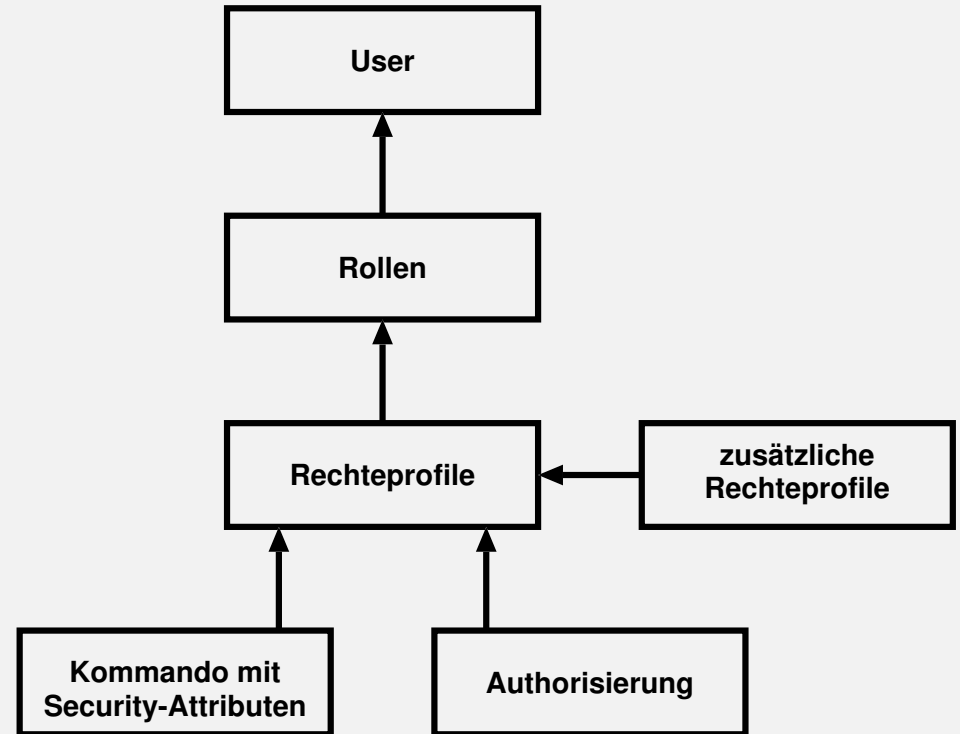
- Privileged Application

- Rechteprofil (*rights profile*)

Sammlung an Fähigkeiten, die einem User oder einer Rolle zugewiesen werden können.

- Rolle (*role*)

Besondere Identität, um eine privilegierte Applikation auszuführen.



Solaris 10 Security Architecture

User Rights Management

Process Rights Management

Solaris Container (aka Zones)

Layered Trusted Solaris

some more features

Process Rights Management

- Einschränkung von Prozessen auf
 - ⇒ Kommandos
 - ⇒ User
 - ⇒ Rollen
 - ⇒ System Level
- kommt von *Trusted Solaris*
- setzt Rechte auf Prozesse, die diese brauchen, um erfolgreich zu laufen
- Rechte sind im Kernel verankert
- Konfiguration über `ppriv(1)`, `usermod(1)`, `privileges(5)`
- mehr als 40 Privilegien verfügbar, unterteilt in 5 Gruppen: FILE, IPC, NET, PROC, SYS
- Privilegien können mit `exec(2)` vererbt werden

Process Rights Management – kurzes Beispiel

```
lugh<root> [/]# ppriv $$
13842:  bash
flags = <none>
      E: all
      I: basic
      P: all
      L: all
```

E – Effective privilege set: Diese Privilegien sind derzeit effektiv wirksam.

I – Inheritable privilege set: Zeigt an, ob und in welchem Ausmass Privilegien durch exec(2) vererbt werden können. Nach einem exec(2)-Aufruf sind sind typischerweise P und E gleich.

P – Permitted privilege set: Diese Privilegien sind insgesamt verfügbar.

L – Limit privilege set: Begrenzt die vererbaren Privilegien von I. Zum Zeitpunkt eines exec(2)-Aufrufs werden damit P und E limitiert.

Process Rights Management – längliches Beispiel

```
lugh<root> [/]# ppriv -v $$
13842:  bash
flags = <none>
E:  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
I:  file_link_any,proc_exec,proc_fork,proc_info,proc_session
P:  cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,file_chown,
    [...]
L:  cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,file_chown,
    [...]
```

Solaris 10 Security Architecture

User Rights Management

Process Rights Management

Solaris Container (aka Zones)

Layered Trusted Solaris

some more features

Solaris Container

Solaris Container sind Verknüpfungen von Solaris Zonen mit dem Solaris Resource Manager auf einer Maschine. Wesentliche Eigenschaften von sog. *Non-global Zones*:

Sicherheit: Weder der in einer Zone laufende Prozess noch davon abstammende Subprozesse können die Zone wechseln. Auch Netzwerkdienste können für sich abgeschottet in einer Zone laufen.

Isolation: Applikationen in unterschiedlichen Zonen auf gleicher Hardware sind voneinander isoliert und können so z. B. jede für sich auf scheinbar globale Ressourcen oder Konfigurationen zugreifen, ohne sich gegenseitig zu stören (z. B. Kernelparameter in `/etc/system` getrennt je Zone).

Virtualisierung: Zonen bieten eine virtualisierte Umgebung in der physikalische Details der darunter liegenden Hardware versteckt bzw. ausgeblendet werden können. Jede Zone kann einen eigenen Administrator haben, lokale Änderungen in einer Zone haben keine Auswirkung auf benachbarte Zonen.

Granularität: Der Grad der Isolation und Virtualisierung ist granular einstellbar.

Umgebung: Durch eine Zone wird das Operating Environment **nicht** verändert. Die Applikation sieht nach wie vor ein Standard Solaris Betriebssystem, allerdings mit leichten Einschränkungen.

Solaris Container – Pro und Contra

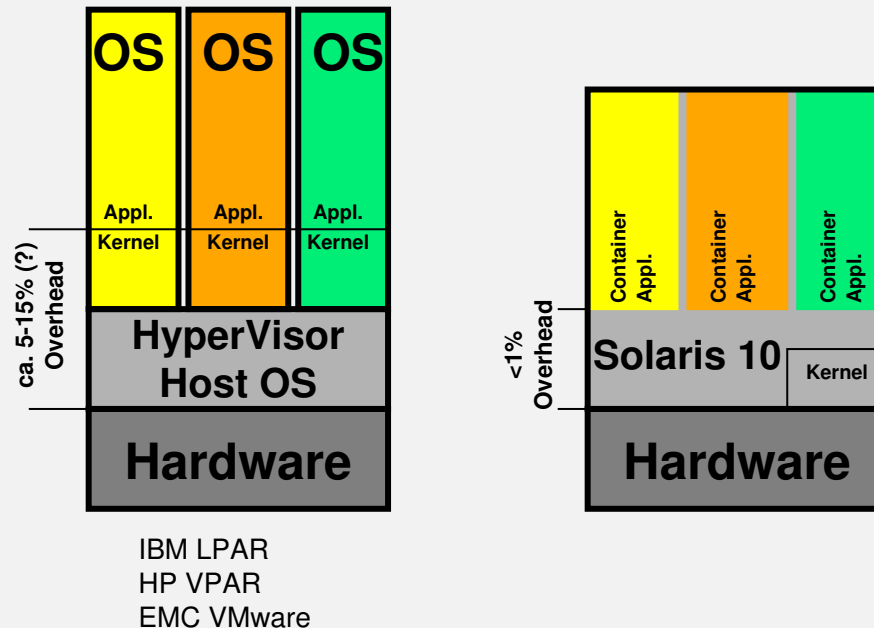
+ Nur eine OS-Instanz \Rightarrow wenig Overhead

Im Laboraufbau: 4000 Zonen auf V880 (8CPU) und 431.427 Zonen auf Fire E25k (144CPU)

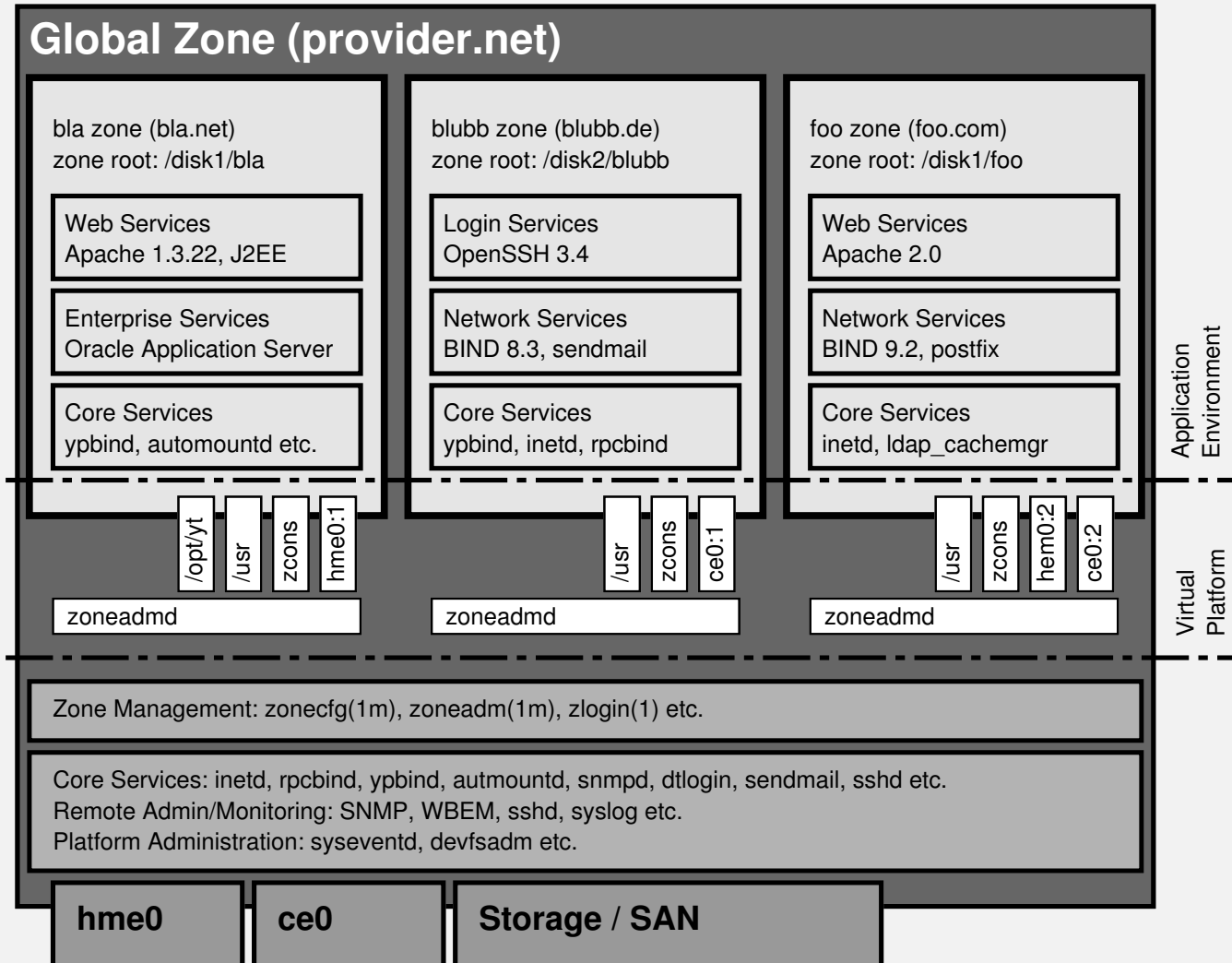
<http://blogs.sun.com/roller/page/jcclingan>

+ *zentrales* Patchmanagement

– keine unterschiedlichen Patchstände und OS-Releases auf einer Plattform möglich



Solaris Zones – Beispiel



Solaris Zones – Hands On

```
lugh<root> [/]# mkdir -p /disk1/bla
lugh<root> [/]# zonecfg -z bla
bla: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:bla> create
zonecfg:bla> set zonepath=/disk1/bla
zonecfg:bla> add net
zonecfg:bla:net> set physical=hme0
zonecfg:bla:net> set address=172.16.42.23/24
zonecfg:bla:net> end
zonecfg:bla> verify
zonecfg:bla> commit
zonecfg:bla> ^D
lugh<root> [/]# zonecfg -z bla info zonepath
zonepath: /disk1/bla
lugh<root> [/]# zonecfg -z bla info net
net:
    address: 172.16.42.23/24
    physical: hme0
```

Solaris Zones – Hands On (cont'd)

```
lugh<root> [/]# zoneadm -z bla install
Preparing to install zone <bla>.
Creating list of files to copy from the global zone.
Copying <3248> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <919> packages on the zone.
Initialized <919> packages on zone.
Zone <bla> is initialized.
lugh<root> [/]# zoneadm list -cv
  ID NAME           STATUS           PATH
  0 global           running          /
  - bla             installed        /disk1/bla
lugh<root> [/]# zoneadm -z bla boot
lugh<root> [/]# zoneadm list -cv
  ID NAME           STATUS           PATH
  0 global           running          /
  1 bla             running          /disk1/bla
```

Solaris Zones – Hands On (cont'd)

```
lugh<root> [/]# zlogin -C bla
[Connected to zone 'bla' console]
# bash
bash-2.05b# zonename
bla
bash-2.05b# ps -ef | wc -l
    31
bash-2.05b# ~.
lugh<root> [/]# zonename
global
lugh<root> [/]# ps -ef | wc -l
    73
lugh<root> [/]# zlogin -l root bla
[Connected to zone 'bla' pts/2]
# bash
bash-2.05b# ifconfig -a
lo0:1: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
hme0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 172.16.42.23 netmask ffffffff broadcast 172.16.42.255
```

Solaris Zones – Hands On (cont'd)

```
bash-2.05b# zonename
```

```
bla
```

```
bash-2.05b# cat /etc/release
```

```
Solaris 10 s10_63 SPARC
```

```
Copyright 2004 Sun Microsystems, Inc. All Rights Reserved.
```

```
Use is subject to license terms.
```

```
Assembled 14 July 2004
```

```
bash-2.05b# ~.
```

```
lugh<root> [/]# zonename
```

```
global
```

```
lugh<root> [/]# cat /etc/release
```

```
Solaris 10 s10_63 SPARC
```

```
Copyright 2004 Sun Microsystems, Inc. All Rights Reserved.
```

```
Use is subject to license terms.
```

```
Assembled 14 July 2004
```

Solaris 10 Security Architecture

User Rights Management

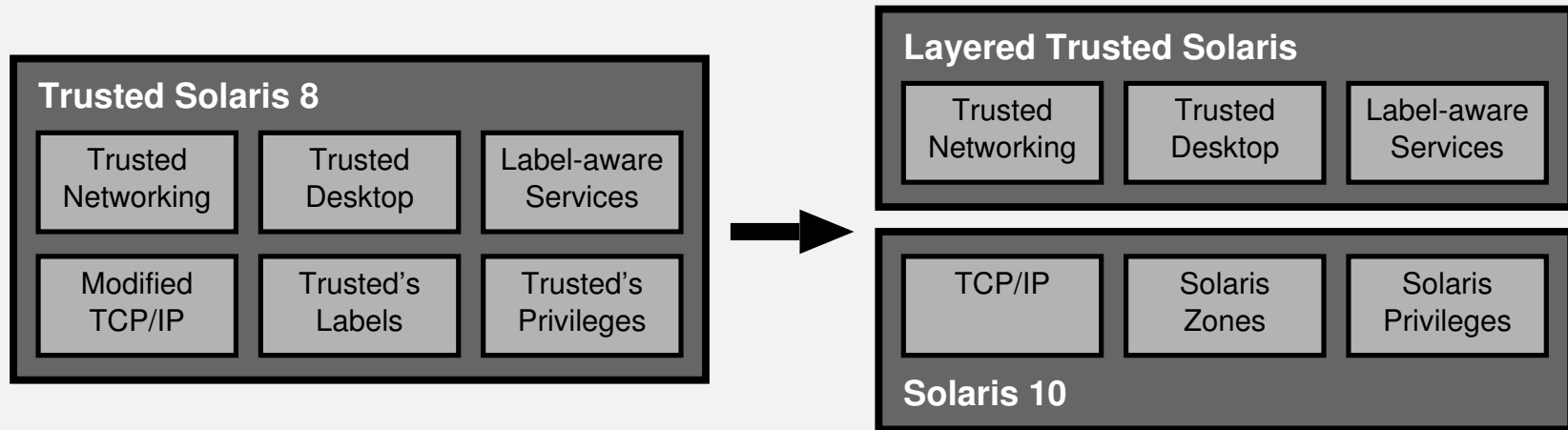
Process Rights Management

Solaris Container (aka Zones)

Layered Trusted Solaris

some more features

Layered Trusted Solaris



- installierbares Package als OS-Aufsatz
- drei *Common Criteria Protection Profile* Zertifizierungen auf EAL 4+ (ISO/IEC 15408)
- derzeit nur *Trusted Solaris 8* erhältlich; eigenständiges Produkt, noch nicht *layered*

Solaris 10 Security Architecture

User Rights Management

Process Rights Management

Solaris Container (aka Zones)

Layered Trusted Solaris

some more features

some more features

Cryptographic Framework – Algorithmen und Libraries gemäß *RSA Security Inc. PKCS #11 Cryptographic Token Interface (Cryptoki)*.

Pluggable Authentication Modules – PAM. Modulare Umgebung zur Authentisierung.

Auditing and Reporting – Zonenübergreifendes Erfassen von Logindaten, Veränderung von Objekten, Benutzung von Privilegien und Rollen, administrative Tätigkeiten etc.

IP Firewall – Stateful Inspection und NAT; Open Source, abgeleitet von *ipf* aus dem Umfeld *FreeBSD* und *NetBSD*.

Stack Overflow Protection – System-Parameter, um *executable stacks* zu unterbinden; Logging von Stack Overflows

Quellen

- Ulrich Gräf
Solaris 10 Neuigkeiten
Vortrag auf Sun Partner-SE University
Februar 2005
- Solaris 10 System Administration Guide: Security Services
Sun Microsystems, Inc.
676 Seiten, Januar 2005
- Solaris 10 System Administration Guide: Solaris Containers – Ressource Management and Solaris Zones
Sun Microsystems, Inc.
334 Seiten, Januar 2005
- Solaris 10 System Administration Guide: IP Services
Sun Microsystems, Inc.
820 Seiten, Januar 2005
<http://docs.sun.com/>
- Solaris 10 Security FAQ
<http://www.sun.com/software/solaris/faqs/security.xml>

Danke für Ihre Aufmerksamkeit.

Fragen?