Step by Step Guide to Building a Secure Solaris 8 KDE Graphical Workstation

Version 1.7

Michael Harris

Introduction

This paper will show a security conscious end user how to install standalone Solaris 8 KDE graphical X workstation on a sun ultra 10 workstation using Solaris 8 version 7/01. Most security texts are geared toward a minimal 'server' approach. I hope to address the minimal 'workstation' approach with this text. I am aware that one can use the Solaris installation media to configure a CDE workstation using the end-user software cluster; but, unfortunately, this loads many software packages that are unnecessary and leaves open many security holes and risks. By following the examples given in this text, you will have a reasonably secure KDE based X windows workstation to do your work.

These are the necessary software packages that will be needed during installation:

- Solaris 8 software CD 1 & 2 and companion CD
- BOLTget.pkg file from www.sunfreeware.com
- FixModes.tar.Z from www.sun.com/blueprints/tools/FixModes.tar.Z
- nddconfig.tar from www.sun.com/blueprints/tools/nddconfig.tar
- 8_Recommended.zip from http://sunsolve.sun.com/pub/patches

1 System Identification Phase

In this section we will identify the system and configure the network parameters for our host. We will need such things as the hostname and IP address of the system.

Use the Stop-A sequence to stop the normal boot process and boot from the CDROM using the command:

```
> boot cdrom
```

Answer the dialogs boxes which will pop-up during the installation with the necessary specific values for your host for proper system identification:

Select a Language	0 – English
Select a Locale	0 - English (C – 7-Bit ASCII)

Host name	Name of host
IP address	IP address of host
Part of a subnet	Yes
Netmask	Subnet mask of host
Enable IPV6	No

Configure Kerberos Security	No

Name service	None
Specify timezone by	Geographic region
Time zone	US/Eastern
Date/Time	Correct Date/Time
Initial/Upgrade Software	Initial

2 Software Interactive Selection Phase

In this phase, we will identify to the installation software which software cluster and specific software packages we want to be installed. For our purposes of security, we will select the CORE software cluster because it offers the minimum software packages for OS functionality. After this selection, we will add on additional packages. This minimal software installation method is done to maximize security by not allowing any unnecessary software and its associated files to be left dormant on the computer system to be exploited by potential hackers.

Software Installation	Initial
Geographic Regions	{Click Continue}

Select the Core System Support option and then click the customize button and add these additional packages for X Window System support:

Software Group	Core System Support
Customize	**see below**

- M64 Graphics Accelerator Support
  - M64 Graphics Configuration Software
  - M64 Graphics Window System Support
- Open Windows version 3
  - ICE Components
  - X Window System Platform Software
  - Nonessential MIT Core Clients and Server Extensions
- Tool Talk End User
  - Tool Talk Runtime
- XIL Runtime Environment
- X Window System & Graphics Runtime Library Links in /usr/lib
- X Window System ICE library (64-bit)
- X Window System library software (64 bit)
- X Window System Runtime Compatibility Package (64 bit)
- Font Server Cluster
- SunOS Header Files
- VIS/XIL Support

After this, select the boot disk and then choose manual layout.

3 Layout the Filesystem

In this section, we will layout our filesystems on the selected boot disk. Now, there are two schools of thought on the matter: put everything on one large root filesystem or divide up the OS installation onto different filesystems. We will choose to divide up our disks because of the following reason: Doing so will give us the opportunity to make certain decisions at filesystem mount time about the actual intended access pattern of the filesystem – whether the files on the filesystem will be read-only or not to allow suid programs to be run, etc. In addition, you also get the added protection of not having one big root filesystem filling up because of excessive logging or malicious activity. I will describe the layout of the filesystem in this section and later on we will define what access patterns are to be allowed (read-only,nosuid) on these filesystems.

Layout the filesystem using the GUI interface so that there is a small root /; a /usr partition; (to be mounted read-only for system binaries); a /export/home for home directories; a /var partition; (for log and downloaded files) and a /usr/local & /opt for value added software.

The filesystem layout for an 18GB internal disk is as follows:

Slice c0t0d0s0	/	129
Slice c0t0d0s1	/usr/local	11042
Slice c0t0d0s2	Overlap	17345
Slice c0t0d0s3	/var	513
Slice c0t0d0s4	Swap	149
Slice c0t0d0s5	/opt	2048
Slice c0t0d0s6	/usr	385
Slice c0t0d0s7	/export/home	3076

Then, just click "CONTINUE" at the remote mount filesystems dialog box because we are setting up a standalone machine which will not need the resources of remote systems:

Remote mount filesystems	{Press Continue}
Auto-reboot?	YES

Click install and the software will then be copied to the system disk you selected and will be rebooted.

4 Set Root's Environment

After the system is restarted, log in as the root user, and ***PLEASE*** give root a password via the *passwd* command. Choose a password that is cryptic, not in a standard dictionary, at least 6 chars in length and contains one or more punctuation marks/symbols. This is done to make dictionary and brute force password guessing routines crackers often use to crack password files take a very long time to complete.

We will now set up a private root home directory where only the root user has access to the contents of this directory. We will also change the file creation modes of the root user so that only root can read/write any file created by root. In addition, we will create another account for your daily activities since we will only use root for necessary system administration tasks and use our 'daily' account for normal tasks. In addition, we will set root's PATH variable to ***NOT*** include the current directory and to have only root owned directories included because of the potential of running Trojan horse programs left by crackers in whatever current directory you are in. Please fill in appropriate values for your machine:

```
Console login: root
Sun Microsystems Inc.  SunOS 5.8    Generic February 2000
# passwd
# New Password: *****
# Re-enter new password: *****
# passwd (SYSTEM): password successfully changed for root
# cat << end > /.profile
> PATH=/usr/local/bin:/opt/sfw/bin:/usr/openwin/bin:$PATH
> VISUAL=vi
> export PATH VISUAL
> umask 077
> end
# chmod 600 /.profile
# . /.profile
# mkdir /root
# chown root:root /root
# passmgmt –m –h /root root
# mv /.profile /root
# useradd -u <uid> -g <gid> -d /export/home/(user) –m  –s <default shell> <loginid>
# passwd <loginid>
```

5 Configuring Network Support

In this section, we will add networking support. At this stage, one would plug the Ethernet cable back into the host. I am aware that this is somewhat of a security risk. But, I feel that this can be tolerated because it is a workstation and not a server. If you feel that this is a big security risk, then I suggest that you review this section and download these files on another secure computer and burn a CD with the downloaded software for installation on this host and install the software via the CD or use some other method of getting these files to the workstation.

So, we will now add networking support. We will download the latest patches and other security software. To accomplish this, we need to configure /etc/resolv.conf with our nameserver for our LAN; /etc/defaultrouter with our network's default router; create an /etc/notrouter file to prevent our system from routing network packets (since our system is a workstation, not a router); create an /etc/nologin file to prevent people from logging in over the network to our computer and lastly change /etc/nsswitch.conf to allow host name resolution to come first from the /etc/inet/hosts file and if the hostname is not there then do a DNS query. We will also disable the inetd server from running by removing its configuration file. This will prevent our workstation from answering TCP/IP requests controlled by this daemon. Please substitute appropriate nameserver and ip address values for your site:

```
# cd /etc
# echo 'default your.domainname.com
> nameserver 1.2.3.4' > resolv.conf
# touch notrouter nologin
# echo '1.2.3.4' > defaultrouter
# cp nsswitch.dns nsswitch.conf
# chmod 644 nsswitch.conf resolv.conf defaultrouter
# rm inetd.conf /etc/inet/inetd
```

Reboot the system!

6 Additional Package Administration

Now, we are ready to ***REMOVE*** packages which are not needed. Answer Y to the package removal questions. Here is a list of packages that were not needed for a basic Ultra 10 workstation mainly because the function of this unit will be a workstation and not a server and most packages are not needed for an Ultra 10:

SUNWatasr	AutoFS, (Root)
SUNWatfsu	AutoFS, (Usr)
SUNWeridx	Sun RIO 10./100 Mb Ethernet Drivers (64-bit)
SUNWfcip	Sun FCIP IP/ARP over Fibrechannel Device Driver
SUNWfcipx	Sun FCIP IP/ARP over FibreChannel Device Driver (64 bit)
SUNWfcp	Sun FCP SCSI Fibre Channel
SUNWfcpx	Sun FCP SCSI Fibre Channel 64 Bit
SUNWfctl	Sun Fibre Channel Transport layer
SUNWfctlx	Sun Fibre Channel Transport layer (64-bit)
SUNWftpr	Sun FTP Server root
SUNWftpu	Sun FTP Server usr
SUNWged	Gigabit Ethernet Driver
SUNWluxop	Sun Enterprise Network Array firmware and utilities
SUNWluxox	Sun Enterprise Network Array libraries (64-bit)
SUNWmdi	Sun Multipath I//O Drivers
SUNWmdix	Sun Multipath I/O Drivers (64-bit)
SUNWnisr	Network Information System (Root)
SUNWnisu	Network Information System (Usr)
SUNWpcelx	3COM Etherlink III PCMCIA Ethernet Driver
SUNWpcmci	PCMCIA Card Services (Root)
SUNWpcmcu	PCMCIA Card Services (Usr)
SUNWpcmcx	PCMCIA Card Services (64-bit)
SUNWpcmem	PCMCIA memory card driver
SUNWpcser	PCMCIA serial card driver
SUNWpsdpr	PCMCIA ATA Driver
SUNWqfed	Sun Quad FastEthernet Adapter Driver

SUNWqfedx	Sun Quad FastEthernet Adapter Driver (64-bit)
SUNWses	SCSI enclosure Services Device Driver
SUNWsesx	SCSI enclosure Services Device Driver (64 Bit)
SUNWssad	SPARCstorage Array Drivers
SUNWssadx	SPARCstorage Array Drivers (64-bit)
SUNWusb	USB device drivers
SUNWusbx	USB device drivers (64 Bit)

List of removed packages

```
# cd /var/sadm/pkg
# pkgrm *fc* *at* *ss* *lu* *pc* *mdi* *us* *qf* *se* *ni* *ps* *eri* *ge*
```

In this section, we will add some useful secondary packages from the Sun Solaris CD 2 of 2. These packages will add some additional functionality to the workstation.

Mount the Second CD of the Solaris CD set and add the following packages: *Accounting, Terminal Information, and Gzip & Zlib*:

```
# mount –F hsfs /dev/dsk/c0t5d0s0 /mnt
# cd /mnt/Sol*/Pro*
# pkgadd –d .*ter* *acc* *gzip *zlib *zlibx
```

Now, mount the Solaris Companion CD into the cdrom and install KDE and it's necessary co-packages as follows:

```
# cd /
# umount /mnt
enter solaris companion cd into cdrom drive
# mount –F hsfs /dev/dsk/c0t5d0s0 /mnt
# cd /mnt/components/sparc/Packages
# pkgadd –d . SFWqt SFWxpm SFWjpg SFWpng SFWtiff SFWgcmn SFWkde
```

7 Add the Pkg-Get Package

In this section, we will obtain various security packages from various sources on the internet. We will use the combination of the wget and pkg-get utilities to obtain these software packages. This is done so as to make the retrieval and installation of the software packages command line driven - which can lend itself to scripting. In addition, when using the pkg-get utility, if there is a software package that has new versions and you just supply the name of the package as the only argument to the utility, the utility will return all the versions available of that particular package. This is very good if you want to install the latest releases of software package because you can simply choose the latest release from the returned list of packages. In this section, I will assume that your access to the internet is sound and you can ping internet hosts and do DNS lookups without problems.

Now add the *wget* for use with the *BOLTpget* package. We will then use the wget utility to retrieve the BOLTpget utility. This package-get utility allows one to download sun compiled third party software and have it installed in one step. So, let's install the *BOLTget.pkg* file to your system for quick download access to www.sunfreeware.com packages and update its package versioning database:

```
# pkgadd –d . SFWwget
# cd /var/tmp
# /opt/sfw/bin/wget http://www.bolthole.com/solaris/BOLTpget.pkg
# pkgadd -d ./BOLTpget.pkg
# /usr/bin/pkg-get update
# umount /mnt
```

8 Obtain Additional Software Security Packages.

Let's use the *pkg-get* utility to install these essential security packages: openssh (the secure shell), openssl (the secure socket layer), wget (the sunfreeware version of wget) and libgcc. Just follow the instructions on the screen to get and install the latest versions of these packages. (If the command returns "Sorry, multiple versions exist, then just rerun the command with the latest stable release version of the particular software in question"). After the packages install successfully then we can remove the original wget package (SFWwget) since we now have the sunfreeware version of that utility (SMCwget).

```
# /usr/bin/pkg-get install openssl openssh wget libgcc
# pkgrm SFWwget
```

Note: If multiple versions exist, just rerun the command and install the latest stable release from the reply message

9 Obtaining Patches

This is a very important step. We will connect to http://sunsolve.sun.com site and obtain the latest patch cluster. Keeping up to date with the latest patches for your operating environment is one of the most important security administration steps you can do to maintain the security of your system. This is so because by applying patches to your system, you will have plugged the holes with the patches that other unpatched systems will have wide open.

Get the latest patch cluster from Sun and install:

```
# wget ftp://sunsolve.sun.com/pub/patches/8_Recommended.zip
# unzip -qq 8_Recommended.zip
# cd 8_Recommended
# ./install_cluster -q –nosave
# cd ..
# rm -r 8_Recommended*
```

Note: Return code 2 -> Patch already installed Return code 8 -> Patch of package that is not installed

In this step, we will tighten our tcp/ip network communication parameters using a script from sun blueprints online. This is very important to avoid common tcp/ip related network attacks such as SYN flood, etc. The script does this by setting various network parameters via the ndd command – see the man page on ndd for the particulars of each ndd setting which will be echoed on the console screen upon reboots.

Let's also tighten our network communication parameters:

```
# wget www.sun.com/blueprints/tools/nddconfig.tar
# tar xf nddconfig.tar
# chmod 744 nddconfig
# chown root:sys nddconfig
# mv nddconfig /etc/init.d
# ln /etc/init.d/nddconfig /etc/rc2.d/S70nddconfig
```

Here we will restrict the default permissions of certain directory and files. The default install of Sun Solaris 8 leaves many directories/files open for public viewing and modification. The Fixmodes script will take care of these issues by removing permissions and setting more secure permissions on certain important files and directories.

Let’s fix our directory and file permissions:

```
# wget www.sun.com/blueprints/tools/FixModes.tar.Z
# uncompress FixModes.tar.Z
# tar xf FixModes.tar
# FixModes/fix-modes
```

Enable eeprom security passwords on your workstation to prevent any user from executing openboot commands without password authorization:

```
# eeprom security-password =
Changing PROM password:
New Password: {obp_password}
Retype new password:
# eeprom security-mode=command
```

We will next tighten the kernel from the execution of programs from the ‘user-stack’. This is a well known exploit in which an attacker feeds an input string into a program that does no ‘bounds-checking’ and writes machine level code into the string to jump to other parts of the string to do malicious things. This is a very subtle attack that is allowed to occur usually because of programmer error and neglect. We will also set the maximum number of user processes that can be run by any user to 128, set the core dump size to 0 for no core dumps, and set the system up so that NFS related services will only be allowed from privileged ports.

Tighten kernel tunables via */etc/system*:

```
# echo “set noexec_user_stack=1” >> /etc/system
# echo “set noexec_user_stack_log=1” >> /etc/system
# echo “set maxuprc=128” >> /etc/system
# echo “set sys:coredumpsize=0” >> /etc/system
# echo “set nfssrv:nfs_portmon=1” >> /etc/system
```

The last step is to stop the booting of services that are security risks and we don’t need such as NFS and RPC by preventing these startup scripts from executing and clean up /var/tmp.

S30sysid.net	Basic network parameters
S71ldap.client	Ldap client
S71rpc	Remote Procedure Call
S71sysid.sys	Setup script for system attributes
S72autoinstall	Jumpstart autoinstall
S73cachefs.daemon	Cachefs filesystem
S73nfs.client	Nfs client
S76nscd	Name service cache daemon
S80PRESERVE	Script to move salvaged edited files /usr/preserve
S88sendmail	Sendmail daemon
S93cacheos.finish	Cachefs finish script

Removed startup scripts

```
# cd /etc/rc2.d
# for file in S30* S71* S72auto* S73* S76* S80* S88send* S93*
> do
> mv $file .NO$file
> done
# rm –r /var/tmp/*
```

Reboot system for changes to take effect!

10 File Configuration

We are now ready to do some file editing/configuration.

First, we will update the mount table to make the accessing of files on filesystems more secure as promised when we configured the filesystems at install time. Edit the */etc/vfstab* file and change these lines so that some filesystems are *ro* for read only access and *nosuid* to allow no set-uid programs from running on that partition. The inclusion of the logging parameter on each filesystem ensures a quicker recovery time from a system crash and reduction of filesystem corruption because all filesystem writes are first written to a log and then written out to disk. If anything happens, the filesystem just has to look at its logs and make the appropriate changes to the filesystem to bring it back to the 'clean' state without the lengthy fsck pass.

```
/dev/dsk/c0t0d0s0  /dev/rdsk/c0t0d0s0   /             ufs   1   no    remount,logging
/dev/dsk/c0t0d0s6  /dev/rdsk/c0t0d0s6   /usr          ufs   1   no    ro
/dev/dsk/c0t0d03s  /dev/rdsk/c0t0d0s3   /var          ufs   1   no    nosuid,logging
/dev/dsk/c0t0d0s7  /dev/rdsk/c0t0d0s7   /export/home  ufs   2   yes   nosuid,logging
/dev/dsk/c0t0d0s5  /dev/rdsk/c0t0d0s5   /opt          ufs   2   yes   logging
/dev/dsk/c0t0d0s1  /dev/rdsk/c0t0d0s1   /usr/local    ufs   2   yes   logging
```

Next, we will do cron administration to enable system accounting, modify cron configuration files and enable the processing of outgoing mail:

```
# crontab -r adm
# crontab -r lp
# crontab –e sys {remove leading # from crontab entries}
```

Edit /etc/init.d/perf and uncomment indicated lines in script to enable system accounting. Then enable system accounting:

```
#/etc/init.d/perf start
```

Cron administration

```
# cd /etc/cron.d
# rm  cron.deny at.deny
# echo root > cron.allow
# echo root > at.allow
# chown root:root *.allow
# chmod 400 *.allow
```

Add this entry to root's cron to process outgoing mail:

```
# crontab –e root
0 * * * * /usr/lib/sendmail -q
```

Next we will modify the banners to for telnet and ftp not to alert would-be hackers of our operating system version. Keeping in mind that there should not even be an incoming telnet or ftp session because we've deleted automatic starting of these daemons, but add just in case:

```
# cd /etc/default
# echo 'BANNER="" ' | tee telnetd | tee ftpd
```

Check the file /etc/default/login and make sure that the line PASSREQ=yes is there and not commented out.

```
PASSREQ=YES
```

Now, edit */etc/default/passwd* and adjust entries for password specific properties. This is highly recommended for security even if this is a workstation. You should come up with some reasonable values for your workstation:

```
MAXWEEKS= { your value }
MINWEEKS= { your value }
PASSLENGTH = 8
```

Edit */etc/default/inetinit* and set the TCP/IP initial sequence numbers parameter to 2 for improved security relating to TCP/IP sequence numbers to prevent the prediction of tcp/ip packet sequence numbers which can lead to session hijacking and other malicious activity:

```
TCP_STRONG_ISS=2
```

Disable *Stop-A* access to your workstation to prevent the halting of the kernel and the dropping into the boot prom prompt by editing the file */etc/default/kbd* and adding:

```
KEYBOARD_ABORT=disabled
```

Edit /etc/default/login and set to these initial values in /etc/default/login for the root account to be able to log in only on the console and also log failed attempts:

```
CONSOLE=/dev/console
SLEEPTIME=4
RETRIES=2
SYSLOG_FAILED_LOGINS=2
```

Remove the following line from */etc/inittab* to prevent serial port logins from attackers who will simply walk up to your box and stick a serial cable in the serial port to get a login prompt:

```
sc:234:respawn:/usr/lib/saf/sac –t 300
```

11 Account Administration

Next, we will delete unnecessary users and assign */dev/null* shells for unnecessary user login shells. This will prevent unauthorized logins by users to unprivileged accounts:

```
# for user in uucp nuucp ip smtp listen nobody4
> do
> /usr/sbin/passmgmt –d $user
> done
# for user in adm daemon bin nobody noaccess
> do
> /usr/sbin/passmgmt –m –s /dev/null $user
> done
```

We will remove rhosts authentication to prevent remote logins and shell access to our workstation via the */etc/pam.conf*: We do this because we will allow access to our system only one way, through the encrypted ssh daemon.

```
# grep –v rhosts_auth /etc/pam.conf > /etc/pam.new
# mv /etc/pam.new /etc/pam.conf
# chown root:sys /etc/pam.conf
# chmod 644 /etc/pam.conf
```

Also, create dead files that cannot be altered for rhosts based authentication so that they cannot be added or modified by hackers for both root and your daily working account:

```
# cd /root
# for file in .rhosts .shosts. .netrc /etc/hosts.equiv
>do
> cp /dev/null $file
> chown root:root $file
> chmod 000 $file
> done
# cd /export/home/user
# for file in .rhosts .shosts .netrc
> do
> cp /dev/null $file
> chown root:root $file
> chmod 000 $file
> done
```

12 Logging

By default, solaris accounting packages do not log login/logout attempts. To enable this feature, create these empty files for future authorization logging:

```
# touch /var/adm/loginlog /var/adm/authlog
# chown root:sys /var/adm/loginlog /var/adm/authlog
# chmod 600 /var/adm/loginlog /var/adm/authlog
```

Also, configure the syslog daemon not to listen for broadcasts and log events only for this host:

```
cp /etc/init.d/syslog /etc/init.d/newsyslog
chmod 744 /etc/init.d/newsyslog
chown root:root /etc/init.d/newsyslog
```

Edit the newsyslog script to add the –t flag to syslogd startup line:

```
/usr/sbin/syslogd –t > /dev/msglog 2>&1 &
```

And now replace the link:

```
rm –f /etc/rc2.d/S74syslog
ln –s /etc/init.d/newsyslog /etc/rc2.d/S74syslog
```

13 Configure SSH

Because earlier we disabled all forms of remote access logins, we will need a way to communicate with this workstation from a remote host. We will use the ssh protocol for this purpose. It is a very secure way of doing application level communications because all traffic is encrypted.

Install an sshd init script as shown as /etc/init.d/ssh with a link to /etc/rc2.d/S89sshd for future remote secure connections:

```
#!/sbin/sh

case "$1" in
'start')
        if [ -x /usr/local/bin/sshd –a –f /usr/local/etc/sshd_config ]; then
                /usr/local/sbin/sshd
        fi
        ;;
'stop')
        pkill sshd
        ;;
*)
        echo "Usage: $0 { start | stop } "
        ;;
esac
exit 0
```

Also, let's generate our server's host keys:

```
# ssh-keygen –t rsa1 –f /usr/local/etc/ssh_host_key –N ""
# ssh-keygen –t rsa –f /usr/local/etc/ssh_host_rsa_key –N ""
# ssh-keygen –t dsa –f /usr/local/etc/ssh_host_dsa_key –N ""
# /etc/init.d/sshd start
```

Edit the /usr/local/etc/sshd_config file to allow X11 Forwarding:

```
X11Forwarding yes
```

And also configure the /usr/local/etc/ssh_config ssh client file to allow X11 Forwarding:

```
ForwardAgent yes
ForwardX11 yes
```

14 Configure KDE - Create an .xinitrc file for starting X windows

First, configure your M64 graphics adapter in the ultra 10 via the m64config command to display at least 1024x768 and depth of 24:

```
m64config –res 1024x768 –depth 24
```

Next, add the necessary path variables for the KDE system to your .profile and root's. Here is a sample of root's .profile:

```
VISUAL=vi
LD_LIBRARY_PATH=/opt/sfw/kde/lib:/usr/local/lib:$LD_LIBRARY_PATH
PATH=/usr/local/bin:/opt/sfw/bin:/opt/sfw/kde/bin:/usr/openwin/bin:/usr/sbin:/usr/bin
export PATH VISUAL LD_LIBRARY_PATH
umask 077
```

root's .profile example

Create a *.xinitrc* file in your home directory for starting X windows for root and your work account. We will not be using xauth or xsession functionality to do this because this is a workstation and we want this to be a very basic environment with as few daemons running as possible.

```
# echo 'startkde 2> /dev/null' /root/.xinitrc
# cp /root/.xinitrc /export/home/user
# chown  user:group /export/home/user/.xinitrc
```

15 Enable Basic Security Module

We will enable C2 security on this box for the tracing of system activity back to specific users on the system and the logging of any suspicious events. Please note that this will create very large log files and will slow the system down by about 5-10%. But, since this is a workstation, I feel this is something you can live with. The ability to trace any suspicious activity through audit logs is a very powerful security feature. For more information on auditing please see the 'Auditing in the Solaris 8 Environment' URL listed in the reference section.

Enable the Basic Security Module for C2 level auditing of system activities:

```
# cd /etc/security
# ./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 move aside /etc/rc2.d/S92volmgt
bsmconv: INFO: turning on audit module
bsmconv: INFO: initializing device allocation files.

The Basic Security Module is ready.
It there were any errors, please fix them now.
Configure BSM by editing files in /etc/security.
Reboot this system now to come up with BSM enabled.
#
```

And now configure the /etc/security/audit_control file to read:

```
dir:/var/audit
flags:lo,ad,-all,^-fm
naflags:lo,ad
minfree:20
```
/etc/security/audit_control file

And have the system generate new audit logs every hour by adding this entry to root's crontab entry.

```
# crontab –e root
0 * * * * /usr/sbin/audit –n
```

Now, reboot the system to single user mode for backup:

```
# reboot -- -s
```

16 Backup

Make two backups of your system using the ufsdump command for emergency purposes and recovery:

According to Step 4.1 of the *Solaris Security Step by Step:*

- ➢ Mount all filesystems
- ➢ fsck
- ➢ mount –a

Backup all ufs filesystems to tape

- ➢ mt /dev/rmt/0
- ➢ for dir in / /usr /var /usr/local
- ➢ do

      ufsdump 0f /dev/rmt/0n $dir
- ➢ done
- ➢ mt /dev/rmt/0 rewoffl

Write protect both tapes and store one tape offsite!

17 Test X Windows

Reboot again and log in. To start X windows from the shell prompt, just type *xinit*. To get out of X windows, just move the mouse to a blank area on the root window, press the right mouse button to obtain the selection menu and choose logout. You will then be back at your shell prompt. In addition, try to telnet to a remote X server and running a simple X application such as xclock to have the window appear on your box. When you check your DISPLAY variable, it should be hostname:10.0.

18 Test access from the outside in

From a machine on your subnet, try to telnet and ftp in to your box. You should get a connection failure. Now, ssh in to your box and you should have success. Also, do a remote nmap scan of your system to see if any suspicious daemons are listening on any ports. Only ssh should be listening on the TCP v4 ports below 1024, the privileged port range. On the local machine, do a netstat –an command and look at the output and compare.

19 Final Thoughts

This setup I feel is a very basic to display remote and local X windows clients using the KDE environment. It adheres to the SANS philosophy of minimal software installation to achieve a task. Although the system is now somewhat secure, it takes a systematic approach of security maintenance to keep it that way. I would suggest that you check the http://sunsolve.sun.com site every week to check for new 8_Recommended.zip files. A new file comes out about once a week with updates and fixes. In addition, it is important to check your own workstation log files regularly for suspicious activity and also change your password frequently using difficult to guess passwords that cannot be found in any dictionary. In addition, it's a good idea to switch off the ssh daemon when your at your machine and when your not expecting any remote logins and file transfers. You can easily do this by executing /etc/init.d/sshd {start|stop}. If you are sitting at your machine, I don't think you'll be using ssh to get into your box, right?? So, follow these steps of maintenance and you should have a very secure workstation over time.

- Check for and download 8_Recommended.zip every week
- Check logs regularly for suspicious activity
- Change password to unguessable password on a regular basis

References

Pomerantz, Hal Solaris Security Step by Step v. 2.0, Sans Institute, 2001

Garfinkel, Simson (et al) Practical UNIX and Internet Security Sebastopol: O'Reilly & Associates,April 1996.

Gregory, Peter H. Solaris Security Upper Saddle River: Sun Microsystems Press, 1999

Freeland, Curt, et al. Solaris for Managers and Administrators–Third Edition Albany: OnWard Press, 2000

Galvin, Peter. The Solaris Security FAQ 01/01/2001
URL: http://www.itworld.com/Comp/2377/security-faq

Solaris Security Guide
URL: http://www.sabernet.net/papers/Solaris.html

Marks, Evan, R. (et. al) Solaris Solutions for System Administrators, John Wiley and Sons: 2000

Noordergraaf, Alex (et al.) Solaris Environment Security 01/2000
URL: http://www.sun.com/blueprints/security.pdf

Osser, William (et. al) Auditng in the Solaris 8 Environment
URL: http://www.sun.com/blueprints/0201/audit_config.pdf

Noordergraaf, Alex (et al) Solaris Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology
URL: http://www.sun.com/blueprints/1299/minimization.pdf

Frost, Armoring Solaris
URL: http://www.packetst0rm.net/armor_solaris.txt

Nash, David KDE Bible California: IDG Books, 2000