

Friday, February 29. 2008

### **Zahlenspiel zur Wahl in Hamburg**

Nils zitiert in "Impressionen eines Wahlhelfers" den Verein Mehr Demokratie e.V.: Beim Volksentscheid vom Oktober stimmten 365.133 Hamburgerinnen und Hamburger für verbindliche Volksentscheide. Bei der jetzigen Bürgerschaftswahl stimmten 331.184 Hamburgerinnen und Hamburger für die CDU und Ole v. Beust. Der Volksentscheid scheiterte, Ole v. Beust aber bleibt Bürgermeister.

Posted by Joerg Moellenkamp at 09:13

### **New tape library from Sun: Sun StorageTek SL3000**

We´ve announced a new tape library: The SL3000. Up to 3000 tapes. Up to 56 drives. It´s a smaller system for people who doesn´t need the massive sized SL8500 (we had a gap here before, the SL8500 was too large and the SL1400 too small for some customers)

Posted by Joerg Moellenkamp in Sun at 08:38

Thursday, February 28. 2008

## Solaris Features: Service Management Facility

Okay, i finalized my SMF tutorial. Was a good distraction from other events ...

Part 1: Introduction  
Part 2: The foundations of SMF  
Part 3: Working with SMF  
Part 4: Developing for SMF  
Part 5: Conclusion

Posted by Joerg Moellenkamp in Solaris at 13:15

## Solaris Features: Service Management Facility - Part 5: Conclusion

Okay, I hope i was able to give you some insights into the Service Management Framework. It's a mighty tool and this article only scratched on the surface of the topic. But there are several excellent resources out there.

Documentation  
Solaris 10 System Administrator Collection - Basic Administration - Managing Services  
man page - smf(5)

FAQ  
opensolaris.org: SMF(5) FAQ

Other resources  
Bigadmin - Solaris Service Management Facility - Quickstart Guide  
Bigadmin - Solaris Service Management Facility - Service Developer Introduction  
SMF shortcuts on wikis.sun.com  
cuddletech.com: An SMF Manifest Cheatsheet

Posted by Joerg Moellenkamp in Solaris at 12:51

## Solaris Features: Service Management Facility - Part 4: Developing for SMF

Okay, now you know how to do basic tasks. But how to use SMF for your your own applications. I will use openvpn as an example. A good source for this program is blastwave. Please install the package tun and openvpn

I want to show a running example, thus we have to do some work. It will be just a simple static shared key configuration, as this is a SMF tutorial, not one for OpenVPN. We will use theoden and gandalf again. gandalf will be the server.  
theoden the client.10.211.55.201 gandalf  
10.211.55.200 theoden

Preparing the server  
Okay ... this is just a short configuration for an working OpenVPN server.

```
# mkdir /etc/openvpn
# cd /etc/openvpn
# openvpn --genkey --secret static.key
# openvpn --dev tun --ifconfig 192.16.1.1 172.16.1.2 --secret static.key --daemon
# scp static.key jmoekamp@10.211.55.200:/tmp/static.key
Now just leave this terminal this way.
```

Preparing the client  
Okay, we need some basic configurations to get the client side of OpenVPN working, even when it's unter control of the SMF # mkdir /etc/openvpn  
# mv /tmp/static.key /etc/openvpn

```
# cd /etc/openvpn/  
# ls -l  
total 2  
-rw----- 1 jmoekamp other    636 Feb 27 16:11 static.key  
# chown -R root:root /etc/openvpn  
# chmod 600 static.key
```

Before working with SMF itself

At first i remove this stinking init.d links. We don't need them anymore:

```
# rm /etc/rc0.d/K16openvpn
```

```
# rm /etc/rc1.d/K16openvpn
```

```
# rm /etc/rc2.d/S60openvpn
```

```
# rm /etc/rcS.d/K16openvpn
```

Okay, and now let's hack the startup script...? Wrong! SMF can do many task for you, but this needs careful planing. You should answer yourself some questions:

1. What variables make a generic description of a service to a specific server?
2. How do i start the process? How do i stop them? How can i force the process to reload it's config?
3. Which services are my dependencies? Which services depend on my new service?
4. How should the service react in the case of a failed dependency?
5. What should happen in the case of a failure in the new service.

Okay, let's answer this questions for our OpenVPN service. The variables for our OpenVPN client are the hostname of the remote hosts and the local and remote IP of the VPN tunnel . Besides of this the filename of the secret key and the tunnel device may differ, thus it would be nice to keep them configurable.

Starting openvpn is easy. We have just to start the openvpn daemon with some command line parameters. We stop the service by killing the process. And a refresh is done via stopping and starting the service.

We clearly need the networking to use a VPN service. But networking isn't just bringing up the networking cards. You need the name services for example. So make things easier, the service don't check for every networking service to be up and running. We just define an dependency for the "network" milestone.

As it make no sense to connect to a server without a network it looks like a sensible choice to stop the service in case of a failed networking. Furthermore it seems a good choice to restart the service when the networking configuration has changed. Perhaps we modified the configuration of the name services and the name of the OpenVPN server resolves to a different IP.

What should happen in the case of a exiting OpenVPN daemon? Of course it should started again.

Okay, now we can start with coding the scripts and xml files.

#### The Manifest

Okay, as i wrote before, the manifest is the source of the configuration of a service. Thus we have to write such a manifest. The manifest is a XML file. Okay, at first we obey the gods of XML and do some definitions:# cat openvpn.xml

Okay, at first we habve to name the service:

In this example, we define some simple dependencies. As i wrote before: Without networking a VPN is quite useless, thus the OpenVPN service depends on the reached milestone.

In this part of the manifest we define the exec method to start the service. We use a script to start the service. The %m is a variable. It will be substituted with the name of the called action. In this example it would be expanded to /lib/svc/method/openvpn start. Okay, we can stop OpenVPN simply by sending a SIGTERM signal to it. Thus we can use a automagical exec method. In case you use the :kill SMF will kill all processes in the actual contract of the service. Okay, thus far we've only define the service. Let's define a service. We call the instance theoden2gandalf for obvious names. The service should run with root privileges. After this we define the properties of this service instance like the remote host or the file with the secret keys.

At the end we add some further metadata to this service:

OpenVPN

I saved this xml file to my homedirectory under /export/home/jmoekamp/openvpn.xml.

The exec method's script  
General considerations

Okay, we referenced a script in the exec method. This script is really similar to a normal init.d script. But there are some important differences. As there's no parallel startup of services in init.d most scripts for this system bringup method tend to return as quickly as possible. We have to change this behaviour.

Scripts for transient or standalone services should only return in the case of the successful execution of the complete script or when we've terminated the process. For services under control of the contract mechanism the script should at least wait until the processes of the service generate some meaningful error messages, but they have to exit, as SMF would consider the service startup as failed, when the script doesn't return after

There are some general tips: When you write your own stop method don't implement it in a way that simply kills all processes with a certain name (e.g. pkill "openvpn") this was and is really bad style, as there may be several instances of service. Just using the name to stop the processes would cause unneeded collateral damage. It's a good practice to include the /lib/svc/share/smf\_include.sh. It defines some variables for errorcodes to ease the development of the method scripts.

Implementing a exec method script

We store configuration properties in the Service Component repository. It would be nice to use them for configuration. Thus we have to access them.

Here comes the svcprop command to help: # svcprop -p openvpn/remotehost

svc:/application/network/openvpn:theoden2gandalf

gandalfWith a little bit of shell scripting we can use this properties to use them for starting our processes.

```
#!/bin/sh
```

```
. /lib/svc/share/smf_include.sh
```

```
getproparg() {  
val=`svcprop -p $1 $SMF_FMRI`  
[ -n "$val" ] && echo $val  
}
```

```
if [ -z "$SMF_FMRI" ]; then  
echo "SMF framework variables are not initialized."  
exit $SMF_EXIT_ERR  
fi
```

```
OPENVPNBIN='/opt/csw/sbin/opencvn'
REMOTEHOST=`getproparg openvpn/remotehost`
SECRET=`getproparg openvpn/secret`
TUN_LOCAL=`getproparg openvpn/tunnel_local_ip`
TUN_REMOTE=`getproparg openvpn/tunnel_remote_ip`
DEVICETYPE=`getproparg openvpn/tunneldevice`

if [ -z "$REMOTEHOST" ]; then
echo "openvpn/remotehost property not set"
exit $SMF_EXIT_ERR_CONFIG
fi

if [ -z "$SECRET" ]; then
echo "openvpn/secret property not set"
exit $SMF_EXIT_ERR_CONFIG
fi

if [ -z "$TUN_LOCAL" ]; then
echo "openvpn/tunnel_local_ip property not set"
exit $SMF_EXIT_ERR_CONFIG
fi

if [ -z "$TUN_REMOTE" ]; then
echo "openvpn/tunnel_remote_ip property not set"
exit $SMF_EXIT_ERR_CONFIG
fi

if [ -z "$DEVICETYPE" ]; then
echo "openvpn/tunneldevice property not set"
exit $SMF_EXIT_ERR_CONFIG
fi

case "$1" in
'start')
$OPENVPNBIN --daemon --remote $REMOTEHOST --secret $SECRET --ifconfig $TUN_LOCAL $TUN_REMOTE --dev
$DEVICETYPE
;;

'stop')
echo "not implemented"
;;

'refresh')
echo "not implemented"
;;

*)
echo $"Usage: $0 {start|refresh}"
exit 1
;;

esac
exit $SMF_EXIT_OK
```

I saved this script to my homedirectory under /export/home/jmoekamp/openvpn.

#### Installation of the new Service

Okay, copy the script to /lib/svc/method/:# cp openvpn /lib/svc/method  
# chmod +x /lib/svc/method/openvpnAfter this step you have to import the manifest into the Service Configuration Repository:# svccfg validate /export/home/jmoekamp/openvpn.xml  
# svccfg import /home/jmoekamp/openvpn.xmlTesting itLet's test our brand new service:# ping 172.16.1.2  
^CThe OpenVPN service isn't enabled. Thus there is no tunnel. The ping doesn't get through.

Now we enable the service and test it again.

```
# svcadm enable openvpn:theoden2gandalf
```

```
# ping 172.16.1.2
```

```
172.16.1.2 is alive
```

Voila ... SMF has started our brand new service. When we look into the list of services, we will find it:# svcs

```
openvpn:theoden2gandalf
```

```
STATE      STIME    FMRI
```

```
online     18:39:15 svc:/application/network/openvpn:theoden2gandalf
```

When we look into the process table, we will

```
finde the according process:# /usr/ucb/ps -auxwww | grep "openvpn" | grep -v "grep"
```

```
root      1588  0.0  0.5 4488 1488 ?        S 18:39:15  0:00 /opt/csw/sbin/openvpn --daemon --remote gandalf --secret
```

```
/etc/openvpn/static.key --ifconfig 172.16.1.2 172.16.1.1 --dev tun
```

```
Okay, we doesn't need the tunnel any longer after a
```

```
few day,thus we disable it:# svcadm disable openvpn:theoden2gandalf
```

```
# /usr/ucb/ps -auxwww | grep "openvpn" | grep -v "grep"
```

```
# No process left.
```

Posted by Joerg Moellenkamp in Solaris at 12:31

### **Solaris Features: Service Management Facility - Part 3: Working with SMF**

After so much theory SMF may look a little bit complex but it isn't. For the admin it's really simple. You can control the complete startup of the system with just a few commands.

What's running on the system

At first let's have a look on all services running on the system:

```
# svcs
```

```
legacy_run  10:04:44 lrc:/etc/rc3_d/S84appserv
```

```
disabled    10:04:22 svc:/system/xvm/domains:default
```

```
online      10:03:48 svc:/system/svc/restarter:default
```

```
offline     10:03:54 svc:/network/smb/server:default
```

This is only a short snippet of the configuration. The output of this command is 105 lines long on my system. But you services in several service states in it. For example i hadn't enabled xvm on my system (makes no sense, as this Solaris is already virtualized, and the smb server is still online.

Let's look after a certain service

```
# svcs name-service-cache
```

```
STATE      STIME    FMRI
```

```
online     10:08:01 svc:/system/name-service-cache:default
```

The output is seperated into three columns. The first shows the service state, the second the time of the last start of the service. The last one shows the exact name of the service.

Starting and stoping a service

Okay, but how do i start the service, how do i use all this stuff:let's assume, you want to disable sendmail. At first we check the current state:

```
# svcs sendmail
```

```
STATE      STIME    FMRI
```

```
online     10:23:19 svc:/network/smtp:sendmail
```

Now we disable the service. It's really straight forward:

```
# svcadm disable sendmail
```

Let's check the state again.

```
# svcs sendmail
```

```
STATE      STIME    FMRI
```

```
disabled   10:23:58 svc:/network/smtp:sendmail
```

Okay, a few days later we realize that we need the sendmail service on the system. No problem we enable it again:

```
# svcadm enable sendmail
```

```
# svcs sendmail
```

```
STATE      STIME    FMRI
```

```
online     10:25:30 svc:/network/smtp:sendmail
```

The service runs again. Okay, we want to restart the service. This is quite simple, too

```
# svcadm restart sendmail
# svcs sendmail
STATE      STIME    FMRI
online*    10:25:55 svc:/network/smtp:sendmail
```

```
# svcs sendmail
STATE      STIME    FMRI
online     10:26:04 svc:/network/smtp:sendmail
```

Did you notice the change in the STIME column. The service has restarted. By the way: STIME doesn't stand for "start time". It's a short form for "State Time". It shows, when the actual state of the services was entered.

Okay, now let's do some damage to the system. We move the config file for sendmail, the glorious sendmail.cf. The source of many major depressions under sys admins.

```
# mv /etc/mail/sendmail.cf /etc/mail/sendmail.cf.old
```

```
# svcadm restart sendmail
```

```
# svcs sendmail
STATE      STIME    FMRI
offline    10:27:09 svc:/network/smtp:sendmail
```

Okay, the service went in the offline state. Offline? At first, the maintenance state would look more sensible. But let's have a look in some diagnostic informations. With `svcs -x` you can print out fault messages regarding services.

```
# svcs -x
svc:/network/smtp:sendmail (sendmail SMTP mail transfer agent)
  State: offline since Sun Feb 24 10:27:09 2008
  Reason: Dependency file://localhost/etc/mail/sendmail.cf is absent.
  See: http://sun.com/msg/SMF-8000-E2
  See: sendmail(1M)
  See: /var/svc/log/network-smtp:sendmail.log
```

Impact: This service is not running.

The SMF didn't even try to start the service. There is an dependency implicit to the service.

```
# svcprop sendmail
config/value_authorization astring solaris.smf.value.sendmail
config/local_only boolean true
config-file/entities fmri file://localhost/etc/mail/sendmail.cf
config-file/grouping astring require_all
config-file/restart_on astring refresh
config-file/type astring path
[...]
```

The service configuration for sendmail defines a dependency to the config-file `/etc/mail/sendmail.cf`. Do you remember the definition of the service states? A service stays in offline mode until all dependencies are fulfilled. We renamed the file, the dependencies isn't fulfilled. The restart leads correctly to the "offline state"

Okay, we repair the damage: `# mv /etc/mail/sendmail.cf.old /etc/mail/sendmail.cf`

And now we restart the service

```
# svcadm refresh sendmail
```

```
# svcs sendmail
```

```
STATE      STIME    FMRI
online     10:33:54 svc:/network/smtp:sendmail
```

All is well, the service in online again

Automatic restarting of a service

Okay, let's test another capability of the SMF. The kill the sendmail daemon.

```
# svcs sendmail
```

```
STATE      STIME    FMRI
online     10:33:54 svc:/network/smtp:sendmail
```

```
# pkill "sendmail"
```

```
# svcs sendmail
```

```
STATE      STIME    FMRI
online     10:38:24 svc:/network/smtp:sendmail
```

The SMF restarted the daemon automatically as you can see from the stime-column

Obtaining the configuration of a service

Okay, as i wrote before every service has some configuration in the SMF Service Configuration Repository. You can dump this configuration with the svcprop command. Let's print out the configuration for the name service cache: #

```
svcprop svc:/system/name-service-cache:default
general/enabled boolean false
general/action_authorization astring solaris.smf.manage.name-service-cache
[...]
```

```
restarter/state astring online
restarter/state_timestamp time 1203844081.231748000
general_ovr/enabled boolean true
Dependencies
```

But how do i find out the dependencies between services. The svcadm commands comes to help:

The -d switch shows you all services, on which the service depends. In this example we check this for the ssh daemon. #

```
svcs -d ssh
STATE      STIME    FMRI
disabled   8:58:07  svc:/network/physical:nwam
online     8:58:14  svc:/network/loopback:default
online     8:58:25  svc:/network/physical:default
online     8:59:32  svc:/system/cryptosvc:default
online     8:59:55  svc:/system/filesystem/local:default
online     9:00:12  svc:/system/utmp:default
online     9:00:12  svc:/system/filesystem/autofs:default
```

To check, what services depend on ssh, you can use the -D switch: #

```
svcs -D ssh
STATE      STIME    FMRI
online     9:00:22  svc:/milestone/multi-user-server:default
There is no further service depending on ssh. But the milestone multi-user-server depends on ssh. As long the ssh couldn't started successfully, the multi-user-server milestone can't be reached.
```

Posted by Joerg Moellenkamp in Solaris at 11:44

## **Solaris Features: Service Management Facility - Part 2: The foundations of SMF**

The additional capabilities of the SMF comes at a price. SMF has to know more about your services. Most of the new components of the SMF has to do with this capabilities. So we have to define some foundations before doing some practical stuff.

### **Service and Service Instance**

At first we start with the service and the service instance. This difference is important. The service is the generic definition how a service is started. The service instance is the exact configuration of a service. A good example is a webserver. The service defines the basic methods how to start or stop an apache daemon, the service instance contains the information, how an specific configuration should work (which port to listen on, the position of the config file). A service can define to allow just one instance, as well you can define, you can have multiple instances of a certain service.

But: A service doesn't have to be a long running process. Even a script that simply exits after executing (e.g. for some commands to do network tuning) is a special kind of a service in the sense of SMF.

### **Milestone**

A milestone is somehow similar to the old notion of runlevel. With milestones you can group certain services. Thus you don't have to define each service when configuring the dependencies, you can use a matching milestones containing all the needed services.

Furthermore you can force the system to boot to a certain milestone. For example: Booting a system into the single user mode is implemented by defining a single user milestone. When booting into single user mode, the system just starts the services of this milestone.

The milestone itself is implemented as a special kind of service. It's an anchor point for dependencies and a simplification for the admin. Furthermore some of the milestones including single-user, multi-user and multi-user-server contain methods to execute the legacy scripts in rc\*.d

### Fault Manager Resource Identifier

Every service instance and service instance has a unique name in the system do designate it precicely. This name is called Fault Management Resource Identifier. For example the SSH server is called:svc:/network/ssh:defaultThe FRMI is divided by the : into three parts. The first part designates the resource as an service. The second parts designates the service. The last part designates the service instance. Into natural language: It's the default configuration of the ssh daemon.

But why is this identifier called Fault Manager Resource Identifier? Fault Management is another important feature in Solaris. The FM has the job to react automatically to failure events. The failure of a service isn't much different to the failure of a hard disk or a memory chip. You can detect it and you can react to it. So the Fault Management is tightly integrated into the service mangement.

### Service Model

As i mentioned before, not at all services are equal and they have different requirements to starting them. Thus the System Managemen Facility knows different service models:

#### Transient service

The simplest service model is transient. You can view it as a script that gets executed while starting the system without leaving a long-lived server process. You use it for scripts to tune or config things on your system. A good example is the script to configure the core dumping via coreadm.

A recommendation at this place: Don't use the transient model to transform your old startup scripts. Albeit possible, you loose all the advantages of SMF. In this case it would be easier to use the integrated methods to use legacy init.d scripts.

#### Standalone model

The third service model is the standalone model. The inner workings of this model are really simple. Whenever the forked process exits, SMF will start it again.

#### Contract service

The standard model for services is contract. This model uses a special facility of the Solaris Operating Environment to monitor the processes

#### A short digression: Contracts

Did you ever wondered about the /system/contract filesystems. It's the most obvious sign of the contracts. The contract model ist based on a kernel level construct to manage the relationships between a process and other kernel managed resources. Such resources are processor sets, memory, devices and most important for SMF other processes. Process contracts describe the relation between a process and it's child process. The contract subsystem generates events available to other processes via listeners. Possible events are:

Event

Description

empty

the last process in the contract has exited

process exit

a process in the process contract has exited

core

a member process dumped core

signal

a member process received a fatal signal from outside the contract

hwerr

a member process has a fatal hardware error

Your system already use this contracts. Let's have a look at sendmail.# ptree -c `pgrep sendmail`

```
[process contract 1]
```

```
1 /sbin/init
```

```
  [process contract 4]
```

```
  7 /lib/svc/bin/svc.startd
```

```
    [process contract 107]
```

```
    792 /usr/lib/sendmail -bd -q15m -C /etc/mail/local.cf
```

```
    794 /usr/lib/sendmail -Ac -q15m
```

With the -c option pstree prints the contract IDs of the processes. In our example, the sendmail processes run under the contract ID 107. With ctstat we can lookup the contents of this

```
contract:# ctstat -vi 107
```

```
CTID  ZONEID  TYPE  STATE  HOLDER  EVENTS  QTIME  NTIME
```

```
107   0      process owned 7   0   -   -
```

```
cookie:      0x20
```

```
informative event set: none
```

```
critical event set:  hwerr empty
```

```
fatal event set:    none
```

```
parameter set:     inherit regent
```

```
member processes:  792 794
```

```
inherited contracts: none
```

Contract 107 runs in the global zone. It's an process id and it was created by process number 7 (the svc.startd). There wasn't any events so far. The contract subsystem should only throw critical evens when the processes terminate due hardware errors and when no processes are left. At the moment there are two processes under the control of the contract subsystem (the both processes of the sendmail daemon)

Let's play around with the contracts:# ptree -c `pgrep sendmail`

```
[process contract 1]
```

```
1 /sbin/init
```

```
  [process contract 4]
```

```
  7 /lib/svc/bin/svc.startd
```

```
    [process contract 99]
```

```
    705 /usr/lib/sendmail -bd -q15m -C /etc/mail/local.cf
```

```
    707 /usr/lib/sendmail -Ac -q15m
```

You can listen to the events with the ctwatch:# ctwatch 99

```
CTID  EVID  CRIT  ACK  CTTYPERE SUMMARY
```

Okay, open a second terminal window to your system and kill the both sendmail processes:# kill 705 707

After we submitted the kill, the contract subsystem reacts and sends an event, that there are no processes left in the contract.

```
# ctwatch 99
```

```
CTID  EVID  CRIT  ACK  CTTYPERE SUMMARY
```

```
99   25   crit  no  process  contract empty
```

Besides of ctwatch the event there was another listener to the event: SMF. Let's look for the sendmail processes again.# ptree -c `pgrep sendmail`

```
[process contract 1]
```

```
1 /sbin/init
```

```
  [process contract 4]
```

```
  7 /lib/svc/bin/svc.startd
```

```
    [process contract 103]
```

```
    776 /usr/lib/sendmail -bd -q15m -C /etc/mail/local.cf
```

```
    777 /usr/lib/sendmail -Ac -q15m
```

Et voila, two new sendmail processes with a different process id and a different process contract ID. SMF has done it's job by restarting sendmail.

To summarize things: The SMF uses the contracts to monitor the processes of a service. Based on this events SMF can take action to react on this events. Per default, SMF stops and restart a service, when any member of the contract dumps core, gets a signal or dies due a hardware failure. Additionally the SMF does the same, when there's no member process left in the contract.

#### Service State

Fault management brings us to the next important definition. Every service instance has a service state. This state describes a point in the lifecycle of the process:

##### Service state

##### Description

##### degraded

The service runs, but somehow the startup didn't fully succeeded and thus the service has only limited capabilities

##### disabled

The service was enabled by the admin, and thus SMF doesn't attempt to start it

##### online

The services is enabled and the bringup of the service was successful

##### offline

The service is enabled, but the service hasn't been started so far, as dependencies are not fulfilled.

##### maintance

The service didn't started properly and exited with an error code other than 0. For example because of typos in config files

##### legacy\_run

This is an special service state. It's used by the SMF for services under the control of the restarter for legacy init.d scripts

Each service under the control of the SMF has an service state throughout it whole lifetime on the system.

#### Service Configuration Repository

All the configurations about the services in the Service Configuration Repository. It's the central database regarding the services. This database is backuped and snapshotted in a regular manner. So it's easy to fall back to a known running state of the repository (after you or a fellow admin FOOBARed the service configuration)

#### Dependencies

The most important feature of SMF is the knowledge about dependencies. In SMF you can define two kinds of dependency in a services. which services this service depends on the services that depend on this service This second way to define a dependency has an big advantage. Let's assume, you have a new service. You want to start it before an other service. But you don't want to change the object itself (perhaps, you need this service only in one special configuration and the normal installation doesn't need your new service ... perhaps it's the authentication daemon for a hyper-special networking connection ). By defining, that another service depends on your service, you don't have to

change the other one.

I will show you how to look up the dependencies in the practical part of this tutorial.

#### Master Restarter Daemon and Delegated Restarter

Okay, now you have all the data. But you need someone to do something: For this task you have the SMF Master Restarter Daemon. This daemon reads the Service Configuration Repository and acts accordingly. It starts a services when all it's dependencies are fulfilled. By this simple rule all services will be started in the process of booting until there are no enabled services left in the offline state.

But not all processes are controlled by the Master Restarter. The Master Restarter can delegate this task to other restarters, thus they are called SMF Delegated Restarter Daemons.

#### Delegated Restarter for inetd services

The most obvious example for such a delegated restarter is inetd, the daemon to start network demons only on demand. One important effect of this is a change in behaviour of the inetd. inetd.conf isn't used to control inetd anymore. The Solaris services which were formerly configured using this file are now configured via SMF. So you don't edit the inetd.conf to disable or enable an inetd service. You use the same commands like for all other services.

Enough theory

Enough theory, let's do some practical stuff ...

Posted by Joerg Moellenkamp in Solaris at 11:22

### **Solaris Features: Service Management Facility - Part 1: Introduction**

The Service Management Facility is a quite new feature. But sometimes I have the impression that the most used feature is the capability to use old legacy init.d scripts. But once you use SMF with all its capabilities, you see an extremely powerful concept.

init.d

For a long time, the de-facto standard of starting up services was the init.d construct. This concept is based on startup scripts. Depending on their parametrisation they start, stop or restart a service. The definition of runlevels (what has to be started at a certain stage of booting) and the sequencing is done by linking these startup scripts in a certain directory and the naming of link.

This mechanism worked quite good, but has some disadvantages. You can't define dependencies between the services. You emulate the dependencies by sorting the links, but that's more of a kludge as a solution. Furthermore the init.d scripts run only once. When the service stops, there are no means to start it again by the system (you have to login to the system and restart it by using the init.d script directly or using other automatic methods)

With init.d a service (like httpd on port 80) is just a consequence of running scripts, not a configurable entity in itself.

Service Management Facility

SMF was invented to solve many of the problems of the old startup mechanism. Most problems result from the lack of knowledge of the system about services it's running. What do I need to run my service? Is this service needed for other services? What is the status of a service? Should I restart another service (e.g. database) to circumvent problems in another service (an old web application for example)? Okay, an expert has the knowledge to do such tasks manually ... but do you want to wake up at night, just to restart this fucking old application?

The concepts of SMF enable the admin to put this knowledge into a machine readable format, thus the machine can act accordingly. This knowledge about services makes the SMF a powerful tool to manage services at your system. SMF enables the system to: starting, restarting and stopping services according to their dependencies. Resulting from this the system startup is much faster, as services are started in a parallel fashion when possible. When a service fails, SMF restarts this service. The delegation of tasks like starting, stopping and configuration of services to non-root users and much more. The following tutorial wants to give you some insights to SMF. Have fun!

Posted by Joerg Moellenkamp in Solaris at 07:47

Wednesday, February 27, 2008

## **OEM agreement between Sun and VMware**

VMware again. Sun and VMware announced an OEM agreement: Sun Microsystems (NASDAQ: JAVA) and VMware Inc. (NYSE: VMW) today announced an OEM agreement to expand their virtualization offerings. Starting today, Sun is offering the VMware Infrastructure product suite on Sun hardware systems with full support from Sun. Hope this will result in another usecase for the internal USB ports of the new x86 systems soon.

Posted by Joerg Moellenkamp at 13:01

## **VMware exploit**

My considerations about the security of virtual machines seem to be correct. The Register writes about a VMware Workstation exploit in VMware vuln exposes the perils of virtualization: The exploit uses a specially crafted path name to access folders that are being shared between the host and virtual environments. VMware applications fail to validate the malicious parameters passed from the guest system to VMware's Shared Folders mechanism. The Shared Folders mechanism then hands off the bad data to the host system's file system, which allows the exploit complete access. At the end virtualisation is based on software and software has bugs. Such exploits are inevitable. You have to consider this in your security policies. You can't assume that a virtual server is as secure as a physical one. This is not a problem, you just have to take it into consideration.

Posted by Joerg Moellenkamp at 10:16

## **RE:Trace**

On the foundations of Dtrace the security specialists Tiller Beauchamp and David Weston created a tool for finding vulnerabilities in your code (or writing exploits, depends on your objectives). It uses the mechanisms of DTrace and extends them. As Internetnews writes in Blackhat: Dtrace a Rootkit: Sun's Dtrace application was developed primarily as a tool to help monitor functions on Solaris. According to a pair of security researchers at the Black Hat conference, you can also use Dtrace as the basis for a rootkit-like tool for offensive and defensive security operations. Finding vulnerabilities and writing exploits are different sides of a special case of debugging code. And debugging is exactly the job, we've developed dtrace for. It was really obvious, that someone will use dtrace for such an usecase.

Posted by Joerg Moellenkamp in Solaris at 07:00

## **Hoerempfehlung: The Secret Meeting - Ultrashiver**

Vor einiger Zeit habe ich hier ja mal eine Hoerempfehlung für Vortex von Collide ausgesprochen (Ist das wirklich schon fast drei Jahre her ??). Jetzt haben kaRIN und Statik von Collide und Dean Garcia (von Curve) unter dem Bandnamen "The Secret Meetin" erstmalig zusammengearbeitet. Herausgekommen ist dabei das Debütalbum Ultrashiver.

Beide Bands sind ja in ihren jeweiligen Musikrichtungen wohlbekannte Groessen. Das merkt man dem Werk auch an. Ausfälle gibt es keine, der Standard wird durchgehend hoch gehalten. Und kaRIN hat nunmal eine perfekte Stimme fuer diese Musik. Wenn man an eher dunkeler elektronischer Musik gefallen gefunden hat, so ist diese Musik definitiv ein Reinhoeren wert. Eine Hoerempfehlung fuer einzelne Stuecke gibt es diesmal nicht. Man muss die CD in voller Länge auf sich wirken lassen.

Posted by Joerg Moellenkamp in Music at 06:29

Tuesday, February 26. 2008

**Elisabeth Möllenkamp: 1914-2008**

To my grandmother:it was only one hour ago  
it was all so different then  
there's nothing yet has really sunk in  
looks like it always did  
this flesh and bone  
it's just the way that you would tied in  
now there's no-one home

(from Peter Gabriels "I grieve")

She died yesterday evening. She passed away gently and without pain after a really long life. Gute Reise, Oma, wo immer du jetzt auch bist ...

Posted by Joerg Moellenkamp at 08:48

Monday, February 25. 2008

**links for 2008-02-25**

Putting terabytes of memory into servers, the cheap way | CNET News.com  
(tags: memory)

Posted by del.icio.us in del.icio.us at 12:26

Sunday, February 24. 2008

### **Less known Solaris Features: Installing packages directly from web**

Until i've finalized my next larger article, i want to give spotlight to a really small, but really useful feature: One relatively unknown feature of recent versions of pkgadd is the ability to load packages directly from web. You just have to specify an URL:# pkgadd -d http://www.blastwave.org/pkg\_get.pkg

```
## Downloading...
.....25%.....50%.....75%.....100%
## Download Complete
```

The following packages are available:

```
1 CSWpkgget  pkg_get - CSW version of automated package download tool
  (all) 3.8.4
```

[..]

Installation of was successful.

# That's all. As the packages just have to be accessible by http, you can use an existing internal webserver to serve your favourite "must-have" extra packages and install them directly from there. Okay, and solves the problem nicely to get started with Blastwave without moving around the pkg\_get package via ftp

Posted by Joerg Moellenkamp in Solaris at 07:03

Saturday, February 23. 2008

### **Wahl in Hamburg**

Ich weiss nicht, ob sich die CDU in Hamburg ueber eine moegliche linke Koalition in Hessen beschweren darf. Immerhin hat sich Herr von Beust mit den Stimmen von Herrn Schills Partei Rechtsstaatliche Initiative in 2001 wählen lassen. Soviel zu diesem Thema ...

Posted by Joerg Moellenkamp at 08:22

Friday, February 22. 2008

### **Presentation: End of RAID5**

I often use a thoughtgame when i talk about ZFS: Can we afford to use large harddisks without stronger mechanisms to ensure data validity? The best examples for this problem came from Robin Harris and his Storagemojo.com blog with the article about the reliability of hard disks. I mostly use this example as an anecdote but i transformed it into a draft presentation in the last few evenings:

In my opinion the problem is much worsen than described in Robins blog entry: At first you have the time bomb effect of transient read errors described in the presentation. Furthermore better error correction doesn't solve the problem completely. There are many components between the heads of the harddisks and the CPU. You need checksums for the data from the disk to the CPU. Everything else isn't a complete solution of storing data on rotating rust while ensuring data validity.

Posted by Joerg Moellenkamp in The IT Business at 22:21

### **links for 2008-02-22**

Tip: Prevent iPhoto from opening when you plug in your iPhone - (37signals)  
(tags: iPhone MacOSx iPhoto)

telekinesis - Google Code  
Controlling your mac via iPhone  
(tags: mac iphone)

Posted by del.icio.us in del.icio.us at 12:23

### **SunSTAR**

Someone has nominated me for a SunSTAR interview. You can read it at the website of the GSE Divas (GSE is the Global Systems Engineering, the department at Sun i work for). Thank you for the nomination (whoever did this) and thanks to the Divas.

PS: I didn't know that aalsoup is a favorite to Hamburg ...

Posted by Joerg Moellenkamp in Sun at 06:39

Thursday, February 21. 2008

**links for 2008-02-21**

Big Clusters and Deferred Repair  
Interesting article about availability of big HPC clusters  
(tags: HPC availability)

The Unix Guardian--IBM Certifies Solaris for Selected X64 Servers  
(tags: IBM Solaris Sun)

Posted by del.icio.us in del.icio.us at 12:28

**Further "Less known Solaris features"**

The articles from the "Less known Solaris features" series were quite successful. I writing further articles for this series at the moment. But i think it's time to ask my readers: What topics or features do you want to see in this series? Do you have a feature in Solaris you find cool but underused?

Posted by Joerg Moellenkamp in Sun at 10:00

Wednesday, February 20, 2008

### **Sun Blade X8450**

We had an announcement out of the "No, our blade isn't too big, your's too small"-department today. We've announced the X8450 blade today. It's a really incredible piece of hardware. Up to four quadcore Xeons. 32 DIMM slots. Up to 256 GB of memory. And all the nice features from the Sun Blade 8000 chassis like the PCI Express slots in the chassis.

Posted by Joerg Moellenkamp in Sun at 20:37

Tuesday, February 19, 2008

### **Paul Murphy about Transactional Memory**

Paul has written an really interesting article about the implications of transactional memory: The most obvious implication here is that Rock and its successors will allow Solaris kernel developers to make most these lock processes go away - and for applications that will initially mean simple re-compiles to take advantage of new libraries but in the longer term spark new designs eliminating many of the cycle absorbing complexities of present day multi-threading. The effects to the performance of Solaris alone will be very interesting. Paul is right, there are many locks in Solaris and everything that speeds up the handling of locks and shortens the lost time by waiting of locks will be beneficial.

Posted by Joerg Moellenkamp in Sun at 22:41

### **TSMC for 45 nm SPARC**

Do you remember my article about a presentation held by Rick Hetherington about future directions in SPARC with many confidential informations? One of the most interesting informations was the new foundry for the 45nm procs, as TI itself won't invest in this manufacturing technology soon.

Now there is an official comment about this. EETimes writes in Sun taps TSMC for 45nm CPUs: Sun Microsystems Inc. has chosen Taiwan Semiconductor Manufacturing Co. to make its multi-core processors at 45nm and finer geometries.[...] "We have had engineering teams working for months now on 45nm designs and we will have multiple 45nm products," Azhari said.

Posted by Joerg Moellenkamp in Sun at 08:31

Monday, February 18. 2008

### **Pre-release of Crossbow**

You can download the pre-release of Crossbow at [opensolaris.org](http://opensolaris.org). The release notes sound promising.

Posted by Joerg Moellenkamp in Solaris at 13:43

Sunday, February 17. 2008

## c0t0d0s0.org at Systemnews

My RBAC/Privileges article made it into the sun.systemnews.com. Really cool!

Posted by Joerg Moellenkamp at 20:35

### Aha ... oehm ... oha ...

How evil are you?

Das mit AOL ist allerdings etwas zu hoch gegriffen ... ich würde niemals die Welt mit CDs überschwemmen

Posted by Joerg Moellenkamp in Fundsache at 20:29

## Mirror trees

Posted by Joerg Moellenkamp in Photographie at 19:34

## Less known Solaris Features: /export/home? /home? autofs?

### History

The ever reoccurring question to me at customer sites relatively new to Solaris is: "Okay, on Linux i had my homedirectories at /home. Why are they at /export/home at Solaris?" This is an old hat for seasoned admins, but i get this question quite often. Well, the answer is relatively simple and it comes from the time when started to use NIS and NFS and it had something to do with our slogan "The network is the computer", because it has to do with directories distributed in the network. Okay, we have to go 20 years in the past.

There was a time, long long ago, you worked at your workstation. The harddisk in your was big and it was a time when you didn't need 200 Megabyte for your office package alone. So you and your working group used it for storing their data. But there were several workstations and even some big server for big computational tasks. The users wanted to share the data. Sun invented NFS to share the files between the systems. And as it was a tedious task to distribute all the useraccounts on all the systems, Sun invented NIS (later NIS+, but this is another story).

But the users didn't want to mount their homedirectories on every system. They wanted to login to a system and work with their homedirectory on every system. They didn't want to search it a seperate places depending if it was there own machine or a different one.

So Sun invented the automounter. It found it's way into SunOS 4.0 in 1988. The automounter mounts directories into a system based on a ruleset. In Solaris 2.0 and later the automounter was implemented as a pseudo filesystem called autofs. autofs was developed to mount directories based on rules defined in so-called maps.

There are two of them. At first there is the /etc/auto\_master. To cite the Manual: The auto\_master map associates a directory with a map. The map is a master list that specifies all the maps that autofs should check. At a freshly installed system the file looks like this:

```
[root@gandalf:/net/theoden/tools/solaris]$ cat /etc/auto_master
+auto_master
/net      -hosts      -nosuid,nobrowse
```

```
/home    auto_home   -nobrowse
```

The file /etc/auto\_home is such a map referenced by the master map. To cite the manual again: An indirect map uses a substitution value of a key to establish the association between a mount point on the client and a directory on the server. Indirect maps are useful for accessing specific file systems, such as home directories. The auto\_home map is an example of an indirect map. We will use this map later in this article.

### The use case

Okay, an example. gandalf is the workstation of Waldorf and Statler. theoden is the workstation of Gonzo and Scooter. They have their homedirectories on their own workstation. Sometimes a team uses the workstations of the other teams

and they on a gentleman agreement they allowed each other to do so. But they want to use their homedirectories on the system of the other team.

### Prerequisites

At first we have to export the directories with the real homedirectories on both hosts via NFS.

```
At first on gandalf:[root@gandalf:/etc]$ echo "share -F nfs -d \"Home Directories\" /export/home" >> /etc/dfs/dfstab
[root@gandalf:/etc]$ shareall
[root@gandalf:/etc]$ exportfs
- /export/home rw "Home Directories"
Now we repeat this steps on theoden:[root@theoden:/export/home]$
echo "share -F nfs -d \"Home Directories\" /export/home" >> /etc/dfs/dfstab
[root@theoden:/export/home]$ shareall
[root@theoden:/export/home]$ exportfs
- /export/home rw "Home Directories"
```

Okay, it's important that both hosts can resolve the hostname of the other system. I've added some lines to /etc/hosts in my test installation:

```
10.211.55.201 gandalf
10.211.55.200 theoden
```

### Creating users and homedirectories

Okay, normally you wouldn't create the homedirectories this way. You would use a centralised user repository with LDAP. But that is another real long tutorial.

The userids and usernames of the user has to be equal. At first i create the local users. I use the -m switch for creating the homedirectory with the user.[root@gandalf:~]\$ useradd -u 2000 -m -d /export/home/waldorf waldorf

64 blocks

```
[root@gandalf:~]$ useradd -u 2001 -m -d /export/home/statler statler
```

64 blocks Now i set to the homedirectory of both users to the /home under the control of autofs:

```
[root@gandalf:~]$ usermod -d /home/statler statler
```

```
[root@gandalf:~]$ usermod -d /home/waldorf waldorf
Now i create the the users for the other team. Now without the -m-Switch and directly with the correct homedirectory. The homedirectories come from the other system. So we don't have to create them:[root@gandalf:~]$ useradd -u 2002 -d /home/gonzo gonzo
```

```
[root@gandalf:~]$ useradd -u 2003 -d /home/scooter scooter
Now we switch to Theoden. We do almost the same on this system. We create the accounts for Waldorf and statler without creating a homedirectory. After this we create the local users together with their homedirectories and set them after this to autofs controlled /home:[root@theoden:~]$ useradd
```

```
-u 2001 -d /home/statler statler
```

```
[root@theoden:~]$ useradd -u 2000 -d /home/waldorf waldorf
```

```
[root@theoden:~]$ useradd -u 2002 -d /export/home/gonzo -m gonzo
```

64 blocks

```
[root@theoden:~]$ useradd -u 2003 -d /export/home/gonzo -m scooter
```

64 blocks

```
[root@theoden:~]$ usermod -d /home/gonzo gonzo
```

```
[root@theoden:~]$ usermod -d /home/scooter scooter
```

### Configuring the automounter

Execute the following four commands on both hosts: echo "statler gandalf:/export/home/&" >> /etc/auto\_home

```
echo "waldorf gandalf:/export/home/&" >> /etc/auto_home
```

```
echo "scooter theoden:/export/home/&" >> /etc/auto_home
```

```
echo "gonzo theoden:/export/home/&" >> /etc/auto_home
```

The ampersand is a variable. It stands for the key in the table. So gonzo theoden:/export/home/& translates to theoden:/export/home/gonzo. Now start the autofs on both hosts:

```
[root@theoden:~]$ svcadm enable autofs and [root@gandalf:~]$ svcadm enable autofs
```

### Testing the configuration

Okay, let's login to theoden as User gonzo. Gonzo is a user with a homedirectory local to theoden:\$ ssh

```
gonzo@10.211.55.200
```

Password:

```
Last login: Sun Feb 17 14:16:41 2008 from 10.211.55.2
```

```
Sun Microsystems Inc. SunOS 5.11 snv_78 October 2007
```

```
$ /usr/sbin/mount
```

```
[...]
```

```
/home/gonzo on /export/home/gonzo read/write/setuid/devices/dev=1980000 on Sun Feb 17 14:13:35 2008
Now we try waldorf on theoden. Waldorf doesn't have its homedirectory on theoden, it's on gandalf. $ ssh waldorf@10.211.55.200
```

Password:

## Blog Export: c0t0d0s0.org, http://www.c0t0d0s0.org/

```
Last login: Sun Feb 17 14:17:47 2008 from 10.211.55.2
Sun Microsystems Inc. SunOS 5.11 snv_78 October 2007
$ /usr/sbin/mount
[...]
/home/waldorf on gandalf:/export/home/waldorf remote/read/write/setuid/devices/xattr/dev=4dc0001 on Sun Feb 17
14:17:48 2008 autofs has mounted the /export/home/waldorf automatically to /home/waldorf, the directory we used when
we created the user.
```

Let's crosscheck. We log into gandalf with the user waldorf. Now this user has a local homedir. It's a local mount again.

```
$ ssh waldorf@10.211.55.201
Password:
```

```
Last login: Sat Feb 16 09:12:47 2008 from 10.211.55.2
Sun Microsystems Inc. SunOS 5.11 snv_78 October 2007
$ /usr/sbin/mount
```

```
[...]
/home/waldorf on /export/home/waldorf read/write/setuid/devices/dev=1980000 on Sat Feb 16 09:12:47 2008
```

Explanation for the separated /home and /export/home

The explanation for the existence of /home and /export/home is really simple. I think you got it already. /export/home is the directory where all the local directories are located. /home is the playground for autofs to unify all homedirectories at a central place, where ever they are located.

The /net directory

Did you ever wonder about the /net in the root directory and its job? It's an autofs controlled directory, too. Let's assume you have an /tools/solaris directory at theoden:

```
[root@theoden:/tools/solaris]$ ls -l /tools/solaris
total 0
-rw-r--r-- 1 root root 0 Feb 17 15:21 tool1
-rw-r--r-- 1 root root 0 Feb 17 15:21 tool2
-rw-r--r-- 1 root root 0 Feb 17 15:21 tool3
Share it via NFS [root@theoden:/tools/solaris]$ share -F nfs -d "Tools"
/tools/solaris
```

```
[root@theoden:/tools/solaris]$ share -F nfs
- /export/home rw "Home Directories"
- /tools/solaris rw "Tools"
```

```
[root@theoden:/tools/solaris]$
```

Now change to the other workstation. Look into the directory /net/theoden:

```
[root@gandalf:/net/theoden]$ ls
```

export tools You will notice all the shared directories by theoden. Change into the tools/solaris

```
directory: [root@gandalf:/net/theoden]$ cd tools
```

```
[root@gandalf:/net/theoden/tools]$ ls
```

```
solaris
```

```
[root@gandalf:/net/theoden/tools]$ cd solaris
```

```
[root@gandalf:/net/theoden/tools/solaris]$ ls -l
```

```
total 0
-rw-r--r-- 1 root root 0 Feb 17 2008 tool1
-rw-r--r-- 1 root root 0 Feb 17 2008 tool2
-rw-r--r-- 1 root root 0 Feb 17 2008 tool3
```

```
[root@gandalf:/net/theoden/tools/solaris]$
```

```
[root@gandalf:/net/theoden/tools/solaris]$ mount
```

```
[..]
```

```
/net/theoden/tools/solaris on theoden:/tools/solaris remote/read/write/nosetuid/nodevices/xattr/dev=4dc0002 on Sat Feb
16 10:23:01 2008 Neat isn't it ... it's configured by default, when you start the autofs.
```

Do you want to learn more?

[docs.sun.com -Solaris 10 System Administrator Collection](#)

[How Autofs Works](#)

[Task Overview for Autofs Administration](#)

Posted by Joerg Moellenkamp in Solaris at 16:10

**Weekend ...**

What a week ... thank god it's weekend. Monday and Tuesday working at the booth at the Hamburger Strategietagen. Wednesday meeting with a customer about test equipment and directly driving from there to the next customer to make a two hour stand up comedy ... err .... presentation about our equipment despite of first signs of an upcoming cold (albeit i thought i had my allergic nose again, in 95% this is the correct diagnosis, well ... this time it were the other five percents ). Thursday hacking together a demo case for a certificate based OpenVPN on some Linksys routers (easy stuff, sure, but not when the firmware seems to have a nasty bug). Was a bad idea two write about something similar in my wiki. In between working on the security presentation. Friday two presentations, voice just gone downhill.

Need my vacation. BTW: I won't fly to New York out of personal reasons. I've postponed this trip to end of May or beginning of June. This postponement has some advantages ... given the subprime crisis and the effects to the US currency shopping in New York will be much better then. You have to see the upside potential on any problem ...

Posted by Joerg Moellenkamp in Braindump at 13:24

Saturday, February 16. 2008

### **NuBIT 2008 Präsentation: Das sicherste Betriebssystem**

Das ist die Präsentation zum Thema "Das sicherste Betriebssystem". Es ist die Version, die ich eigentlich halten wollte bevor ich mit Keynote Mist gebaut habe und die Mittagspause zur schnellen Wiederherstellung vom Hirnbackup nutzen musste. Diese Version ist etwas länger. Habe sie eben aus der Time Machine geholt und grob noch mal durchgesehen. Auch hier fehlt wieder die Tonspur ...

Posted by Joerg Moellenkamp in Solaris at 19:10

### **NuBIT 2008 Präsentation: ZFS - Eine Einführung**

Diese Präsentation habe ich gestern im Rahmen der NuBIT gehalten. Eigentlich ist es nur das Gerüst, an dem ich mich während der Präsentation lose festhalte. Daher weiss ich nicht, wie der Vortrag ohne Tonspur wirkt. Ich habe diese Präsentation von Eric Kustarz Präsentation ZFS - The last word in filesystems abgeleitet und eingedeutscht.

Posted by Joerg Moellenkamp in Solaris at 12:52

### **Weapons**

How many massacres at university does the United States need to realize, that the Second Amendment may pose a problem?

Posted by Joerg Moellenkamp in Braindump at 10:10

### **Seltsam**

Ich guck ja zur Entspannung oder aus Langeweile oefters mal auf das Radarbild des Zuerich. Mittlerweile hat man auch ein ganz gutes Auge fuer die Anflugpattern. Was einen aber ein wenig aufmerken laesst: 8 Flugzeuge in verschiedenen Höhen sind gerade kurzfristig im Anflug auf Zuerich beigedreht und fliegen jetzt Warteschleifen .... Darstellungsfehler, Problem in Zuerich ... hmmm ...

Update: Wohl ein Darstellungsfehler .. laut Flughafenwebseite sind einige Maschinen bereits gelandet, die auf der Radar-Website noch als im Flug befindlich zu sehen sind.

Posted by Joerg Moellenkamp at 08:43

### **Solaris 10 Update 5 Beta**

It's too late to apply for the beta program, but this page gives a nice overview about some of the upcoming new features in Update 5 of Solaris 10: Sockets Direct Protocol (SDP) SDP is the industry standard protocol for delivering Sockets-based networking over Infiniband. Persistent Group Reservation for iSCSI target Defines how cluster nodes are reserved and released and maintains state/data when a node fails. Cool, support for cluster storage on Solaris iSCSI is on it's way... iSNS Client for iSCSI target Client for the Internet Storage Name Service IP addressing ability for IBTF interfaces Infiniband again. A number of IB Upper Level Protocols (e.g. RDS, SDP, NFS) require an interface for IP address resolution that meets the RDMA IP CM Service specification in the InfiniBand Architecture Specification 1.2 Annex. SAS multipathing support Solaris multipathing (Sun StorageTek Traffic Manager, also known as MPxIO) support has been added to the Serial-Attached SCSI (SAS) protocol by extending mpt(7D) driver to utilize the existing MPxIO framework. Capping CPU resource usage This feature allows applications to run on any CPU while limiting their CPU usage and provide fine-grained (specified in fractions of a CPU) limits.

Posted by Joerg Moellenkamp in Solaris at 07:30

Friday, February 15, 2008

## **Infiniband in the datacenter**

EETimes wrote an interesting article about some statements of our top management regarding the future architecture of datacenter. In "Sun preps Infiniband for broad net role" EETimes writes: "In most data centers there are three installations often run by three different teams and using systems from three different vendors," said John Fowler, executive vice president of the systems division at Sun. "We think there are technology and economic advantages for bringing these three together," he said at a press event Wednesday (Feb. 13).

Fowler said Sun will roll out in 2008 products to run all data center traffic on Infiniband. (Nice to see Andy and John sharing my opinion;) Joke aside. I talked about something similar with colleagues some month ago while driving to a Sales Kick Off somewhere in the middle of Nowhere and thought and talked about such a solution while designing the network infrastructure for a really big RfP. It's really obvious, that Infiniband will play a bigger role in future. Andy and John have forgotten more about computer systems than I know about computer systems, so it seems my opinion isn't completely weird.

When you look into a datacenter you have multiple fabrics. You have your SAN, you have your IP networks. They are separated. You have your SAN, you have your Ethernets. In the normal mindset TCP/IP is mentally hardwired to Ethernet or ATM and SAN storage is hardwired to FC. But it hasn't to be this way: FC is just SCSI over a serial line. TCP/IP is inherently independent from the transport channel (just think about RFC 1149 ).

Now just step away from the common design principles of a datacenter for a few minutes. You don't want to have a certain interconnect, you want to use certain protocols, SCSI and TCP/IP for example. At the end it doesn't matter how this protocols reached your systems and storage.

All you need is a high speed low latency interconnect. High speed to fulfill all bandwidth needs of all the consolidated fabrics. Low latency to have at least the latency characteristics of the fastest protocol in the fabric.

Infiniband is such an interconnect. The effective throughput of a quad link data rate infiniband (expected for this year) is 32 GBit per second (there is a roadmap up to 96 GBit per Second). More than enough to substitute some Gigabit Ethernet and Fibre Channel ports. The latency of Infiniband is even faster than the one of Fibrechannel. And by the way, the primary usage of Infiniband in HPC is just an accident, Infiniband was designed as an I/O interconnect at first.

All you have to do is to find efficient ways to encapsulate TCP/IP and FC into Infiniband. There are already such methods like SRP, the SCSI RDMA protocol to enable block device transfers over an Infiniband fabric. Solaris has already the capability to use SRP storage. You can download it on the Sun website. TCP/IP over IB is in solaris for quite a while now.

Okay, when all components are finally in place you get a high performance fabric for all jobs without losing performance. Sure, this isn't something for small installations but for bigger installation it's a viable way to get rid of the multiple fabrics.

We talk about datacenter wide deployments here. So the shorter cable length of Infiniband compared with Ethernet or FC isn't much a factor. For connectivity to legacy networks just use some UltraSPARC T2 system with Infiniband on one side and the legacy interconnects on the other side.

By the way, did you really thought, we designed Sun Datacenter Switch 3456 just for a few huge HPC gigs per year?

Posted by Joerg Moellenkamp in Sun at 22:44

## **16 core porn**

last.fm got a new machine for their central database. They made some hardcore hardware porn shots with the machine. Looks like a fully loaded X4450.

Posted by Joerg Moellenkamp at 21:44

## **1962**

Posted by Joerg Moellenkamp at 20:33

### **Freitags in der Bahn**

An manche Dinge im Rahmen der Emanzipation muss man sich ja wirklich erst gewöhnen : Als ich bei der Bundeswehr war (93-95), haben in den Zügen der Bahn Freitags nur Männer (naja, etwas das vielleicht mal in ein paar Jahren mal sowas ähnliches wie ein Mann werden würde) gesessen , die sich über die Vorkommnisse in der Truppe unterhalten haben. Vieles davon eigentlich sehr belangloser Kram. Mit bloeden Sprüchen galore. Seit heute weiss ich: Frauen in Uniform (auch wenn sie in Zivil mit der Bahn fahren) können das genauso gut mit genau der gleichen Inbrunst und sogar bloeden Sprueche sind gleich geblieben ...

Posted by Joerg Moellenkamp in Bahn at 19:27

### **NuBIT**

Habe meine beiden Vorträge bei der NuBIT gehalten ... sind ganz gut gelaufen. Aber da ich schon mit schmerzenden Hals nach Kiel gefahren bin, bin ich jetzt ganz stimmlos. Wer mich am Wochenende anrufen wollte: Fällt aus ... oder ihr haltet einen Monolog Achja, Murphy war auch da: Ich habe auch noch die Hälfte meines zweiten Vortrags verloren. Ich war mir eigentlich sicher, das ich abgespeichert habe ... so nen Mist ... konnte die Folien aber noch wiederherstellen. Wobei: Der Securityvortrag muss sowieso noch verbessert werden ...

Posted by Joerg Moellenkamp in Sun at 18:25

Thursday, February 14, 2008

### **Uninformed**

What should you think about commentators making such errors? Fujitsu-Siemens was about a year late last year getting the dual-core Sparc64-VI chips to market. Okay, just to repeat it again: Fujitsu-Siemens is only a joint venture to distribute systems. They do not develop or manufacture CPU's. The Sparc64 is a development from Fujitsu. Small, but big difference. Timothy, don't ask our executives about Rock, ask them about Sparc64

Posted by Joerg Moellenkamp in Sun at 10:56

Wednesday, February 13, 2008

**links for 2008-02-13**

Sun Introduces The Industry's First Carrier-Grade, 64-Thread Rackmount Server  
We've announced a NEBS certified version for telcos of the Sun T5220 system .  
(tags: NEBS sun)

Posted by del.icio.us in del.icio.us at 12:23

**Opensolaris Developer Preview 2**

Glynn Foster announced the availability of the second developer preview of Opensolaris. You can download the image of the LiveCD here.

Posted by Joerg Moellenkamp in Solaris at 09:49

Tuesday, February 12, 2008

### **Sun acquires innotek**

We acquired an additional open source company. From the press release: Sun Microsystems, Inc. (NASDAQ: JAVA) today announced that it has entered into a stock purchase agreement to acquire innotek, the provider of the leading edge, open source virtualization software called VirtualBox. Virtualbox on desktops complements nicely our xen-based xVM solution for servers.

Posted by Joerg Moellenkamp in Sun at 21:22

### **Bottom up**

Posted by Joerg Moellenkamp in Photographie at 14:31

### **MacOS 10.5.2 out now ...**

Apple published a new version of Leopard. You find the "What's new" list on Apple's website.

Posted by Joerg Moellenkamp in Apple at 11:16

### **Known, but underused Solaris Features: Live Upgrade**

How to change the world ?

Once in a while root saw some imperfections in his world. He had change some things. But root couldn't stop the turning of the world for hours as people lived in this world. Because of root's special powers, root was able to create a second world without people. Thus root created a second world as an exact copy of the old world. And now root was able to work on the imperfections of his world as long as root wanted. Then he behold and all was good. A magic chant late at night when all people slept and the people woke up in the new world.

What's Live Upgrade

Okay, Live Upgrade isn't really a "less known feature", but in the time working at the keyboard at several customer sites, i've got aware of a fact: One of the most simple, but brilliant feature of Solaris is a somewhat unused feature. The feature is called Live Upgrade.

We've got painfully aware of two facts in the past: At first ... yes, we know of our somewhat suboptimal patch process. And: You can't expect updates of the operating environment when you have to bring down the machine for some time. Thus Sun introduced a feature called Live Upgrade.

Okay, Live Upgrade is so easy that nobody has an excuse not to use it. And with 6 GB size of the SOE and 73 GB boot disks minimum "no space" isn't an excuse too

The concept behind Live Upgrade

The basic concept behind Live Upgrade is pretty simple. All mechanisms are grouped around the concept of alternate boot environments. At first you have your running boot environment and a empty slice or disk (the symbol with the thick lines is the active boot environment).

Now you create an alternate boot environment. It's a copy of the actual boot environment. The system still runs on this environment.

The trick is: The update/patch processes doesn't work on the actual boot environment, they use this alternate but inactive boot environment. The running boot environment isn't touched at all.

After the completion of the updating you have an still running boot environment and a fully patched and updated alternate boot environment. Now the boot environments swap their roles with a single command and a single reboot.

After the role swap the old system stays untouched. So, whatever happens with your new installation, you can fall back to your old system. In case you see problems with your new configuration, you switch back the boot environments and you run with your old operating environment.

A hint for testing this

Use some spare equipment. I've used my MacBook Pro for the first try, and it took forever. Do yourself a favour and don't use a virtual environment. At least use a real DVD and not an ISO to prevent the harddisk from jumping around.

Using Live Upgrade without Updating

Okay, I will present two usecases to you: The first one doesn't exactly match to the "upgrade" moniker. It's a solution for a small problem: You've installed your system and after a few weeks you realize, that your filesystem model wasn't the best choice. /export/home too large, / too small. and a separated /var would be nice. Okay, how you separate those filesystems without a longer service interruption. Bringing the system down, moving files around, booting it up is not an option for productive system. Moving while running isn't a good idea.

Live Update is a nice, but simple solution for this problem: Live Upgrade replicates the boot environments by doing a file system copy. The filesystem layout of the old boot environment and the new environment doesn't have to be the same. Thus you can create a filesystem layout with a bigger /, a smaller /export/home and a separate /var. And the best is: The system runs while doing this steps.

In my example I will start with an operating system on a single partition. The partition is located on /dev/dsk/c0d0s0 and has the size of 15 GB.

```
/ on /dev/dsk/c0d0s0 read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=1980000 on Mon Feb 11 21:06:02 2008
```

At the installation time I've created some additional slices. c0d0s3 to c0d0s6. Each of the slices has the size of 10 GB.

Separating the the single slice install to multiple slices is nothing more than using Live Upgrade without upgrading. At first I create the alternate boot environment: # lucreate -c "sx78" -m /dev/dsk/c0d0s3:ufs -m /usr:/dev/dsk/c0d0s4:ufs -m /var:/dev/dsk/c0d0s5:ufs -n "sx78\_restructured"

Discovering physical storage devices

[..]

Populating contents of mount point .

Populating contents of mount point .

Populating contents of mount point .

[..]

Creation of boot environment successful. We've successfully created a copy of the actual boot environment. But we told the mechanism to put / on c0d0s3, /usr on c0d0s4 and /var on c0d0s5. As this was the first run of Live Upgrade on this system the naming of the environment is more important than on later runs. Before this first run, the boot environment has no name. But you need it to tell the process, which environment should be activated, patched or updated. Okay, my actual environment runs with Solaris Express CE build 78, thus I've called it "sx78". The lucreate command set this name to the actual environment. My new boot environment has the name "sx78\_restructured" for obvious reasons.

Okay, now you have to activate the alternate boot environment. # luactivate sx78\_restructured

Saving latest GRUB loader.

Generating partition and slice information for ABE

Boot menu exists.

Generating direct boot menu entries for ABE.

Generating direct boot menu entries for PBE.

[..]

Modifying boot archive service

GRUB menu is on device: .

Filesystem type for menu device: .

Activation of boot environment successful. Now we have to reboot the system. Just use init or shutdown. If you use any other command to reboot the system, Live Upgrade will not switch to new environment:

```
# init 6 Okay, this takes a minute. But let's have a look on the mount table after the boot. # mount
```

```
/ on /dev/dsk/c0d0s3 read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=1980003 on Tue Feb 12 05:52:50 2008
```

[..]

```
/usr on /dev/dsk/c0d0s4 read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=1980004 on Tue Feb 12 05:52:50 2008
```

[...]

```
/var on /dev/dsk/c0d0s5 read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=1980005 on Tue Feb 12 05:53:12 2008Mission accomplished. Okay, but we want to use LiveUpgrading for upgrading, later. Switch back to your old environment:# luactivate sx78Boot the system. And your are back on your old single-slice installation on c0d0s0:
```

```
/ on /dev/dsk/c0d0s0 read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=1980000 on Mon Feb 12 06:06:02 2008
```

Using Live Upgrade for upgrading Solaris Express

With a new Solaris Express Community Edition every week a Live Upgrade procedure is a good practice to update your system to a new release of the operating system.

Okay, i've burned a DVD with the Solaris Express Community Edition Build 81. I want to upgrade the existing boot environment on the three slices. Just to keep the naming in line, i rename it so sx81.# lurename -e sx78\_restructured -n sx81

Renaming boot environment to .

Changing the name of BE in the BE definition file.

Changing the name of BE in configuration file.

Updating compare databases on boot environment .

Changing the name of BE in Internal Configuration Files.

Propagating the boot environment name change to all BEs.

Boot environment renamed to .You don't have to rename it, you just could use the old name. But why should you confuse your fellow admins by calling your Build 81 boot environment sx78\_restructured.

Okay, now start the upgrade. My installation DVD was mounted under /cdrom/sol\_11\_x86 by Solaris and i want to upgrade the sx81 boot environment. This will take a while. Do this overnight or go shopping or play with your children. Your system is still running and the process will not touch your running installation:# luupgrade -u -n sx81 -s

```
/cdrom/sol_11_x86
```

Copying failsafe kernel from media.

Uncompressing miniroot

[...]

The Solaris upgrade of the boot environment is complete.

Installing failsafe

Failsafe install is complete.Okay. Let's check the /etc/release before booting into the new environment:# cat /etc/release

Solaris Express Community Edition snv\_78 X86

Copyright 2007 Sun Microsystems, Inc. All Rights Reserved.

Use is subject to license terms.

Assembled 20 November 2007Activate the new boot environment:# luactivate sx81

Saving latest GRUB loader.

Generating partition and slice information for ABE

Boot menu exists.

Generating direct boot menu entries for ABE.

Generating direct boot menu entries for PBE.

[...]

Modifying boot archive service

GRUB menu is on device: .

Filesystem type for menu device: .

Activation of boot environment successful.Eject the installation DVD and reboot the system:# eject cdrom /dev/dsk/c1t0d0s2 ejected

```
# init 6Wait a minute, login to the system and let's have a look at /etc/release again: bash-3.2$ cat /etc/release
```

Solaris Express Community Edition snv\_81 X86

Copyright 2008 Sun Microsystems, Inc. All Rights Reserved.

Use is subject to license terms.

Assembled 15 January 2008By the way, the system runs on the three separated slices now:/ on

```
/dev/dsk/c0d0s3 read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=1980003 on Tue Feb 12 07:22:32 2008
```

[..]

/usr on /dev/dsk/c0d0s4 read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=1980004 on Tue Feb 12 07:22:32 2008

[..]

/var on /dev/dsk/c0d0s5 read/write/setuid/devices/intr/largefiles/logging/xattr/onerror=panic/dev=1980005 on Tue Feb 12 07:22:54 2008

Neat, isn't it ?

Do you want to learn more ?

Documentation

Solaris 10 8/07 Installation Guide: Solaris Live Upgrade and Upgrade Planning >> Upgrading With Solaris Live Upgrade

Others

Solaris[™] Live Upgrade Software: Minimum Patch Requirements(the infodoc formerly known as 72099)

Posted by Joerg Moellenkamp in Solaris at 08:06

## **Review of the T5120**

Octave Orgeron did a really good review of the T5120 server. Octave closes his findings with: The T5120 and the T5220 offer many unique and exciting features that set it apart from the competition. The UltraSPARC-T2 processor with 8 cores, 64 threads, 10Gb Ethernet, PCI-E, and cryptographic features are revolutionary in the computing industry. It is amazing to think that not long ago, it would have taken a much larger and more expensive solution to equal the features and benefits of these servers.

Posted by Joerg Moellenkamp in Sun at 07:01

Monday, February 11. 2008

## **Burning Sky**

Posted by Joerg Moellenkamp in Photographie at 20:23

Sunday, February 10, 2008

## **Rock**

Perhaps you wondered a little bit about the fact, that i didn't wrote about the reports about the delayed Rock processor. I wrote an article, but this contained some internal information, as but i'm still not an official leak (i should talk with Jonathan about that ), i just deleted it. Okay, the delay of Rock was a rumour long before and even the sparrows and the nightingale sang that from the roofs. As I wrote before: That isn't much a problem, as we have two strong new procs in this year: The Quadcore Jupiter Sparc64 and the Victoria Falls boxes.

Those offerings were the reason we didn't got butchered by the analysts. It isn't the situation where we wasn't able to compete on the proccessor level at UltraSPARC III vs. Power4 times (interestingly we won many customer benchmarks even at that time despite of the disadvantage in the proc field. A good point for the old speaking: You buy systems and knowledge, not processors). That was really a hard time for us. Don't want them back ...

Today: Single proc Niagara 2 is able to compete with low end Power systems for many workloads, the M-class systems are competitive with the mid and high end offerings of IBM. And VF and Jupiter will make the situation even more comfortable soon. Victoria Falls will open a complete new market for us this year. Think about a 256 thread OLTP system. Think about a 256 thread webserver. Think about a 256 thread Lotus server. Think about an 256 thread SAP application server. But i wrote about this earlier.

Furthermore: Delays in the processor business are normal, not an exception. Think about Power6 (it came late and despite of reports of IBM friendly commentators the 4.7 Ghz is still largely missing in action). Think about Itanium ("The next version will kill all other procs! Really!"). An delayed or underdelivering Tukwila would be a disaster for HP. And the problems with AMD are in my opinion the most sad story of the business. I'm still strongly opinionated in the direction of AMD as the much better x64 architecture (and Quickpath and integrated memory controllers in next-generation Intel CPUs support that, they just call it different than AMD) , but AMD leaved out a window of opportunity to hurt Intel's marketshare with the delay of Barcelona. I never talk about competitors proc slipages at customer sites (such arguments transform to boomerangs to easily most of the times), i only hate benchmarking with procs unavailable to the masses ...

Some commentators (interestingly not the media itself, those comments come from the "reader comments"-section) suggest that we kill Rock in favour with a longer agreement with Fujitsu. Well, this is utter nonsense from outsiders. We did the tapeout, we have running and stable silicon, we have just a few things left to solve - in short: We did the hard part already, why should we kill of Rock now? Furthermore, we need Rock in the long term. This proc is important to keep the SPARC franchise healthy.

Well, i would have prefered Rock at the original schedule .... because of the fun to shock the competitors with the performance of the proc, but hey ... the difference won't be that extremely large (you know, competitors doesn't sit still), but we will be able to kick some serious butts to have fun with the systems And remember: Delays in the proc business are normal, not an exception. That's true for every architecture.

Posted by Joerg Moellenkamp in Sun at 13:04

## **links for 2008-02-10**

AIR TRAFFIC around Zuerich  
TRACON and Google Maps mashup  
(tags: airplane cool fliegen googlemaps)

How Rainbow Tables work  
(tags: article crypto cryptography encryption hash security)

Localwuerstchens Tagebuch

Frau Localwurst (Ich mag den Namen immer noch nicht ) hat ihrem Blog ein neues Aussehen gegeben ... gut geworden :o)

Die U4 der Hamburger U-Bahn  
Webseite zur Ole-von-Beust-Gedaechtnis-UBahn U4, die leider keinen Anschluss zur  
Ole-von-Beust-Gedaechtnis-Elbphilharmonie haben wird, sondern da wohl den Ole-von-Beust-Gedaechtnislaufsteg  
vorweisen wird.  
(tags: Hamburg U-Bahn)

Posted by del.icio.us in del.icio.us at 12:27

Saturday, February 9, 2008

### **links for 2008-02-09**

Solaris TCP Latency for HPC

This is important for software like memcached, too.

Interposing on malloc

How to write your own interposer library

Posted by del.icio.us in del.icio.us at 12:27

### **Coverflow in Finder**

I wasn't sure about the usefulness of Coverflow in Finder. But now i have a good use case for it. The pdf's from docs.sub.com doesn't have a speaking name (like 816-4557.pdf). I have dozens of such documents on my notebook. With Coverflow it's much easier to find the correct document.

Posted by Joerg Moellenkamp in Apple at 10:26

Friday, February 8. 2008

## Problems with Akismet

To keep spam outside my blog, i configured my software to use Akismet. This is a centralised AntiSpam service. This worked reasonably well, thus i configured an automatic rejection of comments declared as spam by Akismet.

After reading this article in Alec Muffet's blog i searched for comments from Sun employees in the logs and found some of them rejected by Akismet.

Sorry for that. I recovered them from the logfiles. Furthermore i deactivated the automatic rejection.

Posted by Joerg Moellenkamp at 23:11

## Rauchverbot

Kommentar einer Bekannten, die als Bedienung ihr Studium finanziert: "Mir wäre es lieber, wenn die Gäste wieder rauchen. Jetzt riecht man die Gäste, ihre Parfüms und die Schweissflecken." Sie ist wohlgermerkt Nichtraucherin.

Posted by Joerg Moellenkamp in Fundsache at 22:12

## Solaris Containerleitfaden 2.0

Detlef Drewanz und Ulrich Gräf haben den Containerleitfaden in der Version 2.0 fertig gestellt. Wesentliche Neuerungen dieser Ausgabe sind IP Instances, neues Ressource Management, Live Upgrade mit Zonen und Branded Zones.

Posted by Joerg Moellenkamp in Solaris at 13:37

## links for 2008-02-08

Emergency Zombie Defense Station - Die wunderbare Welt von Isotopp  
(tags: zombies humor)

Posted by del.icio.us in del.icio.us at 12:21

## Less known Solaris Features: Signed binaries

One of problems in computer security is the validation of binaries: Is this the original binary or is it a counterfeit binary? Since Solaris 10 Sun electronically signs the binaries of the Solaris Operating Environment. You can check the signature of the binaries with the elf-sign tool.  
[root@gandalf:/etc]\$ elfsign verify -v /usr/sbin/ifconfig

elfsign: verification of /usr/sbin/ifconfig passed.

format: rsa\_md5\_sha1.

signer: CN=SunOS 5.10, OU=Solaris Signed Execution, O=Sun Microsystems Inc. Obviously you have to trust the elfsign. But you can check it, when you boot the system from a trusted media (like a original media kit or a checksum validated iso-image. This enables you to check the signature of the elfsign independently from the system.

By the way: This certificate and the signature is very important for crypto modules. The crypto framework of solaris just loads modules signed by Sun to prevent the usage of malicious modules (for example to read out the key store and send it somewhere) into the framework.

Posted by Joerg Moellenkamp at 11:23

## zZz is playing: Grip

**Blog Export: c0t0d0s0.org, <http://www.c0t0d0s0.org/>**

Posted by Joerg Moellenkamp in Fundsache at 10:38

Thursday, February 7, 2008

## ISSCC 08 paper about Rock

The paper for the Rock presentation (held on Monday at the ISSCC) is now online. A good read ...

Posted by Joerg Moellenkamp in Sun at 22:16

## Spiegel

Was muss in einem Menschen vorgehen, das dieser wegen einem Verhalten angepöbelt ist, das man nur gespiegelt hat. Also einfach mal das gleiche Verhalten an den Tag, wie es auch einem selbst von von dieser Person gezeigt wird ...

Posted by Joerg Moellenkamp in Braindump at 21:08

## Less known Solaris features: IPsec

The secrets of root

There were other systems with other roots, other priests and other believers. But how should they communicate with each other without put their secrets in danger. Some chants and documents were really precious. The messengers of root had to be sure, that they gave the chants to the roots with the same belief. But the messengers had another problems: There were many bandits on their way to the other systems. And it was unthinkable that a mere bandit would know the secrets of root.

Foundations

This will be the only article in this series without an explanation of the technologies. I could write for days about it and still leaving out important things. Instead of this, please look at the links section at the end of the article.

Only some short words about this topic. Encryption is essential in networking. The Internet is an inherently insecure media. You have to assume, that you don't talk to the right person as long as you didn't authenticated him, you have to assume, that the data will be read by someone as long as you won't encrypt the data.

IPsec solves these problems. I won't tell you that IPsec is an easy protocol. The stuff around IPsec is defined in several RFC and the documentation is rather long. The encryption itself isn't the complex part. But you need a key to encrypt your data. And there starts the complexity. It's absolutely essential, that this key stays secret. How do you distribute keys through an inherently insecure transport channel. And the next problem: How do you negotiate the encryption mechanism? And how to ensure, that you talk with the right system? Problems ... problems ... problems!

IPsec solves these problems. IPsec isn't a single protocol. It's a suite of protocols consisting of Internet Security Association and Key Management Protocol, this protocol builds on the protocol for Internet Key Exchange, this protocol is based on the Oakley protocol. And there is a whole wad of further protocols and mechanisms and algorithms to ensure secure communication.

IPsec in Solaris

Okay, but now I want to give you an example to configure IPsec on Solaris. Although the matter is complex, it isn't hard to use IPsec within Solaris. IPsec is a rather old feature in Solaris. We've introduced it in Solaris 8 and improved the implementation since this time. So this implementation is now in year eight of its availability.

Example

The task for this example is to secure all traffic between two hosts. I've used two VM with Solaris Express Build 78 for this configuration. (You need at least a Solaris 10 Update 4 for this tutorial. In this update the ipsecconf command was introduced making the configuration much easier) theoden has the IP number 10.211.55.200. Gandalf has the IP number 10.211.55.201. I don't want to use manual keying, instead of this the example will use self signed certificates.

Prepare the installation

At first: Obviously you have to change hostnames and IP numbers and names for certificate corresponding to your own site.

## Blog Export: c0t0d0s0.org, http://www.c0t0d0s0.org/

We have to work on two hosts. It's a best practice to open up two terminal windows. Get root in both of them. To keep things apart use the bash shell and modify the shell prompt: # bash

```
# PS1="[u@h:w]"$ "
```

[root@gandalf:~]\$ Look at the login prompt in the examples. They designate on which system you have to work on.

Okay ... at first you have to ensure, that the names of the systems can be resolved. It's a good practice to put the names of the systems into the /etc/hosts:::1 localhost loghost

```
127.0.0.1 localhost loghost
```

```
10.211.55.201 gandalf
```

```
10.211.55.200 theoden
```

Okay, we don't want manual keying or some stinking preshares keys. Thus we need to create keys. Login to gandalf and assume the root role: [root@gandalf:~]\$ ikecert certlocal -ks -m 1024 -t rsa-md5 -D "C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=gandalf" -A IP=10.211.55.201

Creating private key.

Certificate added to database.

```
-----BEGIN X509 CERTIFICATE-----
```

```
MIICozCCAaSgAwIBAgIFAJRpUUKwDQYJKoZIhvcNAQEEBQAwtZELMAkGA1UEBhMC
```

```
[ ... some lines omitted ... ]
```

```
oi4dO39J7cSnooqnekHjajn7ND7T187k+f+BVcFVbSenIzblq2P0u7FIgljdlv0=
```

```
-----END X509 CERTIFICATE-----
```

Do the same on the other host. [root@theoden:~]\$ ikecert certlocal -ks -m 1024 -t

```
rsa-md5 -D "C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=theoden" -A IP=10.211.55.200
```

Creating private key.

Certificate added to database.

```
-----BEGIN X509 CERTIFICATE-----
```

```
MIICozCCAaSgAwIBAgIFAIRuR5QwDQYJKoZIhvcNAQEEBQAwtZELMAkGA1UEBhMC
```

```
[ ... some lines omitted ... ]
```

```
UHJ4P6Z0dtjnToQb37HNq9YWFRguSsPQvc/Lm+S9cJCLwINvg7NOXXgnSfY3k+Q=
```

```
-----END X509 CERTIFICATE-----
```

You need the output of this commands later, so past them to a texteditor or at a save place ...

### Configuration of IPsec

Okay, now we have to tell both hosts to use IPsec when they talk to each other: [root@gandalf:~]\$ echo "{laddr gandalf raddr theoden} ipsec {auth\_algs any encr\_algs any sa shared}" >> /etc/inet/ipsecinit.conf

This translates to: When i'm speaking to theoden, i have to encrypt the data and can use any negotiated and available encryption algorithm and any negotiated and available authentication algorithm.

Such an rule is only valid on one direction. Thus we have to define the opposite direction on the other host to enable bidirectional traffic: [root@theoden:~]\$ echo "{laddr theoden raddr gandalf} ipsec {auth\_algs any encr\_algs any sa shared}" >> /etc/inet/ipsecinit.conf

Okay, the next configuration is file is a little bit more complex. Go into the directory /etc/inet/ike and create a file config with the following content:

```
cert_trust "10.211.55.200"
```

```
cert_trust "10.211.55.201"
```

```
p1_xform
```

```
{ auth_method preshared oakley_group 5 auth_alg sha encr_alg des }
```

```
p2_pfs 5
```

```
{
```

```
label "DE-theoden to DE-gandalf"
```

```
local_id_type dn
```

```
local_id "C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=theoden"
```

```
remote_id "C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=gandalf"
```

```
local_addr 10.211.55.200
```

```
remote_addr 10.211.55.201
```

```
p1_xform
```

```
{auth_method rsa_sig oakley_group 2 auth_alg md5 encr_alg 3des}
```

```
} This looks complex, but once you've understand this it's quite easy: cert_trust "10.211.55.200"
```

```
cert_trust "10.211.55.201"
```

We use self-signed certificate. The certificate isn't signed by an independent certification authority. Thus there is no automatic method to trust the certificate. You have to configure the ike explicitly to trust this certificates. This both lines tell the ike to trust the certificates with the alternate name 10.221.55.200 and

10.211.55.201. Where did this alternate names came from? You set them! Look in the command line for creating the certificate. You defined this name by using the -a switch.label "DE-gandalf to DE-theoden"

```
local_id_type dn
```

```
local_id "C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=gandalf"
```

```
remote_id "C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=theoden"
```

Now you define an key exchange. You have to give each connection an unique name. After this you define, what part of the certificate is used to authenticate the remote system. In this example we use the distingushed name. The local system identifies itself with the certificate named C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=gandalf to a remote system, and expect a trusted certificate with the distingushed name C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=theoden.local\_addr 10.211.55.200

```
remote_addr 10.211.55.201No the iked knows the ip addresses of the local and the remote host.
```

```
p1_xform
```

```
{auth_method rsa_sig oakley_group 2 auth_alg md5 encr_alg 3des}
```

After defining the authentication credentials, we have to define how the system should communicate. This line means: Use the certificates to authenticate the other system. The key determination protocol is based on a prime number. We use md5 as the groundlaying algorithm to authenticate and 3des for encryption. This is the part where you configure the methods for authentication and encryption. They have to be the same on both hosts, otherwise they won't be able to negotiate to a common denominator thus you won't be able to communicate between the both hosts at all.

Now we do the same on the other system.[root@gandalf:/etc/inet/ike]\$ cd /etc/inet/ike

```
[root@gandalf:/etc/inet/ike]$ cat conf
```

```
cert_trust "10.211.55.200"
```

```
cert_trust "10.211.55.201"
```

```
p1_xform
```

```
{ auth_method preshared oakley_group 5 auth_alg sha encr_alg des }
```

```
p2_pfs 5
```

```
{  
label "DE-gandalf to DE-theoden"  
local_id_type dn  
local_id "C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=gandalf"  
remote_id "C=de, O=moellenkamp, OU=moellenkamp-vpn, CN=theoden"
```

```
local_addr 10.211.55.201
```

```
remote_addr 10.211.55.200
```

```
p1_xform
```

```
{auth_method rsa_sig oakley_group 2 auth_alg md5 encr_alg 3des}
```

}Obviously you have to swap the numbers for the local and remote system and you have to assign a unique label to it.

Okay, we are almost done. But there is still a missing but very essential thing when you want to use certificates. We have to distribute the certificates of the systems.[root@gandalf:/etc/inet/ike]\$ ikercert certdb -l

```
Certificate Slot Name: 0 Key Type: rsa
```

```
(Private key in certlocal slot 0)
```

```
Subject Name:
```

```
Key Size: 1024
```

```
Public key hash: 28B08FB404268D144BE70DDD652CB874A
```

At the beginning there is only the local key in the system. We have to import the key of the remote system. Do you remember the output beginning with -----BEGIN X509 CERTIFICATE----- and ending with -----END X509 CERTIFICATE-----? You need this output now.

The next command won't come back after you hit return. You have to paste in the key. On gandalf you paste the output of the key generation on theoden. On Theoden you paste the output of the key generation on gandalf. Let's import the key on gandalf [root@gandalf:/etc/inet/ike]\$ ikercert certdb -a

```
-----BEGIN X509 CERTIFICATE-----
```

```
MIICozCCAaSgAwIBAgIFAIRuR5QwDQYJKoZIhvcNAQEEBQAwtZELMAkGA1UEBhMC
```

```
UHJ4P6Z0dtjnToQb37HNq9YWFRguSsPQvc/Lm+S9cJCLwINVg7NOXXgnSfY3k+Q=
```

```
-----END X509 CERTIFICATE-----
```

[root@gandalf:/etc/inet/ike]\$After pasting, you have to hit Enter once and after this you press Ctrl-D once. Now we check for the sucessful import. You will see two certificates now.[root@gandalf:/etc/inet/ike]\$ ikercert certdb -l

## Blog Export: c0t0d0s0.org, http://www.c0t0d0s0.org/

Certificate Slot Name: 0 Key Type: rsa  
(Private key in certlocal slot 0)  
Subject Name:  
Key Size: 1024  
Public key hash: 28B08FB404268D144BE70DDD652CB874

Certificate Slot Name: 1 Key Type: rsa  
Subject Name:  
Key Size: 1024  
Public key hash: 76BE0809A6CBA5E06219BC4230CBB8B8  
Okay, switch to theoden and import the key from gandalf on this system.[root@theoden:/etc/inet/ike]\$ ikecert certdb -l

Certificate Slot Name: 0 Key Type: rsa  
(Private key in certlocal slot 0)  
Subject Name:  
Key Size: 1024  
Public key hash: 76BE0809A6CBA5E06219BC4230CBB8B8

```
[root@theoden:/etc/inet/ike]$ ikecert certdb -a
-----BEGIN X509 CERTIFICATE-----
MIICozCCAaSgAwIBAgIFAJRpUUKwDQYJKoZIhvcNAQEEBQAwtZELMAkGA1UEBhMC
```

```
oi4dO39J7cSnooqnekHjajn7ND7T187k+f+BVcFVbSenIzblq2P0u7FIgljdlv0=
-----END X509 CERTIFICATE-----
```

```
[root@theoden:/etc/inet/ike]$ ikecert certdb -l
Certificate Slot Name: 0 Key Type: rsa
(Private key in certlocal slot 0)
Subject Name:
Key Size: 1024
Public key hash: 76BE0809A6CBA5E06219BC4230CBB8B8
```

Certificate Slot Name: 1 Key Type: rsa  
Subject Name:  
Key Size: 1024  
Public key hash: 28B08FB404268D144BE70DDD652CB874  
Activate the configuration  
Okay, now we have to activate this configuration on both systems:[root@gandalf:/etc/inet/ike]\$ svcadm enable ike  
[root@gandalf:/etc/inet/ike]\$ ipsecconf -a /etc/inet/ipsecinit.conf  
and [root@theoden:/etc/inet/ike]\$ svcadm enable ike  
[root@theoden:/etc/inet/ike]\$ ipsecconf -a /etc/inet/ipsecinit.conf  
Check the configuration

Login into theoden as root and start a packet sniffer:

```
[root@theoden:~]$ snoop host gandalf
Using device ni0 (promiscuous mode)
Okay ... now login to gandalf and start some pings to theoden: [root@gandalf:~]$
ping theoden;ping theoden;ping theoden
```

```
theoden is alive
theoden is alive
```

theoden is alive  
Okay, theoden can speak with gandalf and vice versa. But is it encrypted? In the terminal window for theoden the following output should be printed: gandalf -> theoden ESP SPI=0x1d2c0e88 Replay=4

```
theoden -> gandalf ESP SPI=0x84599293 Replay=4
gandalf -> theoden ESP SPI=0x1d2c0e88 Replay=5
theoden -> gandalf ESP SPI=0x84599293 Replay=5
gandalf -> theoden ESP SPI=0x1d2c0e88 Replay=6
theoden -> gandalf ESP SPI=0x84599293 Replay=6
```

ESP stands for Encapsulated Security Payload. Mission accomplished.

Do you want to learn more?

Documentation

System Administration Guide: IP Services >> IP Security

IPsec at Wikipedia

IPsec

Internet Key Exchange

Oakley Protocol  
Internet Security Association and Key Management Protocol

Other  
An Illustrated Guide to IPsec

Posted by Joerg Moellenkamp in Solaris at 18:51

**links for 2008-02-07**

Das Leben sollte mit dem Tod beginnen  
(tags: humor)

Posted by del.icio.us in del.icio.us at 12:21

Wednesday, February 6, 2008

### **SAS Target Mode for Opensolaris**

The packages of the Common Multiprotocol SCSI Target have been updated. You will find them at the project page [opensolaris.org](http://opensolaris.org) - COMSTAR. This update introduces some interesting new features: SAS Target support. Added port provider for LSI 1068 series of HBAs. See the COMSTAR - mptt page for more information. Support for dynamic LUN expansion and Thin Provisioning in sbd. See COMSTAR - thin provisioning for more information.

Posted by Joerg Moellenkamp at 20:08

### **Forrester about Solaris**

Forrester published a paper titled with European Financial Services Architecture shows Clear Strategic Direction. There are some really interesting takeaways in it: Solaris is back on the winner's podium. [...] - thus making Solaris the third most strategic operating system [...]

(page 3 of the document) and Linux has lost traction [...] While this puts Linux in a strong forth place - only two points behind Solaris - it is also a surprisingly weaker position than in the previous survey [...]

(page 4 of the document)

Posted by Joerg Moellenkamp in Solaris at 19:37

### **Solaris Express Developer Edition 1/08**

There is a new Solaris Express Developer Edition. The 1/08 release has some nice new features:  
Solaris (Solaris Express Community Edition build 79b) Sun xVM Server - provides virtual machine technology derived from the Xen open source community work that supports running Windows, Linux, Solaris guest operating systems on a Solaris Express Developer Edition server. CIFS Server - a native, kernel-based service supporting integrated file sharing in Microsoft Windows networks. Solaris Installer- 3 clicks minimum to upgrade, 6 screens for full install. Automatic provisioning for the use of Solaris Live Upgrade if Solaris is given at least 21GB of disk. Web applications stack (Apache 2.2.6, MySQL 5.0.45, PHP 5.2.4, PostgreSQL 8.2.5, Ruby 1.8.6, Squid 2.6 etc.)... and much more You can download it on the Sun website .

Posted by Joerg Moellenkamp in Solaris at 08:49

Tuesday, February 5. 2008

## **Die Bahn fördert die Arterhaltung**

Obwohl ich einen fahrbaren Untersatz mein Eigen nenne, fahre ich gerne mit S-Bahn. Ich habe einfach keine Lust mich durch den Verkehr zu quälen fuer eine Strecke, auf der der Motor nur maessig warm wird. Ausserdem würde man eine Menge verpassen. Wo findet man sonst Menschen mit so gegensätzlichen Lebensentwürfen auf einem Haufen sich gegenseitig durch die pure Anwesenheit nerven?

Manchmal ... wirklich selten hört man aber auch Sätze, bei denen man sofort überlegt, ob das nun wirklich gesagt worden ist. Gestern sogar über die Lautsprecher des Bahnsteigs an der Haltestelle Hammerbrook. Man muss dazu wissen, das die Bahn für den Valentinstag in das Datinggeschäft eingestiegen ist: Es gibt naemlich nun den Flirtexpress. In Hamburg heisst das , das ein Zug voller einsamer Herzen in den Zug steigt und vom Berliner Tor nach Aumühle und wieder Zurueck fährt (vulgo: der stadttöstliche Teil der S21) und einen ganzen Haufen genauso einsamer, aber nun zusätzlich enttäuschter und genervter Herzen wieder ausspuckt. Und nebenbei angemerkt: Diese Strecke ist so ziemlich die langweiligste Strecke, die Hamburg zu bieten hat.

Ich habe ja schon häufiger gehört, das Hamburg vergeist und versinglet, aber das es so schlimm geworden ist, das jetzt auch die Bahn um die Arterhaltung bemüht, wusste ich noch nicht. Andererseits .... ohne eine solche Aktion müsste man auf Durchsagen wie "Vorsichtig bei der Einfahrt, sie könnten sich verlieben" nebst ungläubigen Gesichter der am Bahnsteig stehenden Menschen ob des gerade Gehörten verzichten.

PS: Ich möchte mir gar nicht vorstellen, wie sich die Bahn Dating vorstellt. Die Bahn glaubt ja auch, das ein Minireisepack Orangensaft eine Entschuldigung darstellt, das man gerade 45 Minuten zu spät an seinem Ziel angekommen ist ...

Update : Die alte Überschrift "Der Bahn fördert die Arterhaltung" ist in der Entstehung leicht erklart. Ich dachte das wäre der HVV, der dafür verantwortlich ist. Ist aber die Bahn. Man sollte dann nur noch den Artikel ändern ...

Posted by Joerg Moellenkamp in Bahn at 15:02

## **links for 2008-02-05**

N-JOY - N-JOY Abstrait  
(tags: music)

IBM responds to Microsoft: OOXML is "technically inferior"  
(tags: ooxml odf openoffice microsoft)

WTF per minute  
The only valid measurement of code quality  
(tags: programming geek humour comics)

Posted by del.icio.us in del.icio.us at 12:22

## **Jon Stokes about Rock**

Jon Stokes of Ars Technica writes in Sun: Can you smell what the Rock is cookin'?: In the end, I can't say that I'm really sold on Sun's very aggressive use of speculative execution, but I will say that Rock is one of the most interesting and novel processors that I've seen in 10 years of covering this space. In its own way, it's every bit as exotic as IBM's Cell processor, but because all of that exoticism is hidden from the programmer it won't be nearly as difficult for developers

to deal with.

Posted by Joerg Moellenkamp in Sun at 08:58

## **Hundehaufen**

Am Anfang waren die Überwachungskameras dafür da um Verbrechen wie Körperverletzung oder einen Banküberfall besser aufklären zu können oder um wirklich Sicherheit zu schaffen, wie beispielsweise auf den Bahnsteigen, damit der Lokführer der S-Bahn sehen kann was am Bahnsteig vor sich geht, dann wurden sie für die allgemeine Überwachung eingesetzt. Und jetzt werden sie benutzt, um herauszufinden, wer in einen Hundehaufen getreten ist, bevor er oder sie in die Bank kam. Und da soll man noch vertrauen haben, das mit all den Daten da draussen sorgsam umgegangen wird?

Posted by Joerg Moellenkamp in Privacy at 08:43

Monday, February 4, 2008

## **Burning sky**

Posted by Joerg Moellenkamp in Photographie at 21:30

## **Less known Solaris features: RBAC and Privileges**

I've wrote a quite long tutorial about RBAC and privileges. I had to divide it into 4 separate parts:  
Less known Solaris features: RBAC and Privileges - Part 1: Introduction  
Less known Solaris features: RBAC and Privileges - Part 2: Role based access control  
Less known Solaris features: RBAC and Privileges - Part 3: Privileges  
Less known Solaris features: RBAC and Privileges - Part 4: Epilogue  
PS: The topic is quite complex, thus i simplified some coherences. I hope i didn't oversimplified them ...

Posted by Joerg Moellenkamp in Solaris at 16:57

## **Less known Solaris features: RBAC and Privileges - Part 4: Epilogue**

The story of root - reprise

And then there was root again. And root was allmighty. And this time, it wasn't a bad thing. root was able to distribute the powers between the servants. But no one was as powerful as root. They had only the powers to do their job. But not more. One servant had still the problem with to many prayers, but now this servant wasn't able to destroy the world.

Even the mere mortals learned to use the power of root. Only a small group of trusted priests had the knowledge of magic chant to speak through root. All other had only the knowledge of minor chants to pray for their special wishes.

root got really cautious to give away his powers. From this day forth root casted only the absolutly needed powers to his servants and believers. And the world of root was a more secure place. It was still possible to destroy the world, but now it was much harder as there were now several guardians to protect the power of root.

The End

Interesting Links

Documentation

System Administration Guide: Security Services - Roles, Rights Profiles, and Privileges

RBAC

SMF and RBAC authorizations

Sun Whitepaper: RBAC in the Solaris Operating Environment

Custom Roles Using RBAC in the Solaris OS

Privileges

Limiting Service Privileges in the Solaris 10 Operating System

Privilege Debugging in the Solaris 10 Operating System

Posted by Joerg Moellenkamp at 16:28

## **Less known Solaris features: RBAC and Privileges - Part 3: Privileges**

Privileges

We've talked a lot about RBAC, roles, role profiles. But what are Privileges? Privileges are rights to do an operation in the kernel. This rights are enforced by the kernel. Whenever you do something within the kernel the access is controlled by the privileges. At the moment, the the rights to do something with the kernel are seperated into 70 classes:

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\_flag\_set 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\_ip\_config sys\_ipc\_config sys\_linkdir sys\_mount sys\_net\_config sys\_nfs sys\_res\_config sys\_resource sys\_smb 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

Every UNIX-System does this task hidden behind this privileges. There are many different privileges in the kernel. This privileges are not Solaris specific. It's the way to control the access to this privileges.

#### Conventional Unix

On conventional unix systems you have a root user, he has all privileges. And you have a normal user, who has only a limited set of privileges. Sometimes you need the rights of an admin to do some tasks. You don't even need to admin the system. You can only traceroute or ping a system, because both tools are setuid tools

```
$ ls -l /usr/sbin/traceroute
-r-sr-xr-x 1 root bin 42324 Nov 21 00:09 /usr/sbin/traceroute
$ ls -l /usr/sbin/ping
-r-sr-xr-x 1 root bin 51396 Nov 18 19:31 /usr/sbin/ping
```

setuid is nothing else than a violation of the security policy. You need a special privilege to ping: The privilege to use access ICMP. On conventional system this right is reserved to the root user. Thus the ping program has to be executed with the rights of root. The problem: At the time of the execution of the programm, the programm has all rights of the user. Not only to access ICMP, the programm is capable to do everything on the system, as deleting files in /etc. This may not a problem with ping or traceroute but think about larger programs. An exploit in a setuid program can lead to the escalation of the users privileges. Setuid root and you are toast.

Let's have a look at the privileges of an ordinary user. There is a tool to get the privileges of any given process in the system, it's called ppriv. \$\$ is a shortcut for the actual process id (in this case the process id of the shell):

```
bash-3.2$ ppriv -v $$
646: bash
flags =
```

```
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, (..) ,win_upgrade_sl
```

Every process in the system has four sets of privileges that determine if a process is enabled to use a privilege or not. The theory of privileges is quite complex. I would suggest to read the chapter "How Privileges Are Implemented" in the Security Services manual to learn, how each set controls or is controlled other privilege sets.

At this time, I want only to explain the meaning of the first letter:

- E: effective privileges
- I: inheritable privileges
- P: permitted privileges
- L: limit privileges

You can think about the privilege sets as keyrings. The effective privilege set are the keys the janitor has on it's keyring. The permitted privilege set are the keys the janitor is allowed to put on it's keyring. The janitor can decide to remove some of the keys. Perhaps he thinks: I work only in room 232 today. I don't need all the other keys. I leave them in my office. When he loses his keyring he lost only the control about this single room, not about the complete campus.

The inheritable privilege is not a really a keyring. The janitor thinks about his new assistant: "Good worker, but I won't give him my key for the room with the expensive tools." The limited privilege set is the overarching order from the boss of janitor to his team leaders: "You are allowed to give your assistant the keys for normal rooms, but not for the rooms with all this blinking boxes from Sun".

At the moment the most interesting set is the E:. This is the effective set of privileges. This is the set of privilege effectively available to process. Compared to the full list of privileges mentioned above the set is much smaller. But this matches your experience when you use a unix system.

#### Some practical insights to the system

You logged in as a normal user, and you have only a few privileges. It's called the basic set.

```
bash-3.2$ ppriv $$
815: bash
flags =
```

```
E: basic
I: basic
```

```
P: basic
L: allOkay, this example looks different than the one shown before. Nevertheless is has the same meaning. With
the switch -v you can expand the aliases.bash-3.2$ ppriv -v $$
815: bash
flags =
E: file_link_any,proc_exec,proc_fork,proc_info,proc_session
I: file_link_any,proc_exec,proc_fork,proc_info,proc_session
P: file_link_any,proc_exec,proc_fork,proc_info,proc_session
L: contract_event, (..) ,win_upgrade_slLooks a little bit more familiar? Okay, now let's login as root.$su root
Password:
# ppriv $$
819: sh
flags =
E: all
I: basic
P: all
L: all
```

This user has much more privileges. The effective set is much broader. The user has all privileges in the system.

How to give an user additional privileges

Now let's assume, you have an user, that wants to use dtrace. You need three privileges to use Dtrace: dtrace\_kernel,dtrace\_proc,dtrace\_user. root has this privileges. A normal user not. Giving root to a developer? God beware! This is a prelude to disaster. But no problem. Assign the matching privileges to the user, and the user is enabled to use dtrace.\$ su root

```
Password:
# usermod -K defaultpriv=basic,dtrace_kernel,dtrace_proc,dtrace_user jmoekamp
UX: usermod: jmoekamp is currently logged in, some changes may not take effect until next login.Exit to the login
prompt and login as the user you've assigned the privileges $ ppriv $$
829: -sh
flags =
E: basic,dtrace_kernel,dtrace_proc,dtrace_user
I: basic,dtrace_kernel,dtrace_proc,dtrace_user
P: basic,dtrace_kernel,dtrace_proc,dtrace_user
L: all
```

Simple ...

RBAC and privileges combined

Well, but we can do better than that. We've learned there is a thing like RBAC. There is no reason that inhibits the assignment of privileges to a role. At first we create a role bughunt, for simplicity we use the Process Management role profile. After this we set the role password.# roleadd -m -d /export/home/bughunting -P "Process Management" bughunt

```
# passwd bughunt
```

New Password: Re-enter new Password:Now we assign the privileges ...

```
# rolemod -K defaultpriv=basic,dtrace_kernel,dtrace_proc,dtrace_user bughunt... and the user to the role.# usermod -R bughunt jmoekamp
```

UX: usermod: jmoekamp is currently logged in, some changes may not take effect until next login.As you might have expected, the user itself doesn't have the privileges to use dtrace. \$ ppriv \$\$

```
883: -sh
flags =
E: basic
I: basic
P: basic
L: allBut now assume the role bughunt$ su bughunt
```

Password:

```
$ ppriv $$
```

```
893: pfsh
```

```
flags =
E: basic,dtrace_kernel,dtrace_proc,dtrace_user
I: basic,dtrace_kernel,dtrace_proc,dtrace_user
P: basic,dtrace_kernel,dtrace_proc,dtrace_user
L: allAnd DTrace is at your service.
```

### Privilege-aware programming

The idea of managing the privileges is not limited to users and their shells. In any given system you find dozens of programs as daemons.

These daemons interact in several ways with the privileges. The best way is "Privilege-aware programming". Okay. Let's assume, you code a daemon for your system. For example: You know, that you never will do an exec() call. So you can safely drop this privilege. The process modifies the permitted privilege set. The process can remove a privilege but not add it. Even when someone is able to your code, the attacker can't make an exec() call. The process doesn't even have the privilege to do such a call. And the attacker can't add the privilege again.

Several processes and programs in Solaris are already privilege aware. For example the kernel-level cryptographic framework daemon. Let's look at the privileges of the daemon. # ps -ef | grep "kcfcd"

```
daemon 125 1 0 14:24:19 ? 0:00 /usr/lib/crypto/kcfd
root 734 728 0 15:54:08 pts/1 0:00 grep kcfd
```

```
# ppriv -v 125
```

```
125: /usr/lib/crypto/kcfd
```

```
flags = PRIV_AWARE
```

```
E: file_owner,proc_prioctl,sys_devices
```

```
I: none
```

```
P: file_owner,proc_prioctl,sys_devices
```

```
L: none
```

This daemon doesn't have even the basic privileges of a regular user. It has the only the bare minimum of privileges to do it's job.

### Non-privilege aware processes

But the world isn't perfect. Not every process is privilege aware. Thus you have to limit the privileges by other mechanisms. The service management framework comes to help. The following example is copied from Glen Brunettes Blueprint Limiting Service Privileges in the Solaris 10 Operating System

Let's take the Apache Webserver as an example. The apache isn't privilege aware. We start the daemon via the Service Management Framework. # svcadm -v enable -s apache2

```
svc:/network/http:apache2 enabled. Okay, now we look at the processes of the Apache daemons. # ps -ef | grep "apache"
```

```
webservd 1123 1122 0 19:11:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start
```

```
webservd 1125 1122 0 19:11:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start
```

```
root 1122 1 1 19:11:50 ? 0:00 /usr/apache2/2.2/bin/httpd -k start
```

```
webservd 1128 1122 0 19:11:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start
```

```
webservd 1127 1122 0 19:11:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start
```

```
webservd 1126 1122 0 19:11:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start
```

```
webservd 1124 1122 0 19:11:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start
```

Six daemons running as webservd,

and one running as root.

```
# ppriv 1122
```

```
1122: /usr/apache2/2.2/bin/httpd -k start
```

```
flags =
```

```
E: all
```

```
I: basic
```

```
P: all
```

```
L: all
```

As expected for a root process, this process has the complete set of privileges of a root user. Okay, now one of it's child. # ppriv 1124

```
1124: /usr/apache2/2.2/bin/httpd -k start
```

```
flags =
```

```
E: basic
```

```
I: basic
```

```
P: basic
```

```
L: all
```

Much better ... only basic privileges.

Okay, There is a reason for this configuration. On Unix systems, you have two groups of ports. Privileged ones from 1-1023 and unprivileged ones from 1024 up. You can only bind to a privileged port with the privilege to do it. A normal user doesn't have this privilege, but root has it. And thus there has to be one process running as root. Do you remember the list of privileges for the apache process running at root. The process has all privileges but needs only one of them, that isn't part of the basic privilege set.

How to get rid of the root apache

Well, but it hasn't to be this way. With Solaris you can give any user or process the privilege to use a privileged port. You don't need the root process anymore.

Now, let's configure it this way. At first we have to deactivate the running apache. `svcadm -v disable -s apache2`  
`svc:/network/http:apache2 disabled.` I won't explain the Service Management Framework here, but you can set certain properties in SMF to control the startup of a service. `# svccfg -s apache2`  
`svc:/network/http:apache2> setprop start/user = astring: webservd`  
`svc:/network/http:apache2> setprop start/group = astring: webservd`  
`svc:/network/http:apache2> setprop start/privileges = astring: basic,!proc_session,!proc_info,!file_link_any,net_privaddr`  
`svc:/network/http:apache2> setprop start/limit_privileges = astring: :default`  
`svc:/network/http:apache2> setprop start/use_profile = boolean: false`  
`svc:/network/http:apache2> setprop start/supp_groups = astring: :default`  
`svc:/network/http:apache2> setprop start/working_directory = astring: :default`  
`svc:/network/http:apache2> setprop start/project = astring: :default`  
`svc:/network/http:apache2> setprop start/resource_pool = astring: :default`  
`svc:/network/http:apache2> end` Line 2 to 4 are the most interesting ones. Without any changes, the Apache daemon starts as root and forks away processes with the webservd user. But we want to get rid of the root user for this configuration. Thus we start the daemon directly with the webservd user. Same for the group id.

Now it gets interesting. Without this line, the kernel would deny Apache to bind to port 80. webservd is a regular user without the privilege to use a privileged port. The property start/privileges sets the privileges to start the service. At first, we give the service basic privileges. Then we add the privilege to use a privileged port. The service would start up now.

But wait, we can do more. A webserver shouldn't do any hardlinks. And it doesn't send signals outside its session. And it doesn't look at processes other than those to which it can send signals. We don't need this privileges. `proc_session`, `proc_info` and `file_link_any` are part of the basic privilege set. We remove them, by adding a ! in front of the privilege.

Okay, we have to notify the SMF of the configuration changes: `# svcadm -v refresh apache2`  
Action refresh set for `svc:/network/http:apache2`.

Until now, the apache daemon used the root privileges. Thus the ownership of files and directories were unproblematic. The daemon was able to read and write in any directory of file in the system. As we drop this privilege by using a regular user, we have to modify the ownership of some files and move some files. `# chown webservd:webservd /var/apache2/2.2/logs/access_log`  
`# chown webservd:webservd /var/apache2/2.2/logs/error_log`  
`mkdir -p -m 755 /var/apache2/run`

We need some configuration changes, too. We have to move the LockFile and the PidFile. There wasn't one of the two configuration directives in my config file, thus I've simply appended them to the end of the file. `# echo "LockFile /var/apache2/2.2/logs/accept.lock" >> /etc/apache2/2.2/httpd.conf`  
`# echo "PidFile /var/apache2/2.2/run/httpd.pid" >> /etc/apache2/2.2/httpd.conf` Okay, everything is in place. Let's give it a try.

`# svcadm -v enable -s apache2`  
`svc:/network/http:apache2 enabled.` Now we check for the running httpd processes: `# ps -ef | grep "apache2" | grep -v "grep"`  
`webservd 2239 2235 0 19:29:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start`  
`webservd 2241 2235 0 19:29:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start`  
`webservd 2235 1 1 19:29:53 ? 0:00 /usr/apache2/2.2/bin/httpd -k start`  
`webservd 2238 2235 0 19:29:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start`  
`webservd 2240 2235 0 19:29:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start`  
`webservd 2242 2235 0 19:29:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start`  
`webservd 2236 2235 0 19:29:54 ? 0:00 /usr/apache2/2.2/bin/httpd -k start` You notice the difference? There is no httpd running as root. All processes run with the userid webservd. Mission accomplished.

Let's check the privileges of the processes. At first the one, who ran as root before. `# pprv 2235`  
`2235: /usr/apache2/2.2/bin/httpd -k start`  
`flags =`  
`E: basic,!file_link_any,net_privaddr,!proc_info,!proc_session`  
`I: basic,!file_link_any,net_privaddr,!proc_info,!proc_session`

```
P: basic,!file_link_any,net_privaddr,!proc_info,!proc_session
L: allOnly the least privileges to do the job, no root privileges.
```

And even the other processes are more secure now:# ppriv 2238

```
2238: /usr/apache2/2.2/bin/httpd -k start
```

flags =

```
E: basic,!file_link_any,net_privaddr,!proc_info,!proc_session
```

```
I: basic,!file_link_any,net_privaddr,!proc_info,!proc_session
```

```
P: basic,!file_link_any,net_privaddr,!proc_info,!proc_session
```

L: allBefore we changed the configuration of the webserver, it has the basic privileges of a regular user. Now we limited even this set.

Posted by Joerg Moellenkamp in Solaris at 13:06

## Less known Solaris features: RBAC and Privileges - Part 2: Role based access control

Some basic terms

As usual the world of RBAC has some special terms. Before using it, i´m want to explain the jargon. I copy the exact definition from the RBAC manual:Rights: A right is the description, to execute an executable as an privileged user.For example the permission to execute the command reboot as root.Authorisation: A permission that enables a user or role to perform a class of actions that could affect security. For example, security policy at installation gives ordinary users the solaris.device.cdrw authorization. This authorization enables users to read and write to a CD-ROM deviceRight Profiles: A collection of administrative capabilities that can be assigned to a role or to a user. A rights profile can consist of authorizations, of commands with security attributes, and of other rights profiles. Rights profiles offer a convenient way to group security attributes.Role: A special identity for running privileged applications. The special identity can be assumed by assigned users only. In a system that is run by roles, superuser is unnecessary. Superuser capabilities are distributed to different roles.

Practical side of RBAC.

After so much theory, let´s work with roles. After installation of a Solaris system there are no rules assigned to a normal user: \$ roles

No rolesLet´s use the standard example for RBAC: reboot the system. To do this task, you need to be root.

```
$ /usr/sbin/reboot
```

reboot: permission deniedYou are not allowed to do this. Okay, until now you would give the root account to all people, who have to reboot the system. But why should someone be able to modify users, when all he or she should to is using the reboot command ?

Okay, at first you create a role. As mentioned before, it´s a special user account.# roleadd -m -d /export/home/reboot reboot

```
64 blocksAfter creating the role, you have to assign a role password.# passwd reboot
```

New Password:

Re-enter new Password:

```
passwd: password successfully changed for rebootOkay, when you look into the /etc/passwd, you see a quite normal user account. # grep reboot /etc/passwd
```

```
reboot:x:101:1::/export/home/reboot:/bin/pfshThere is one important difference. You use a special kind of shell. This shell are called profile shells and have special mechanisms to check executions against the RBAC databases.
```

Okay, we´ve created the role, now we have to assign them to a user:# usermod -R reboot jmoekamp

UX: usermod: jmoekamp is currently logged in, some changes may not take effect until next login.The RBAC system stores the role assignments in the /etc/user\_attr file# grep "jmoekamp" /etc/user\_attr

```
jmoekamp:::type=normal;roles=rebootBut at the moment, this role isn´t functional, as this role has no assigned role profile. It´s a role without rights an privileges.
```

At first, lets create a REBOOT role profile. It´s quite easy. Just a line at the end of prof\_attr. This file stores all the attributes of # echo "REBOOT:::profile to reboot:help=reboot.html" >> /etc/security/prof\_attrOkay, now assign the role profile REBOOT to the role reboot# rolemod -P REBOOT reboot The information of this assignment is stored in the /etc/usr. Let´s have a look into it:# grep reboot /etc/user\_attr

```
reboot:::type=role;profiles=REBOOT
```

```
jmoekamp:::type=normal;roles=rebootBut this isn´t enough: The profile is empty. You have to assign some administrative command to it.# echo "REBOOT:suser:cmd::/usr/sbin/reboot:euid=0" >> /etc/security/exec_attr
```

Using using the new role

Okay, let´s check the role assignments.\$ roles

```
reboot We have still no rights to execute the reboot command.$ /usr/sbin/reboot
reboot: permission deniedBut now we assume the reboot role.$ su reboot
Password: And as you see ...
$ /usr/sbin/reboot
Connection to 10.211.55.3 closed by remote host.
Connection to 10.211.55.3 closed.Connection terminates, systems reboots.
```

### Authorisations

But RBAC can do more for you. There is an additional concept in it: Authorisations.

Authorisations is a mechanism that needs support of the applications. This application checks if the user has the necessary authorisation to use a program.

Let's use the example of the janitor: Rights give him the access to the drilling machine. But this is a rather strange drilling machine. It checks, if the janitor has the permission to drill holes, when he trigger the button.

The concept of authorisation is a fine grained system. An application can check for a vast amount of privileges. For example the application can check for the autorisation to modify the configuration, to read the configuration or printing the status. A user can have all this authorisations, none or something in between.

It's like the janitors new power screwdriver. It checks if the janitor has the permission to use it at anticlockwise rotation, the permission to use it at clockwise rotation and the permission to set different speeds of rotation.

### Using authorisations for Services

Although applications need support to this modell, you can even use it as an admin. SMF has build in support for authorisations. You can assign authorisations to a service. Every role or user with this authorisation is allowed to work with the service (restarting,stop,start,status, ...). Let's use Apache for this example.

A normal user has no permission to restart the service:

```
jmoekamp$ /usr/sbin/svcadm -v disable -s apache2
svcadm: svc:/network/http:apache2: Couldn't modify "general" property group (permission denied).Wouldn't it be nice, to
have an authorisation that enables an regular user to restart it? Okay, no problem. Let's create one:$ su root
# echo "solaris.smf.manage.apache/server:::Apache Server management::" >> /etc/security/auth_attrThat's all. Where is
the definition of the permission that the authorisation measn? There is no defintion. It's the job of the application to work
with.
```

Now assign this authorisation to the user:# usermod -A solaris.smf.manage.apache/server jmoekamp

UX: usermod: jmoekamp is currently logged in, some changes may not take effect until next login.Okay, but at the moment no one checks for this authorisation, as no application is aware of it. We have to tell SMF to use this authorisation.

The authorisations for an SMF servers is part of the general properties of the service. Let's have a look at the properties of this services.

```
# svcprop -p general apache2
general/enabled boolean false
general/entity_stability astring EvolvingNo authorisation configured. Okay ... let's add the authorisation we've defined
before:svccfg -s apache2 setprop general/action_authorization=astring: 'solaris.smf.manage.apache/server'Check the
properties again:# svcadm refresh apache2
# svcprop -p general apache2
general/enabled boolean false
general/action_authorization astring solaris.smf.manage.apache/server
general/entity_stability astring EvolvingOkay, a short test. Exit your root shell and login as the regular user you have
assigned the authorisation.bash-3.2$ svcs apache2
STATE      STIME     FMRI
disabled   22:49:51  svc:/network/http:apache2Okay, i can view the status of the service. Now i try to start it.bash-3.2$
/usr/sbin/svcadm enable apache2
svcadm: svc:/network/http:apache2: Permission denied.What the hell ...? No permission to start the service? Yes,
enabling the service is not only a method (the start up script), it's a value of a certain parameter. When you only have
the action_authorization you can only do task, that doesn't change the state of the service. You can restart it (no change
of the service properties), but not enable or disable it (a change of the service properties). But this is not a problem. You
have to login as root again and assign the the solaris.smf.manage.apache/server authorisation to the value
```

```
authorisation.# svccfg -s apache2 setprop general/value_authorization=astring: 'solaris.smf.manage.apache/server'With
the value authorisation SMF allows you to change the state of the service. Try it again.bash-3.2$ /usr/sbin/svccadm
enable apache2
bash-3.2$Cool, isn't it ... try this with init.d ...
```

### Predefined roles

This was a really simple example. Roles can get really complex. But you don't have to define all role profiles at your own. For some standard tasks, there are some predefined roles. Just look at the `/etc/security/prof_attr`. There are 70 role profiles defined in this file. For example the right profile "Software Installation"Software Installation:::Add application software to the system:help=RtSoftwareInstall.html; auths=solaris.admin.prodreg.read, solaris.admin.prodreg.modify, solaris.admin.prodreg.delete,solaris.admin.dcmgr.admin, solaris.admin.dcmgr.read,solaris.admin.patchmgr.\*This role profile has already some predefined command, that need special security attributes to succeed:

```
Software Installation:solaris:act:::Open;*:JAVA_BYTE_CODE;*:uid=0;gid=2
```

```
Software Installation:suser:cmd:::/usr/bin/ln:euid=0
```

```
Software Installation:suser:cmd:::/usr/bin/pkginfo:uid=0
```

```
Software Installation:suser:cmd:::/usr/bin/pkgmk:uid=0
```

```
Software Installation:suser:cmd:::/usr/bin/pkgparam:uid=0
```

```
Software Installation:suser:cmd:::/usr/bin/pkgproto:uid=0
```

```
Software Installation:suser:cmd:::/usr/bin/pkgtrans:uid=0
```

```
Software Installation:suser:cmd:::/usr/bin/prodreg:uid=0
```

```
Software Installation:suser:cmd:::/usr/ccs/bin/make:euid=0
```

```
Software Installation:suser:cmd:::/usr/sbin/install:euid=0
```

```
Software Installation:suser:cmd:::/usr/sbin/patchadd:uid=0
```

```
Software Installation:suser:cmd:::/usr/sbin/patchrm:uid=0
```

```
Software Installation:suser:cmd:::/usr/sbin/pkgadd:uid=0;gid=bin
```

```
Software Installation:suser:cmd:::/usr/sbin/pkgask:uid=0
```

```
Software Installation:suser:cmd:::/usr/sbin/pkgchk:uid=0
```

```
Software Installation:suser:cmd:::/usr/sbin/pkgrm:uid=0;gid=bin
```

This is all you need to install software on your system. You can use this predefined role profiles at your will. You don't have to do define all this stuff on your own.

Posted by Joerg Moellenkamp in Solaris at 12:32

## Less known Solaris features: RBAC and Privileges - Part 1: Introduction

### The Story of root

And then there was root. And root was allmighty. And that wasn't a good thing. root was able to control the world without any control. And root needed control. It was only a short chant between the mere mortals and root. Everybody with the knowledge of the magic chant was able to speak through root.

But root wasn't alone. root had servants called daemons. Some of one them needed divine powers to do their daily job. But root was an undividable being. So the servants had to work with the powers of root. But the servants wasn't as perfect as root: Some of the servants started to do everything mere mortals said to them if they only said more than a certain amount of prayers at once.

One day, the world of root experienced a large disaster, the negation of beeing. Top became bottom, left became right, the monster of erem-ef anihilated much of the world. But it got even stranger. root destroyed it's own world, and by the power of root the destruction was complete.

Then there was a FLASH. The world restarted, root got a second attempt to reign his world. But this time, it would be different world.

### Superuser

The old model of rights in a unix systems is based on a duality. There is the superuser and the normal user. The normal users have a restricted set of rights in the system, the superuser has an unrestricted set of rights. To modify the system, a normal user has to login as root directly or assume the rights of root (by `su -`). But such a user has unrestricted access to system. Often this isn't desirable. Why should you enable an operator to modify a system, when all he or she has do to on the system is creating some users from time to time. You've trained him to do `useradd` or `passwd`, but it's a Windows admin who doesn't know anything about beeing an Unix admin. What do you do when he gets to curious. He needs root privileges to create a user or change a password. You need some mechanisms to limit this operator.

But it's get more problematic. Programs have to modify the system to work. A webserver is a nice example. It uses port 80. Ports beneath port number 1024 have a special meaning. They are privileged ports. You need special rights to modify the structures of the system to listen to the port 80. A normal user doesn't have this rights. So the webserver has to be started as root. The children of this process drop the rights of root by running with a normal user. But there is this single instance of the program with all the rights of the user. This process has much rights than needed, a possible attack vector for malicious users.

This led to the development to differet models of handling the rights of users in the system: Privileges and Role Based Access Control.

### Least privileges

There is a concept in security. It's called least privileges. You give someone only least amount of privileges, only enough to do it's tasks. An example of the real world. You won't give the janitor the master key for all the rooms on the campus, when all he has to do is working in Building C. The other way round: There are some trusted people who have access to all rooms in case of emergency.

You have the same concept in computer security. Everyone should have only the least amount of privileges in the system to do it's job. The concept of the superuser doesn't match to this. It's an all or nothing. You are an ordinary user with basic privileges or you are an user with unrestricted rights. There is nothing in between. There is no least privileges in this concept.

### Role Based Access Control

The example with the key for the janitors is a good example. Let's imagine a large campus. You have janitors responsible for the plumbing (let's call them Lenny and Carl), for the park (let's call him Homer), for the security system (let's call Marge, Lenny helps from time to time).

These roles have different sets of privileges: For example the plumbing janitor has access to all rooms of the heating system. The janitor for the park has only access to the garage with the lawnmover and the cafeteria and the janitor for the security system.

When they start to work in their job, they assume a role. From the privilege perspective it's not important who is the person, but what role the person has assumed. Lenny punches the clock and assumes the role of the plumbing janitor for the next 8 hours. And while he is doing it's job he uses the privileges inherent to the role. But he has to do tasks in his office or in his workshop. It's his own room, so he doesn't need the privileges. He doesn't need the special privileges.

Role Based Access Control is quite similar. You login to the system, and then you start work. You read your emails (no special privileges needed), you find an email "Create user xy45345. Your Boss". Okay, now you need special privileges. You assume the role of an User Administrator create the user. Job done, you don't need the privileges anymore. You leave the role and write the "Job done" mail to your boss with your normal users.

Role Based Access Control is all about this: Defining roles, giving them privileges and assigning users to this roles.

### Privileges

I've used the word quite often in the article so far. What is a privilege. A privilege is the right to do something. For example, having the keys for the control panel of the heating system.

Unix users are nothing different. Every user has privileges in a unix system. A normal user has the privilege to open, close, read write and delete files when he his allowed to do this (Because he created it, because he belongs to the same group as the create of the file or the creator gave everybody the right to do it). This looks normal to you, but it's privilege based on the login credentials you gave to system. You don't have the privilege to read all files on the system or to use a port number 1024.

Every thing done in the system is based on this privileges. Solaris seperated the tasks into many privilege sets. At the moment, there are 70 different privileges in the system. The difference between the normal user is that the users has only a basic set, the root has all.

But it hasn't to be this way. Privileges and users aren't connected with each other. You can give any user the power of the root user, and restrict the privileges of the root user. It's just our binary compatibility guarantee that mandates that the standard configuration of the system resembles the superuser model. There are application out there, which assume that only the root user or the uid 0 as unrestricted rights and exit when they are started with a different user.

#### RBAC and Privileges in Solaris

Both features have their root in the Trusted Solaris development. Trusted Solaris was a version of Solaris to ensure highest security standards. Today, these mechanisms are part of the normal Solaris in conjunction with the Trusted Extensions. So RBAC is a really old feature: It's in Solaris since version 8 (published in 2000). Privileges found their way into the generic Solaris with the first availability of Solaris 10 in February 2005.

Posted by Joerg Moellenkamp in Solaris at 09:55

Saturday, February 2, 2008

### **Less known Solaris features: BART**

Apropos auditing. There is a small but cool tool in Solaris. It solves the problem of "No, i haven't changed anything on the system". It's called BART, the Basic Audit Reporting Tool. It a really simple tool and it's really easy to use.

Usage

Okay, let's assume after some days of work you finally configured all components of your new system. Okay, create a nice place to store the output of the bart tool. After this you start bart for the first time to create the first manifest of your system.  
# mkdir /bart-files

# bart create -R /etc > /bart-files/etc.control.manifest  
The manifest stores all informations about the files. This is the example for the /etc/nsswitch.nisplus:  
# cat etc.control.manifest | grep "/nsswitch.nisplus"

/nsswitch.nisplus F 2525 100644 user::rw-,group::r--,mask:r--,other:r-- 473976b5 0 3

79e8fd689a5221d1cd059e5077da71b8  
Now lets change some files:  
# touch /etc/thisisjustatest

# chmod 777 /etc/nsswitch.files

# echo "#just a test" >> /etc/nsswitch.nisplus  
Okay, enough changes. Let's create a new manifest of the changed /etc.

Pipe it to a different file.  
# bart create -R /etc > /bart-files/etc.check20080202.manifest  
Now we can compare the baseline manifest with the actual manifest.  
# cd /bart-files

# bart compare etc.control.manifest etc.check20080202.manifest

This command prints all differences between the two manifests and thus the difference between the tow states of the system/nsswitch.files:

mode control:100644 test:100777

acl control:user::rw-,group::r--,mask:r--,other:r-- test:user::rwx,group::rwx,mask:rwx,other:rwx

/nsswitch.nisplus:

size control:2525 test:2538

mtime control:473976b5 test:47a44862

contents control:79e8fd689a5221d1cd059e5077da71b8 test:3f79176ec352441db11ec8a3d02ef67c

/thisisjustatest:

addAs i wrote before: A really nice tool.

Want to learn more?

For more information about this tool visit [Using the Basic Audit Reporting Tool](#).

Posted by Joerg Moellenkamp at 17:17

### **Less known Solaris features: Auditing**

One of the less known features in Solaris is the Auditing. Auditing solves an important problem: What happens on my system, and whodunnit. When something strange happens on your system or you recognize, that you are not the only one who owns your system, it's a good thing to have some logs for analysis.

The nice thing about the auditing in Solaris: It's quite simple to activate. In the this article i will give you a short overview to enable and use the auditing in Solaris. This feature is really old, it's in Solaris for since the last century but nevertheless it's a less known Solaris feature.

Some terms

There are some special terms in auditing. I want to give you a short definition of them as i have to use them in this article. I've copied this definitions from the manual for Solaris Auditing.

Audit events: A security-related system action that is audited. For ease of selection, events are grouped into audit classes.

Audit Class: A grouping of audit events. Audit classes provide a way to select a group of events to be audited.

Audit policy: A set of auditing options that you can enable or disable at your site. These options include whether to record certain kinds of audit data. The options also include whether to suspend auditable actions when the audit trail is full.

Configuring basic auditing

You have to search for a place in your filesystem. It's a good practice to use an own filesystem, as auditing will eat away your filesystem space until there is nothing left and this is a bad idea for the root file system. But in this example i will omit this step.

At first login as root. Okay, you need a place to store the audit logs. It's important to change the rights of the directory to

```
assure only root can access it.mkdir /var/audit/aragorn-sol
chmod -R 750 /var/audit/aragorn-solThen go to to /etc/security and edit the file /etc/security/audit_control. This file
controls where what classes of information are logged and where you write the log. For example: The lo is the audit
class for all events in regard of logins and logoffs:dir:/var/audit/aragorn-sol
flags:lo
minfree:20
naflags:loOkay, configuration is done.But let's have another look the file /etc/security/audit_startup. The commands in
this script control the audit policies and thus the behaviour of the logging and the amount of informations in the log
records:/usr/bin/echo "Starting BSM services."
/usr/sbin/auditconfig -setpolicy +cnt
/usr/sbin/auditconfig -conf
/usr/sbin/auditconfig -aconfThe second line is the most interesting. Without this line the system would stop user
interaction when the system is unable to log. You would deactivate this behaviour, when logging is more important than
system availability. For the moment we don't change this file.
```

### Start the auditing

```
Now activate auditing. You have to reboot after the activation.# ./bsmconv
This script is used to enable the Basic Security Module (BSM).
Shall we continue with the conversion now? [y/n] y
bsmconv: INFO: checking startup file.
bsmconv: INFO: turning on audit module.
bsmconv: INFO: initializing device allocation.
```

The Basic Security Module is ready.

If there were any errors, please fix them now.

Configure BSM by editing files located in /etc/security.

Reboot this system now to come up with BSM enabled.

```
# rebootTwo short checks ... auditd runs ...# svcs | grep "auditd"
```

```
online      23:30:03 svc:/system/auditd:default... and the system starts to gather audit logs
```

```
# ls -la
```

```
total 6
```

```
drwxr-x--- 2 root  root    512 Feb  1 23:30 .
```

```
drwxr-xr-x 3 root  sys     512 Feb  1 23:18 ..
```

```
-rw-r----- 1 root  root    255 Feb  1 23:33 20080201223003.not_terminated.aragorn-sol
```

Okay, now you have completed the configuration. The system has started to write audit logs.

### Managing the audit logs

Audit logs grows infinitely. To the maximum filesize in the used filesystem or the end of disk capacity ... whatever occurs first. It's a good practice to checkpoint the audit logs in a regular interval. It's quite simple:audit -nWith this command the actual file gets closed and a new one gets opened. # cd /var/audit/aragorn-sol/

```
# ls -l
```

```
total 24
```

```
-rw-r----- 1 root  root    684 Feb  1 23:55 20080201223003.20080201225549.aragorn-sol
```

```
-rw-r----- 1 root  root    571 Feb  2 00:06 20080201225549.20080201230639.aragorn-sol
```

```
-rw-r----- 1 root  root   2279 Feb  2 00:10 20080201230834.20080201231010.aragorn-sol
```

```
-rw-r----- 1 root  root    755 Feb  2 00:12 20080201231010.20080201231245.aragorn-sol
```

```
-rw-r----- 1 root  root   4274 Feb  2 08:36 20080201231245.20080202073624.aragorn-sol
```

```
-rw-r----- 1 root  root    200 Feb  2 08:36 20080202073624.not_terminated.aragorn-sol
```

### Analysing the audit trails

It doesn't make sense to create audit logs without looking at them. You can't look directly at them as this file are binary ones. You need to command to analyse the audit log. One to extract the data out of the log files based on certain rules and one command to translate it into an human readable format.

You use the auditreduce command for the first step, and the praudit command for the second one.# cd

```
/var/audit/aragorn-sol
```

```
auditreduce * | praudit -sThis sequence of commands translate all you audit logs into an human readable form. I've cut
```

```
out some of the lines for an example:header,69,2,AUE_ssh,,localhost,2008-02-01 23:49:17.687 +01:00
```

```
subject,jmoekamp,jmoekamp,other,jmoekamp,other,720,3447782834,6969 5632 10.211.55.2
```

```
return,success,0
```

```
header,77,2,AUE_su,,localhost,2008-02-01 23:49:55.336 +01:00
```

```
subject,jmoekamp,root,other,jmoekamp,other,729,3447782834,6969 5632 10.211.55.2
```

```
text,root
return,failure,Authentication failed
header,69,2,AUE_su,,localhost,2008-02-01 23:50:11.311 +01:00
subject,jmoekamp,root,root,root,root,730,3447782834,6969 5632 10.211.55.2
return,success,0What tells this snippet to you: I've logged into my system as the user jmoekamp, tried to assume root
privileges, failed the first time (due wrong password), tried it again and succeeded.
```

#### More auditing

Sometimes it's important to know what users have done on you system. For example: Which programs has been executed. With Solaris auditing it's really easy to collect this information.

At first you have to configure auditing to collect this kind of information:dir:/var/audit/aragorn-sol

```
flags:lo,ex
```

```
minfree:20
```

naflags:lo,exThe ex audit class matches to all events in system in regard to the execution of a program. This tells the auditing subsystem to log all execve() system calls. But you have to signal this change to the audit subsystem to start the auditing of this events. With audit -s you notify the audit daemon to read the /etc/security/audit\_control file again.

```
header,113,2,AUE_EXECVE,,localhost,2008-02-02 00:10:00.623 +01:00
```

```
path,/usr/bin/l
```

```
attribute,100555,root,bin,26738688,1380,0
```

```
subject,jmoekamp,root,root,root,root,652,2040289354,12921 71168 10.211.55.2
```

return,success,0But this configuration only logs the path of the command, not the command line parameters. You have to configure to log this information. You remember: The audit policy controls the kind of information in the audit logs.

Thus we have to modify the audit policy. With the command auditconfig -setpolicy +argv you change the policy. You don't have to activate it, it's immediately effective:header,124,2,AUE\_EXECVE,,localhost,2008-02-02 00:12:49.560

```
+01:00
```

```
path,/usr/bin/l
```

```
attribute,100555,root,bin,26738688,1380,0
```

```
exec_args,2,ls,-l
```

```
subject,jmoekamp,root,root,root,root,665,2040289354,12921 71168 10.211.55.2
```

```
return,success,0
```

To make this behaviour persistent, you have add the auditconfig -setpolicy +argv to the file

Want to learn more?

This is only a really short introduction to the topic. You will find the documentation for this feature at docs.sun.com: Part VII Solaris Auditing of System Administration Guide: Security Services is a good place to start.

Posted by Joerg Moellenkamp at 11:21

Friday, February 1. 2008

### **More fun with the TSA**

There is a new TSA regulation: You have to take out all electronic devices including cabling out of your carry-on luggage. I'm in New York at the end of this month. I really hope that this regulation is only a valid at SFO. Otherwise travelling to the US will be a major pain in the a... with all my electronic gadgets ....

Posted by Joerg Moellenkamp in Business Travel at 14:24

### **links for 2008-02-01**

Interview: MySQL wird anders aussehen - CIO & Karriere - silicon.de  
(tags: mysql sun)

Sun xVM - Foundation for a dynamic datacenter? - Virtually Speaking - ZDNet.com  
(tags: sun virtualization)

Posted by del.icio.us in del.icio.us at 12:25

### **Alpha version of the second Indiana preview**

Stephen Hahn announced in this mail a test version of the second OpenSolaris Developer Preview a.k.a Indiana. Give it a try and report every bug you find.

Posted by Joerg Moellenkamp in Solaris at 10:57

### **Burchardkai**

Posted by Joerg Moellenkamp in Photographie at 06:31