Global Information Assurance Certification Paper

Building a Secure Solaris 9 JumpStart Server

Bayly Eley
September 14, 2004

GCUX Practical v2.0
Option1 - Securing UNIX Step by Step

Abstract

Over the past ten years there has been a rapid proliferation of servers in our data centers, as well as, a need for increased security. Solaris JumpStart can help with both of these challenges. Solaris JumpStart is an installation server included with the SPARC Solaris operating environment. Gone are the days of shuffling CDROMs for hours while performing an installation. Nor does each installation have to be painstakingly hardened. JumpStart enables administrators to install SPARC Solaris in an automated, unattended, and consistent manner.

This paper will present how to build a secure Solaris 9 JumpStart server to be used to install secure Solaris 9 servers. The necessary hardware and software will be discussed, and the installation, configuration, hardening, and maintenance will be detailed. In addition, the hardening process will be tested using common UNIX applications and common vulnerability assessment tools, such as Nessus.

System Description

It is always a good practice to perform installations in a non-networked or securely networked environment. The JumpStart server will be located on an isolated private network, separate from the production network. In addition to the JumpStart server, only JumpStart clients will be placed on this network. Once a client's installation is complete, it can be easily moved to the production network.

The hardware for the JumpStart server will consist of a single Sun Ultra 5 with one 360 MHz UltraSPAR-IIi processor, 128 MB of memory, one 10/100 Ethernet interface, a CDROM drive, and one 8.3 GB IDE hard drive. For the purposes of this project, an additional Sun Ultra 5 will be used as a JumpStart client to test the installation process. The client utilizes a 270 MHz UltraSPARC-IIi processor, 256 MB of memory, one 10/100 Ethernet interface, a CDROM drive, and one 4.0 GB IDE hard drive.

Solaris has a rich history as a UNIX operating system. It is one of the most widely distributed commercial UNIX operating systems in use today. As such, it is the primary operating system in our environment. JumpStart can be configured to install Solaris 2.6, 7, 8, or 9. However, Solaris 9 is the most recent release from Sun Microsystems and is used exclusively in our environment. Therefore, this document will detail the configuration of JumpStart for Solaris 9, and not any previous versions.

The needs and roles of servers vary from site to site. Each of our servers run different applications, so we have found it to be more efficient to use JumpStart for the basic operating system installation, but still install any server specific applications manually. Therefore, this document aims to describe how to create a basic and secure installation that can be built upon.

In addition to the essential operating system daemons, the JumpStart server will need RPC (for bootp), and NFS and TFTP to perform network installations. NFS will be restricted to read-only access on the isolated network. TCP Wrappers will be used to restrict the use of TFTP and other network services to the private network. SSH will be installed and will be the only remote access available to the server and clients. In addition, no remote root log in will be permitted; users must SU to root after being authenticated as a normal user. Only authorized administrators should have access to the JumpStart server.

Regular backups are not necessary for the JumpStart server. The information contained on the server is static. An initial backup after configuration is complete and incremental backups when any configuration changes are made will be sufficient. For this, we will use TAR and burn the archives to CDROM.

The JumpStart server should be located in a secure location. Physical access should be limited to authorized administrators. A computer room with some type of access control should be used. Effective security policies should be in place to enforce such access restrictions.

Risk Analysis

A JumpStart server is a key and vulnerable system. From it, all new installations will be created. The potential for an attacker to poison all installations is real. Therefore, the following risks must be considered:

- Hardware Failure
- Physical Damage
- Software Malfunction
- Internal Attacks
- External Attacks

Hardware failure and physical damage for our JumpStart server is a real and acceptable risk. The JumpStart server is not an operationally critical system; basic vendor support is sufficient for replacement of any failed hardware components, such as: disks, power supplies, etc. Initial and incremental backups will be burned to CDROM and stored in a secure location in case of a catastrophic hardware failure or damage. However, the JumpStart clients will most likely become more operationally critical systems. So, disk mirroring will be implemented, and redundant hardware components should be used when available.

Software malfunction risk is low for the JumpStart server, since JumpStart is part of the Solaris operating system and no third-party applications will be used. To help guard against any operating system issues, Solaris patch clusters will be installed at monthly intervals. Likewise, software malfunction risk is low for the JumpStart clients, as the installation of any third-party applications after installation is complete is beyond the scope of this document. However, all clients will receive the latest Solaris patch cluster as part of the installation process.

Internal attacks do represent some risk, but can be mitigated by restricting the JumpStart services to an isolated network with TCP Wrappers. SSH will insure secure remote access. Adequate security policies and physical isolation will help deter or stop most internal risks.

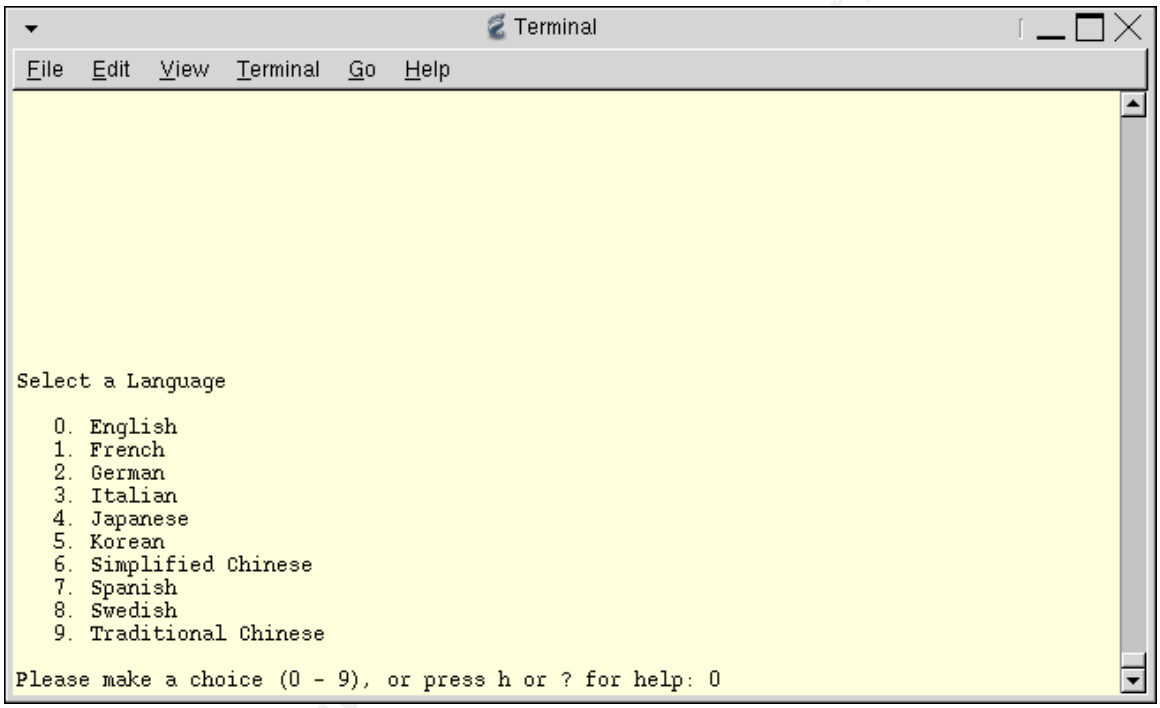
External attacks are possible, however unlikely. A firewall provides perimeter security and SSH, TCP Wrappers, and network isolation provide additional layers of security. No services will be available to the outside, and a well-defined security policy will help make it extremely difficult for an external attacker to penetrate the JumpStart server.

Installation (Operating System)

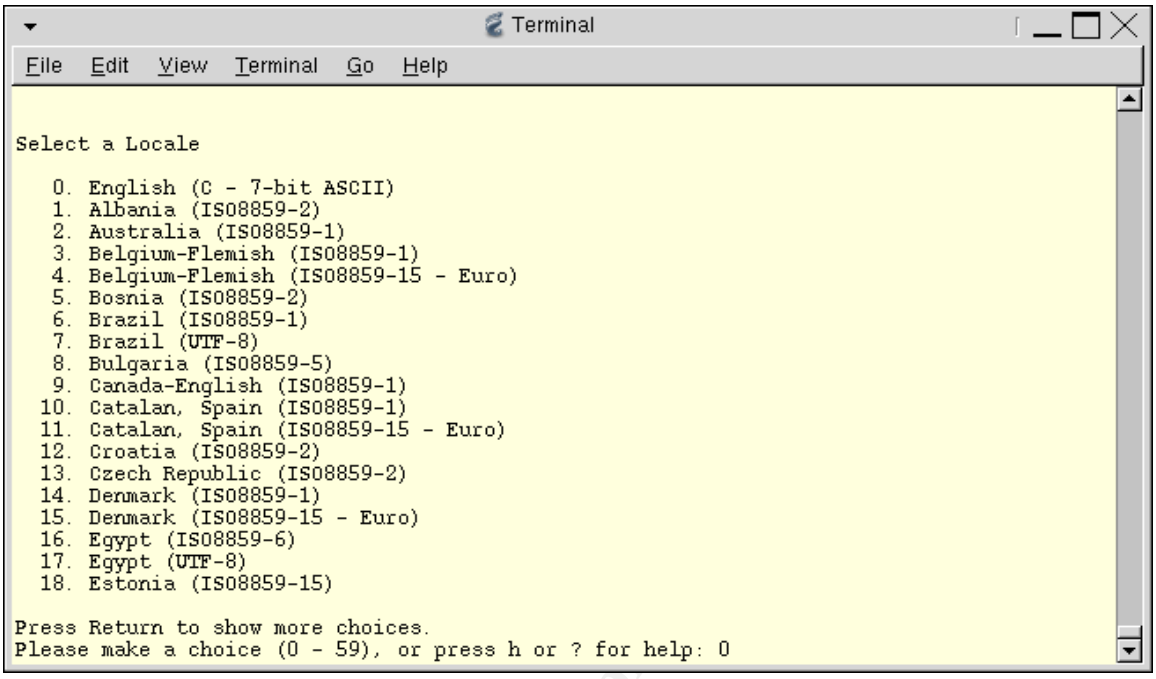
1. Begin by powering on the system. For the Ultra 5 platform, the power button is located on the top right of the front face. At this point, you should be able to insert the Solaris 9 Software CDROM 1/2. Press the eject button on the CDROM, insert the CDROM, and close the drive sled. Once the OpenBoot memory banner is displayed, you may send a "break" using the Stop-A keys (the precise method may differ depending your specific type of terminal connection).

2. At the OK prompt type **`boot cdrom`**.

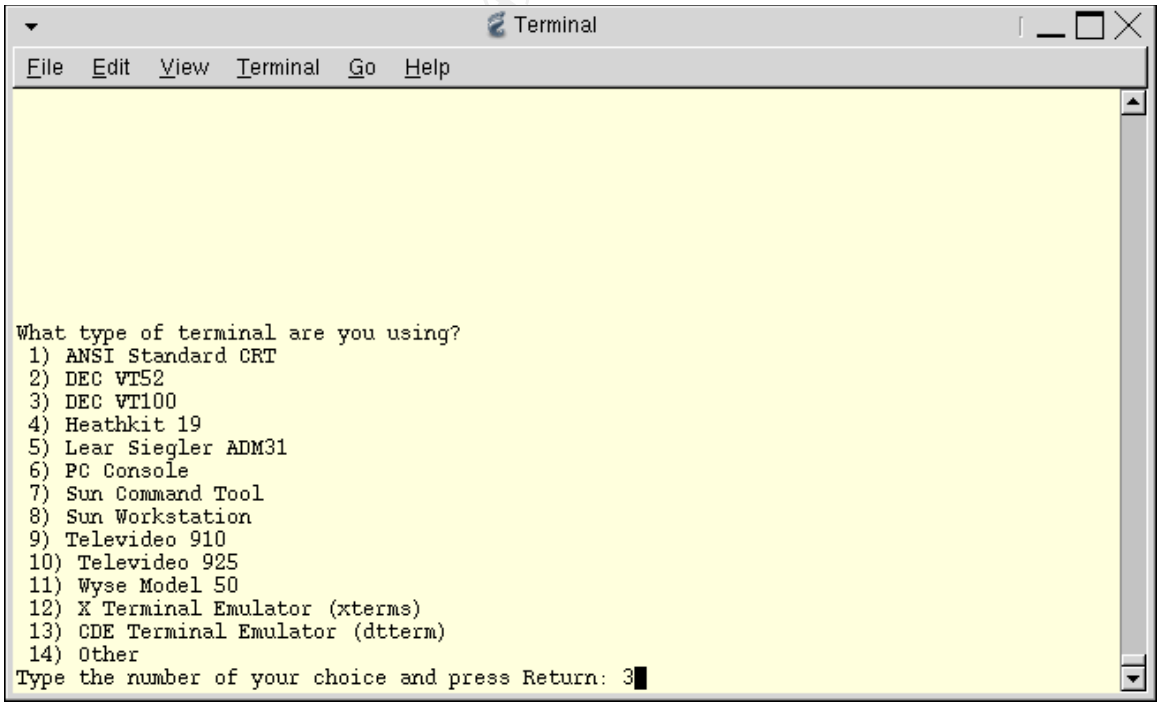
3. The first screen will ask for choice of language. Type “0” for English.



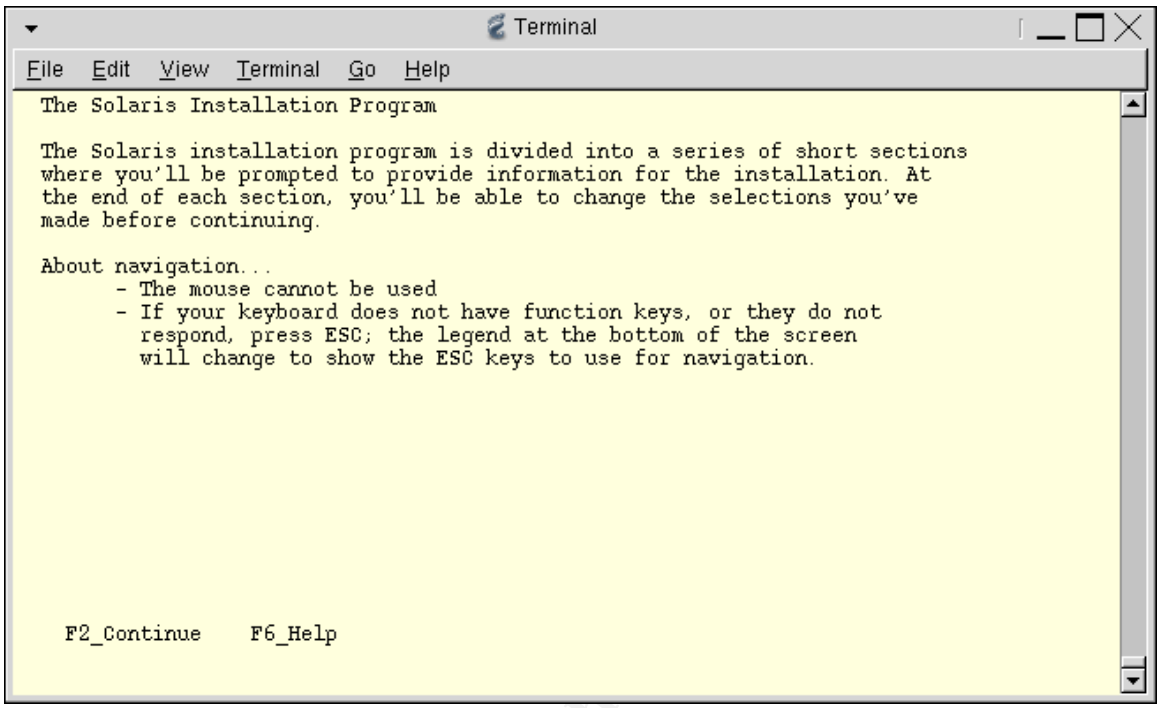
4. The next screen will ask for locale. Type “0”for English.



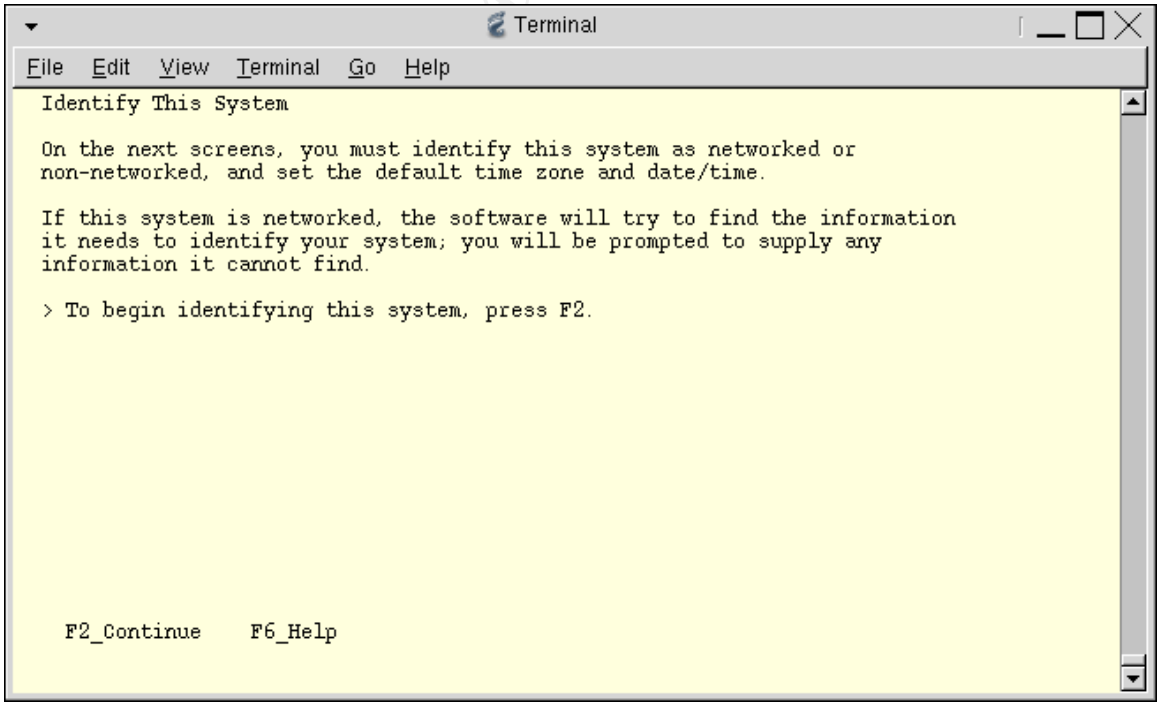
5. The next screen will ask what type of terminal you are using. In most cases, VT100 is appropriate, but your exact environment may differ. Select “3” for VT100.



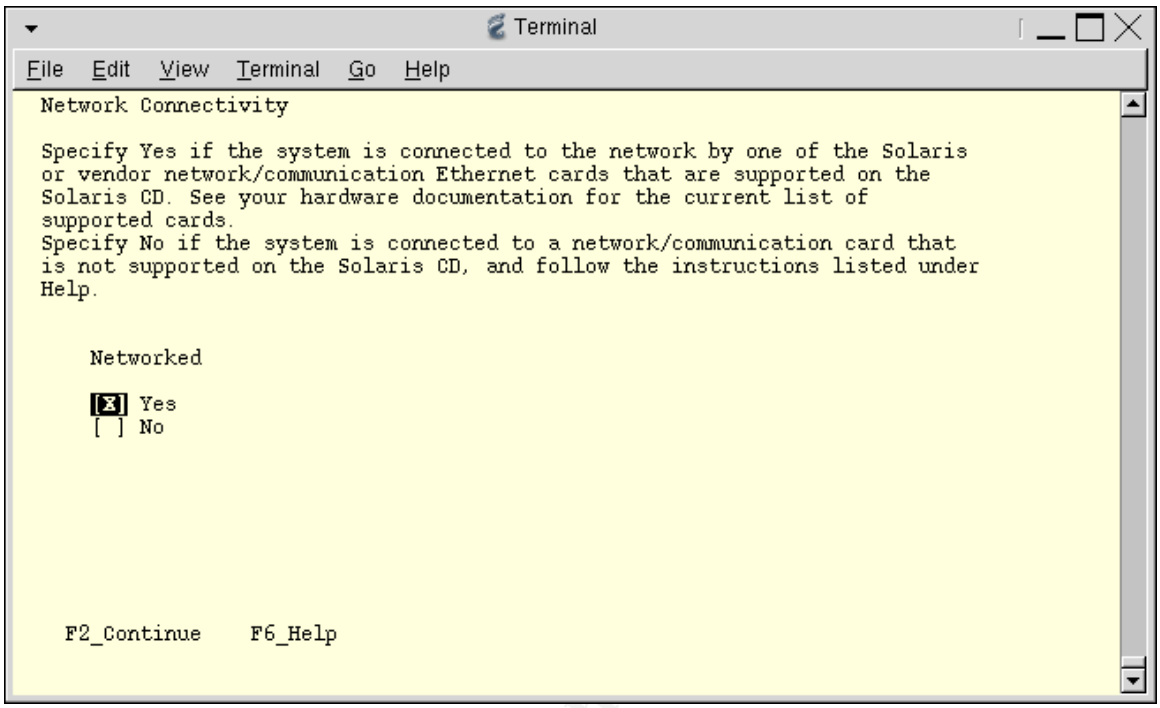
6. Select **F2_Continue** after reading a description of the installation program.



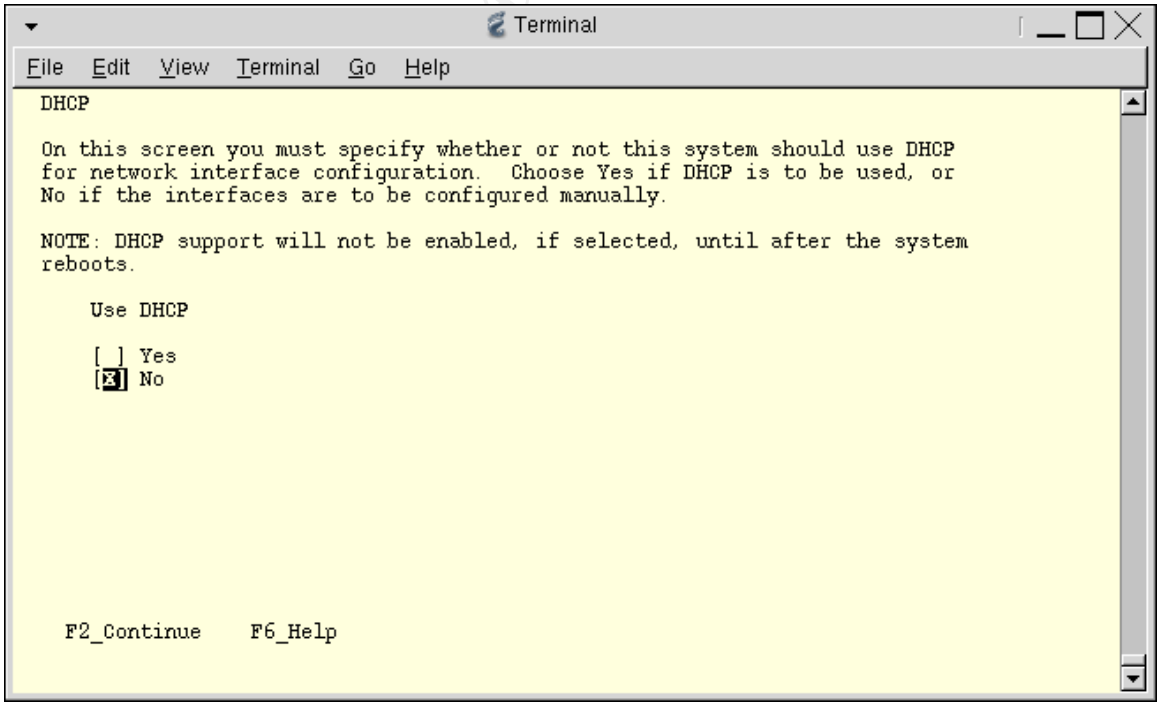
7. Select **F2_Continue** after reading a description about the identification process.



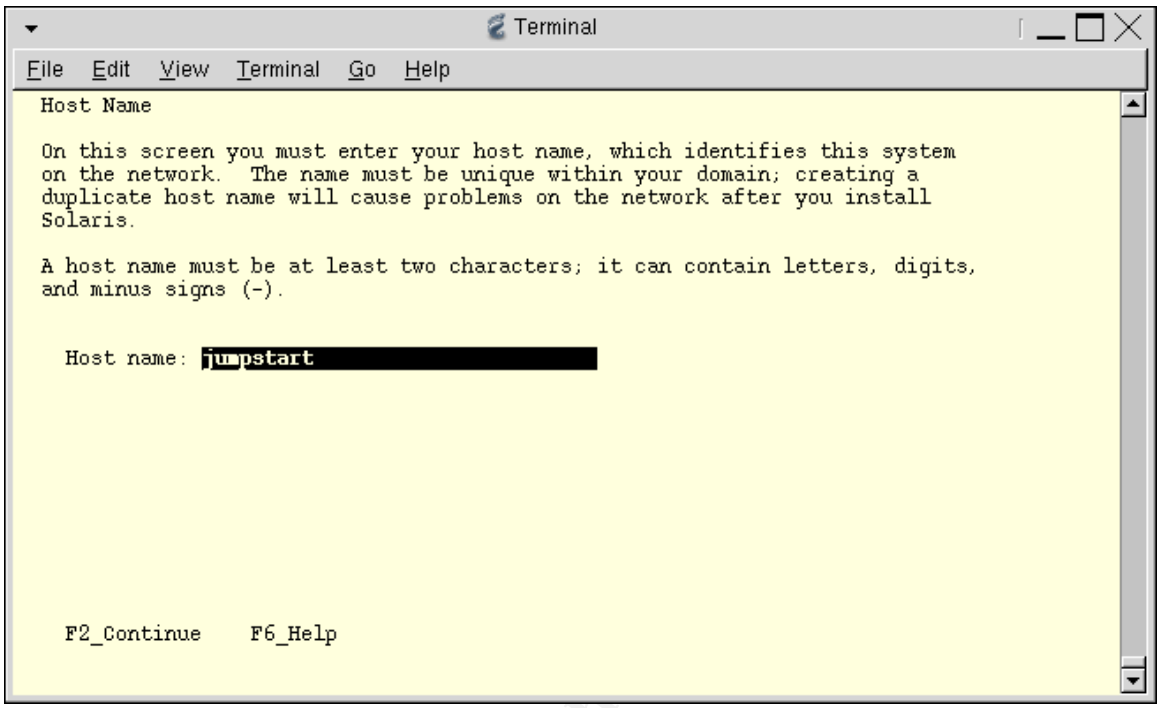
8. Select "Yes" for a networked system, and **`F2_Continue`**.



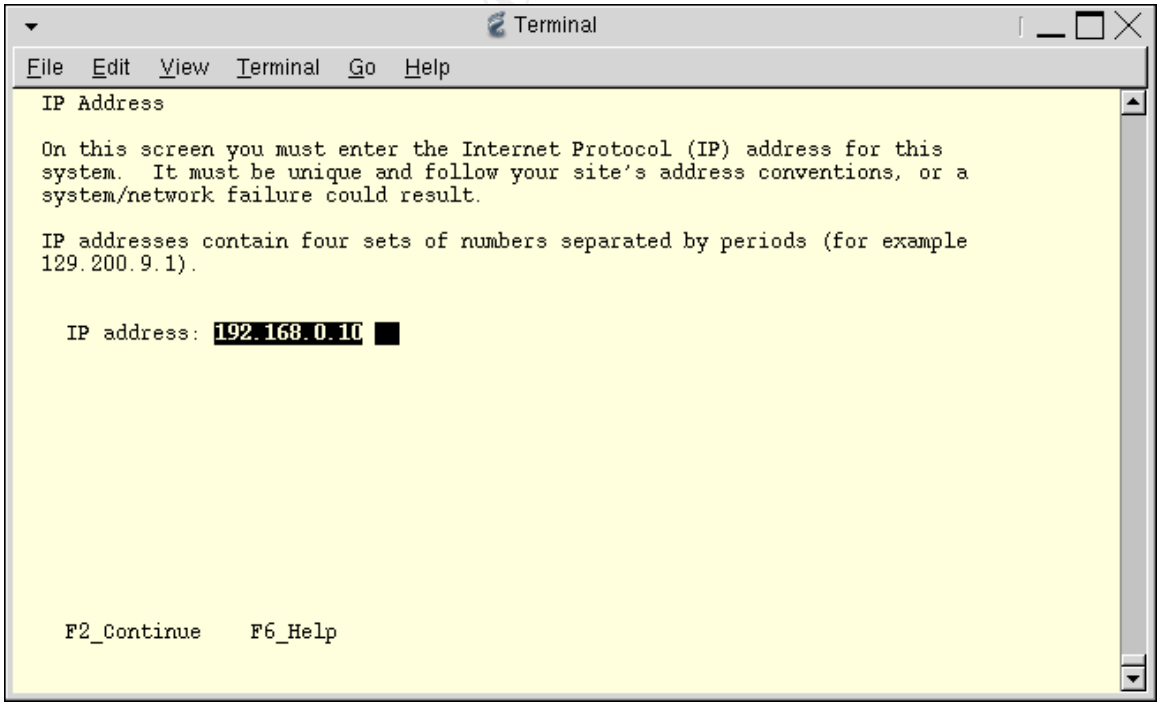
9. Select "No" for DHCP and **`F2_Continue`**.



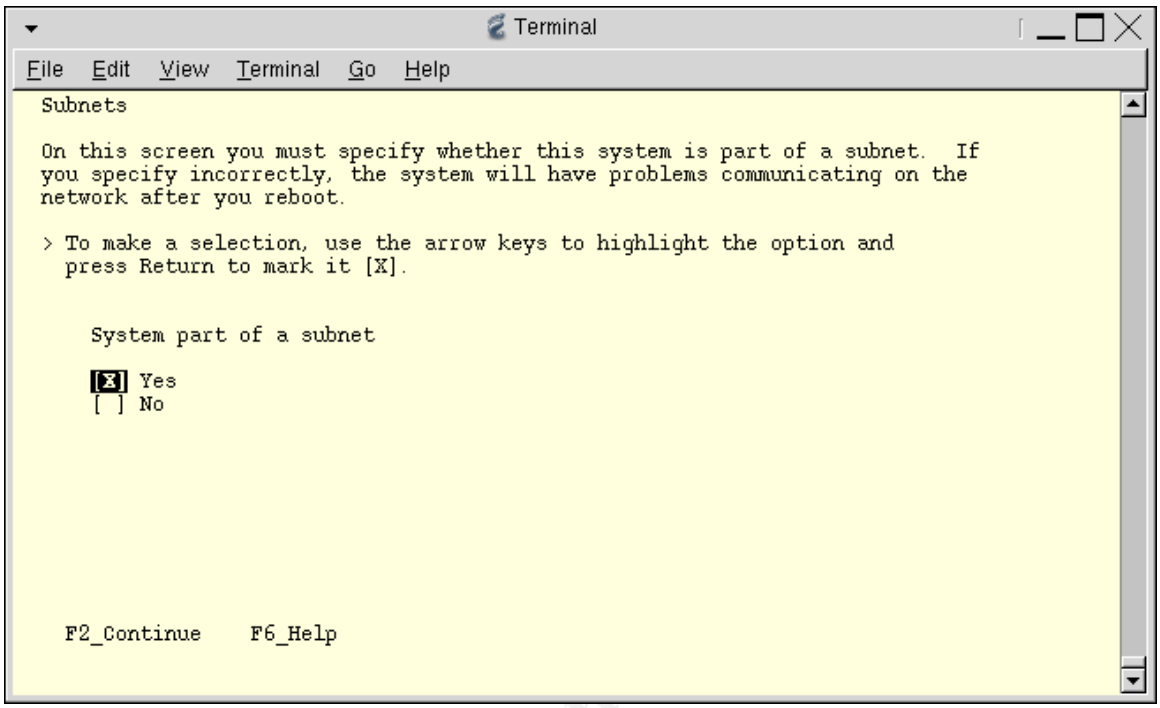
10.Enter a "*Host name"* for the JumpStart server in the space provided and **F2_Continue**.



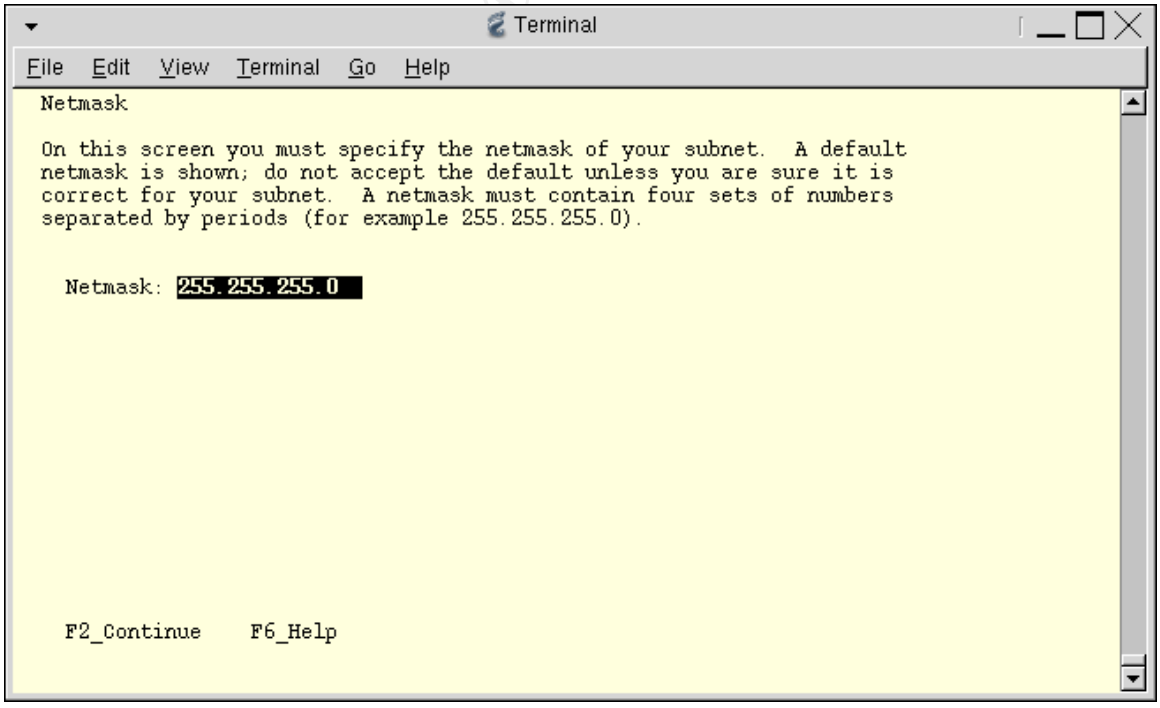
11.Enter the "*IP address"* for the JumpStart server and **F2_Continue**.



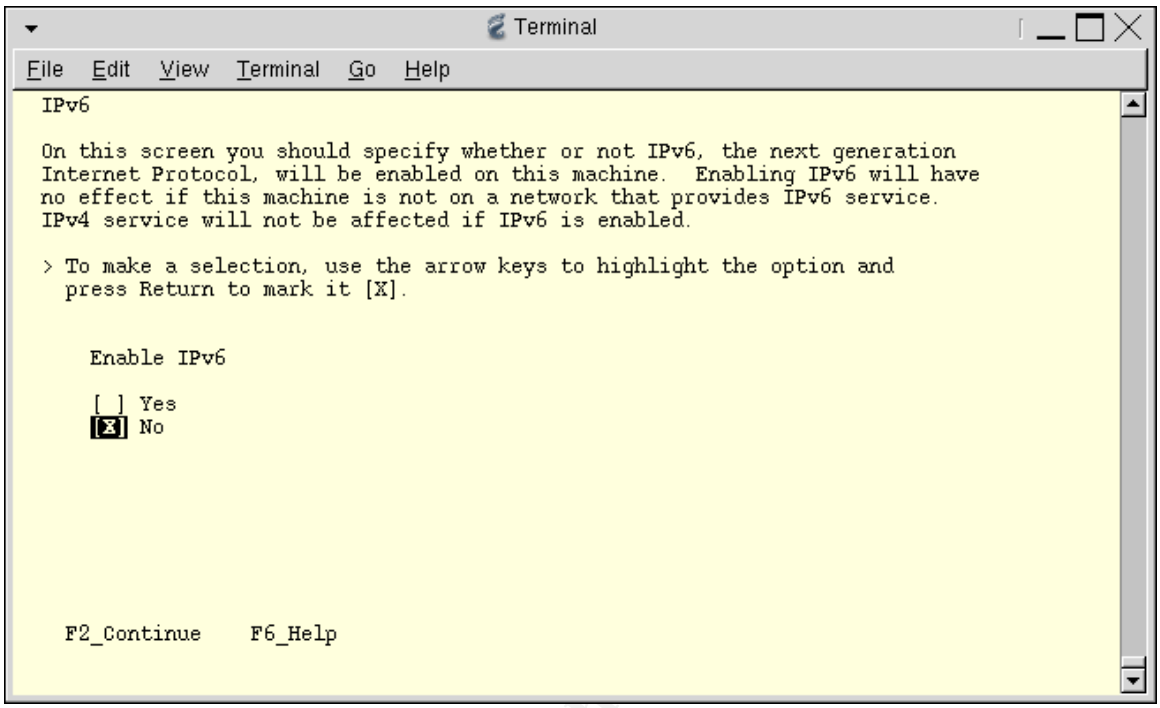
12.Most network configurations are part of a subnet. Select "Yes" and **F2_Continue**.



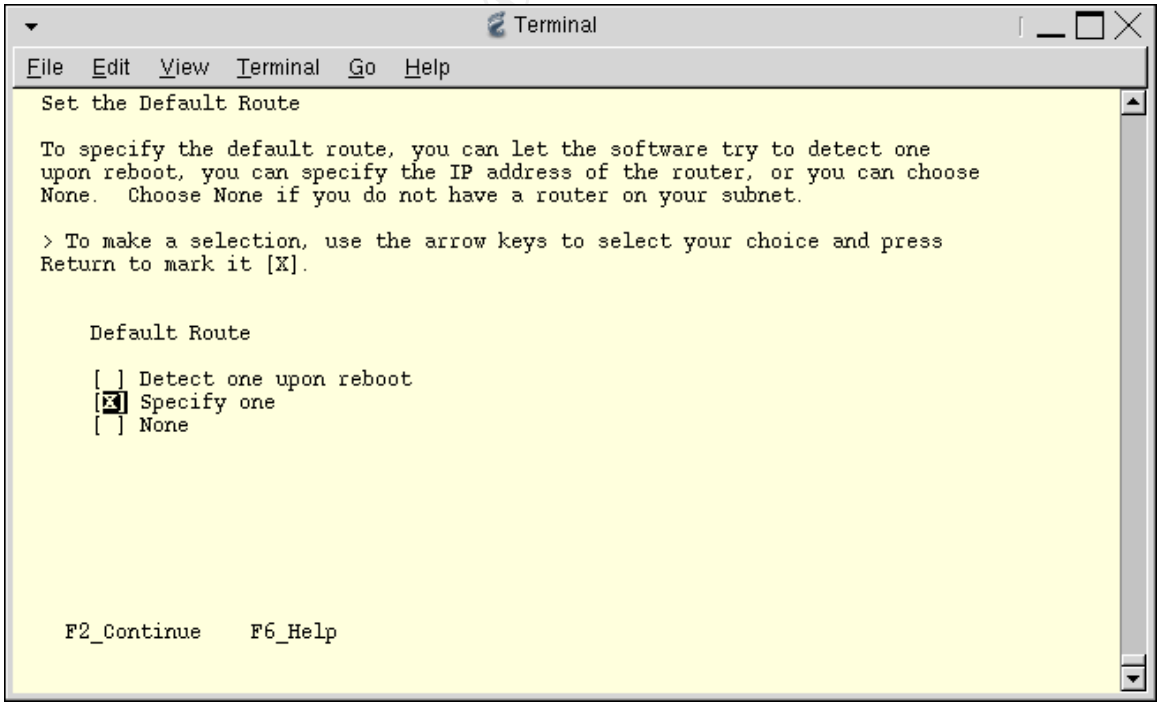
13.Enter the "*Netmask"* and **F2_Continue**.



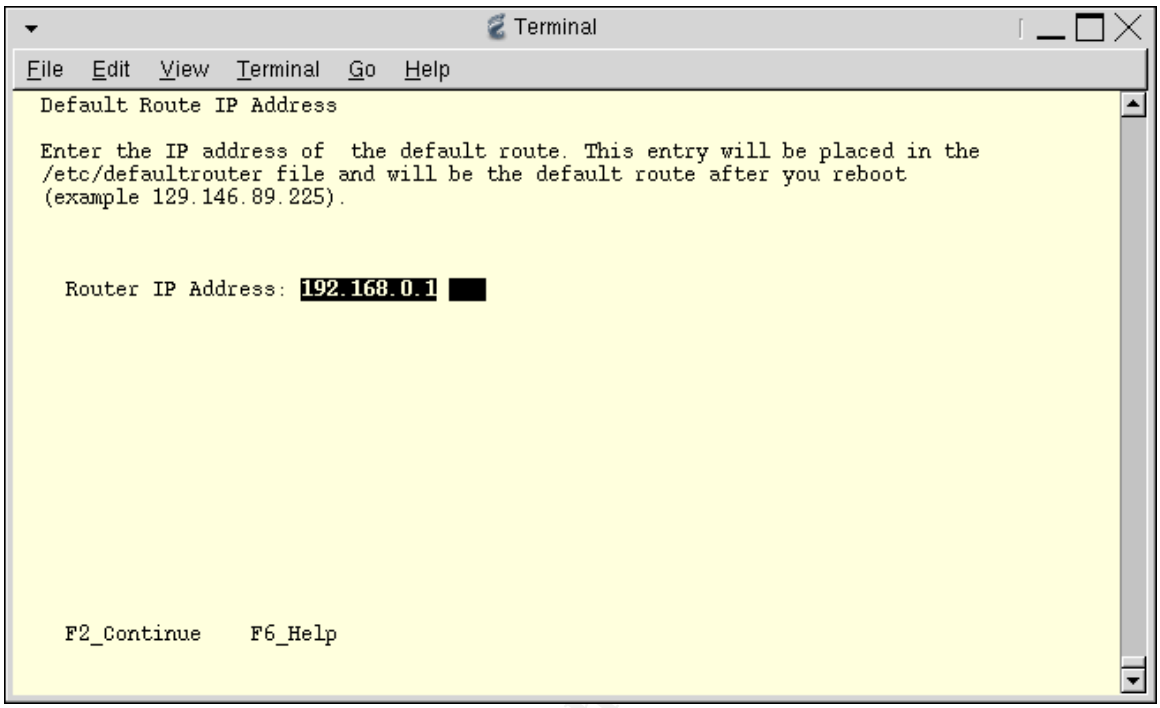
14.Most environments do not yet support IPv6. Select “No” and **F2_Continue**.



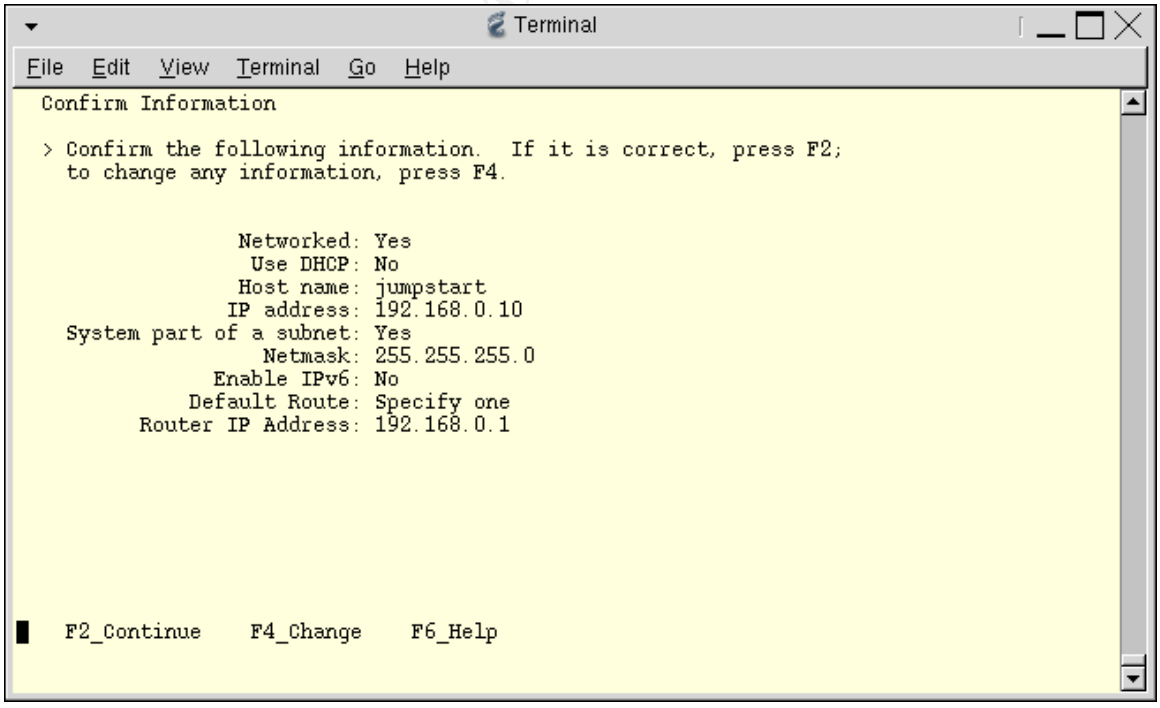
15.Select to specify a “Default Route” (if needed) and **F2_Continue**.



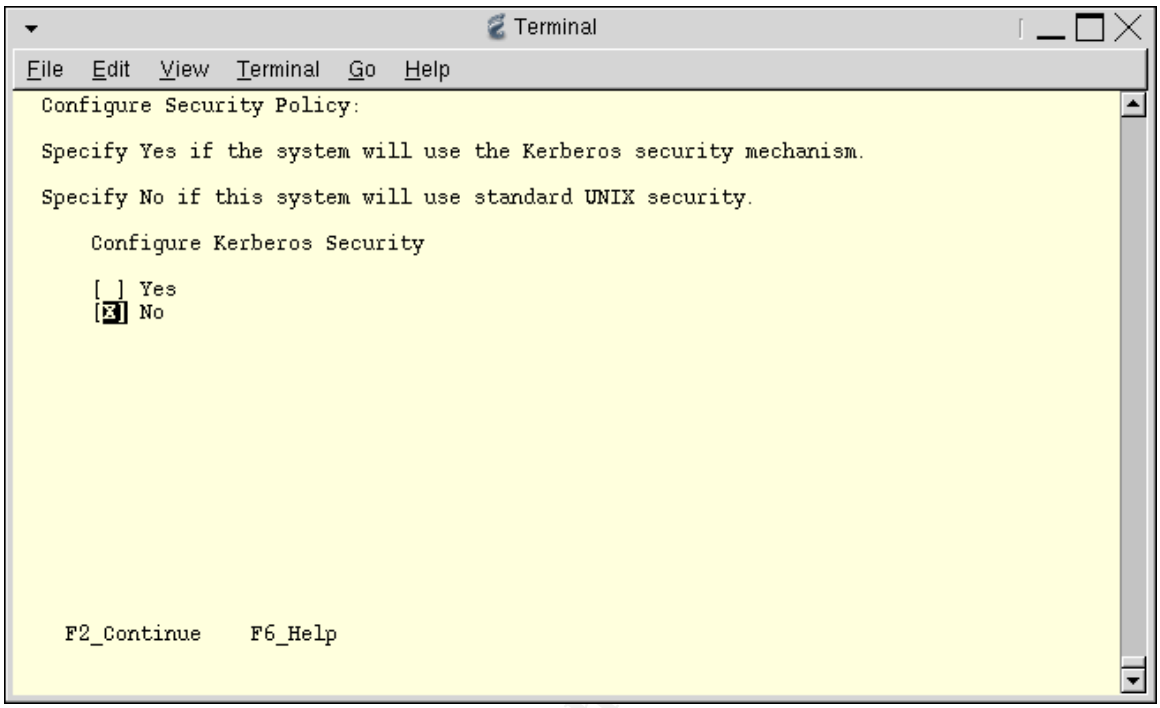
16.Enter the "*Default Router IP Address"* and **`F2_Continue`**.



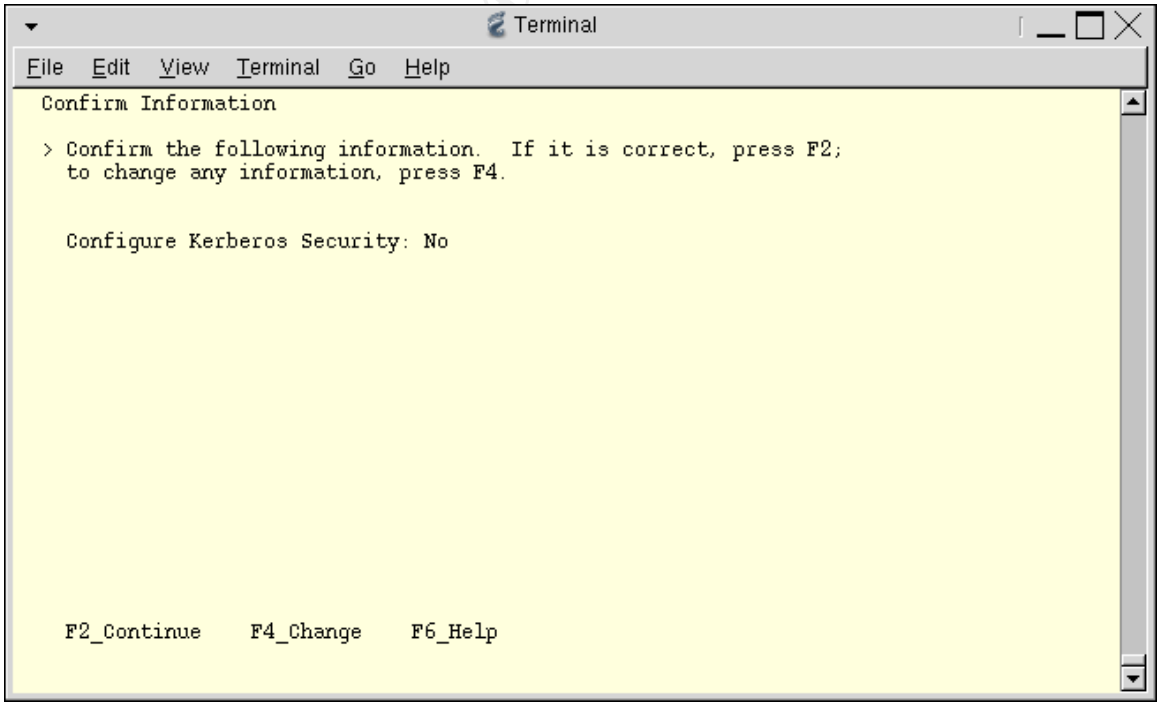
17.Confirm the identification information and **`F2_Continue`**.



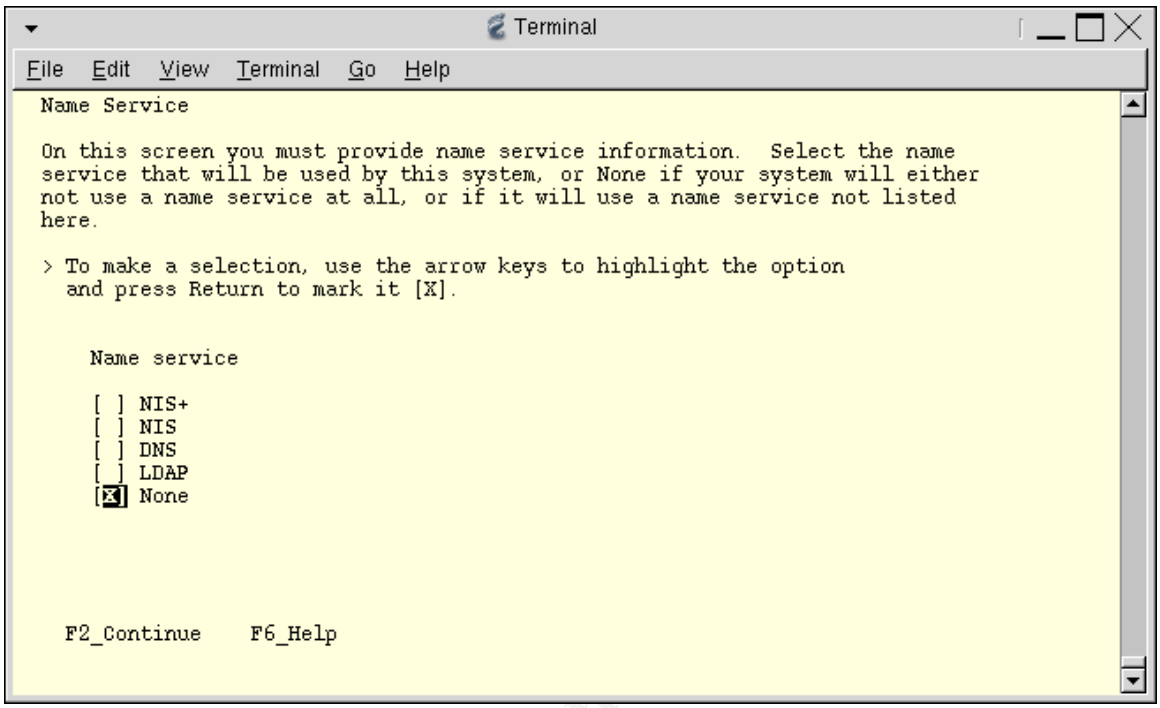
18.Select “No” for “Configure Kerberos Security” and **F2_Continue**.



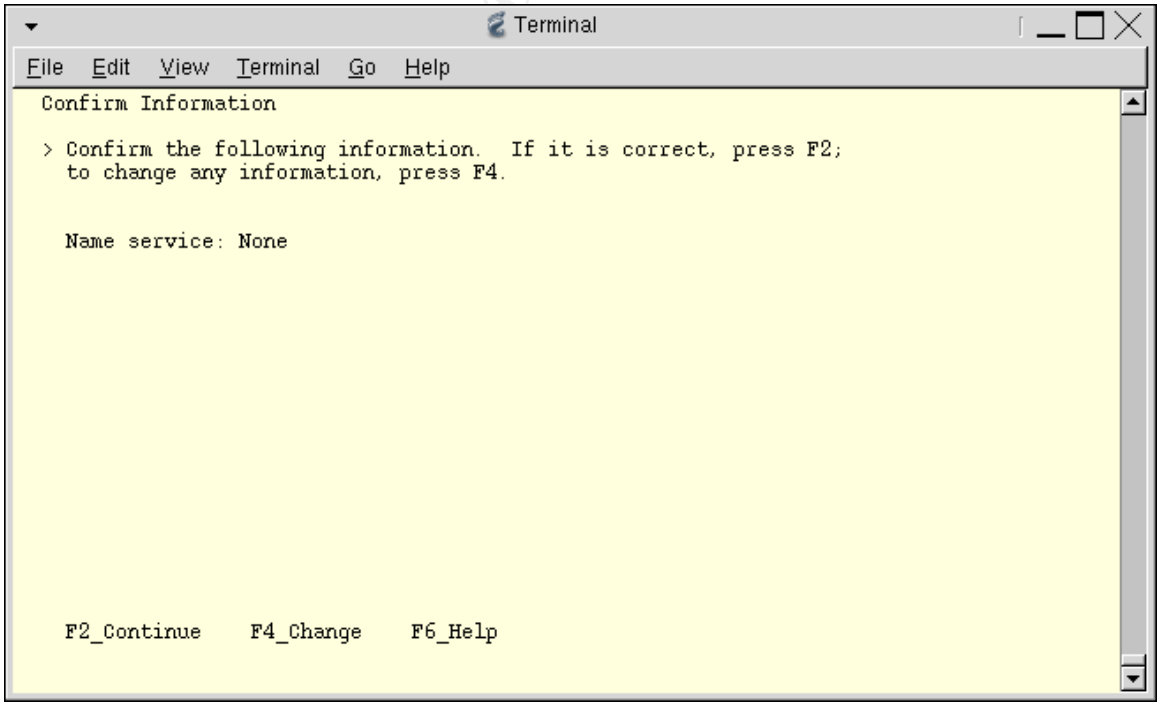
19.Yes, we are sure we do not want to use Kerberos Security. Select **F2_Continue**.



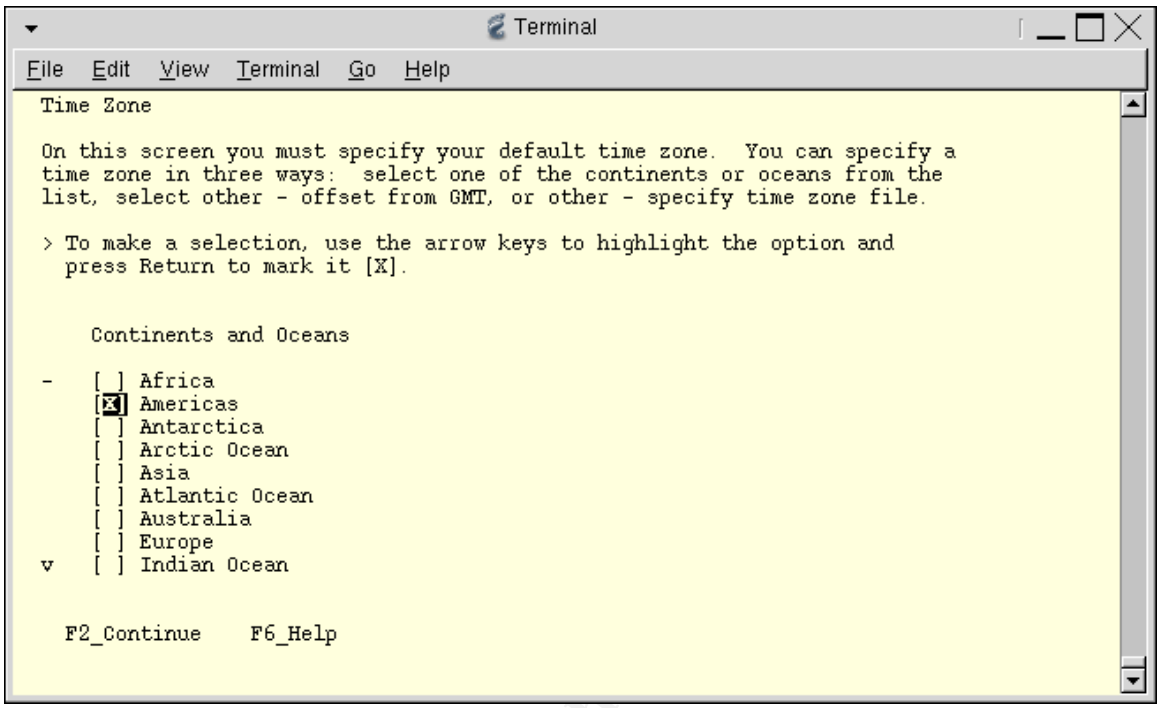
20.Select “None” for the Name Service and **F2_Continue**.



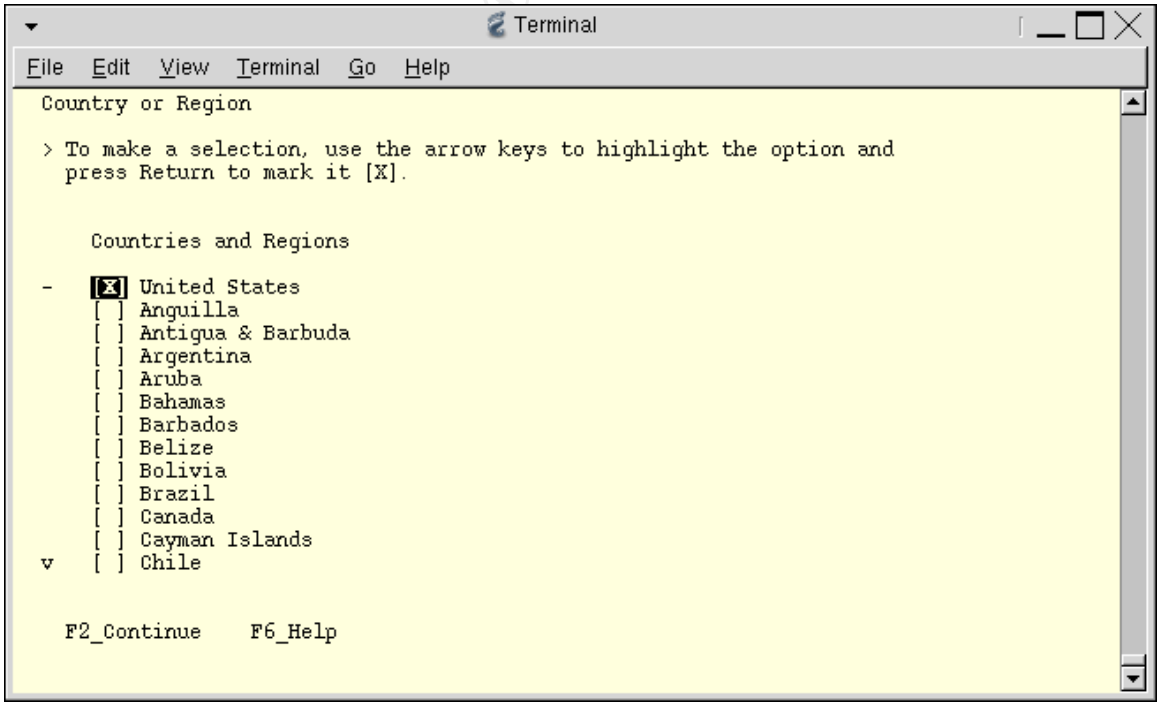
21.Yes, we are sure we do not want to use a Name Service. **F2_Continue**.



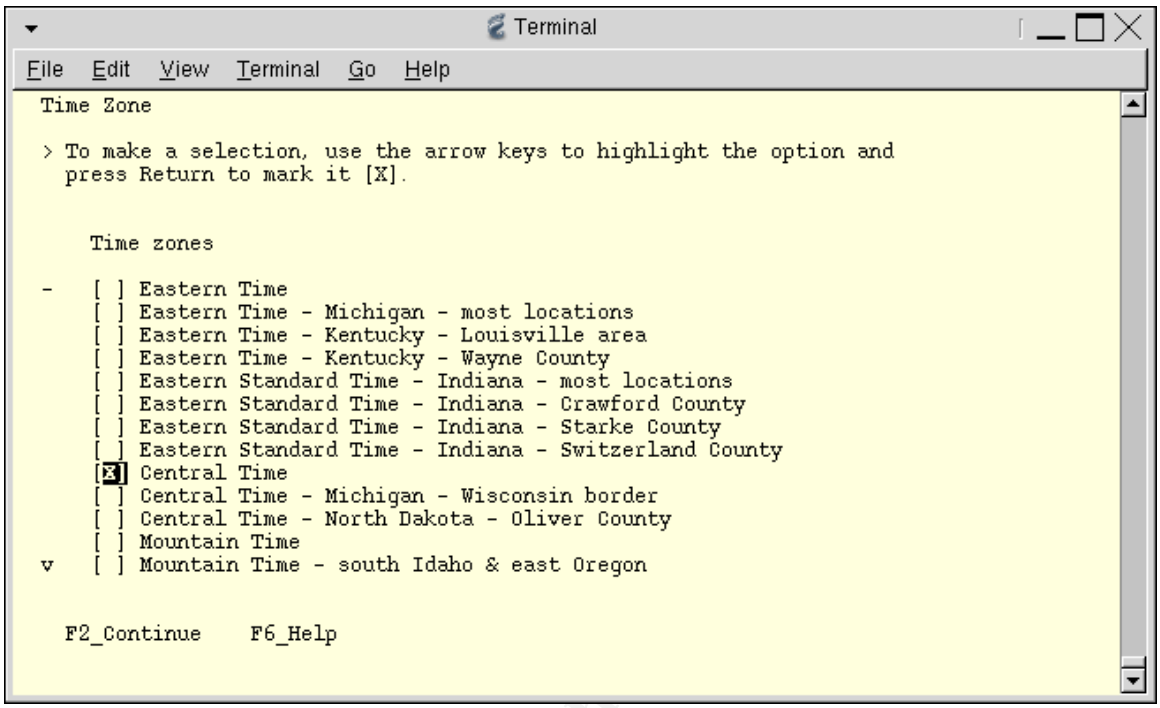
22.Select “Americas” for the Time Zone and **F2_Continue**.



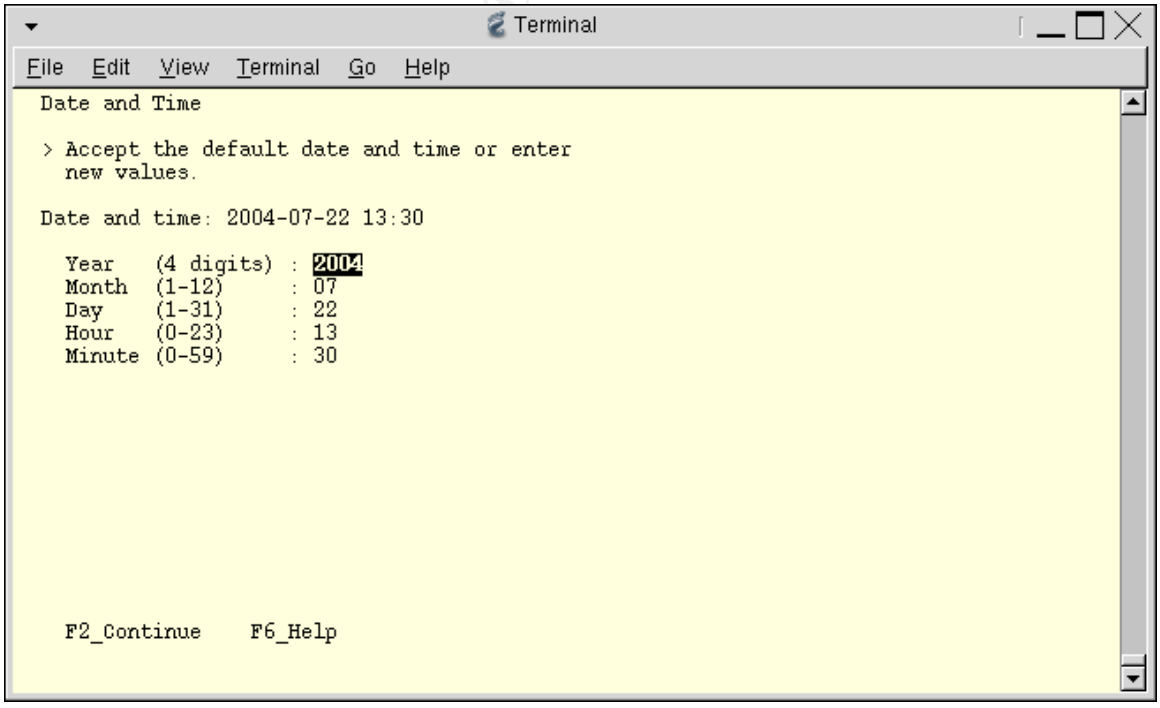
23.Select “United States” for the Country and **F2_Continue**.



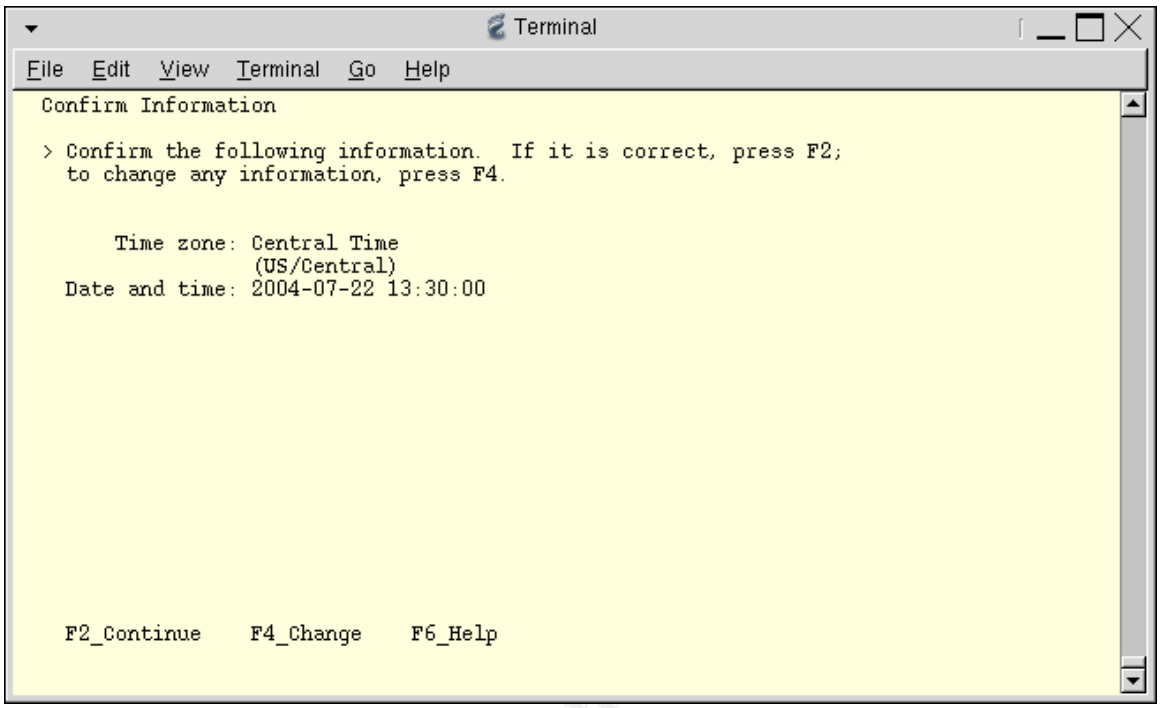
24.Select the “Central” (or other) Time Zone and **`F2_Continue`**.



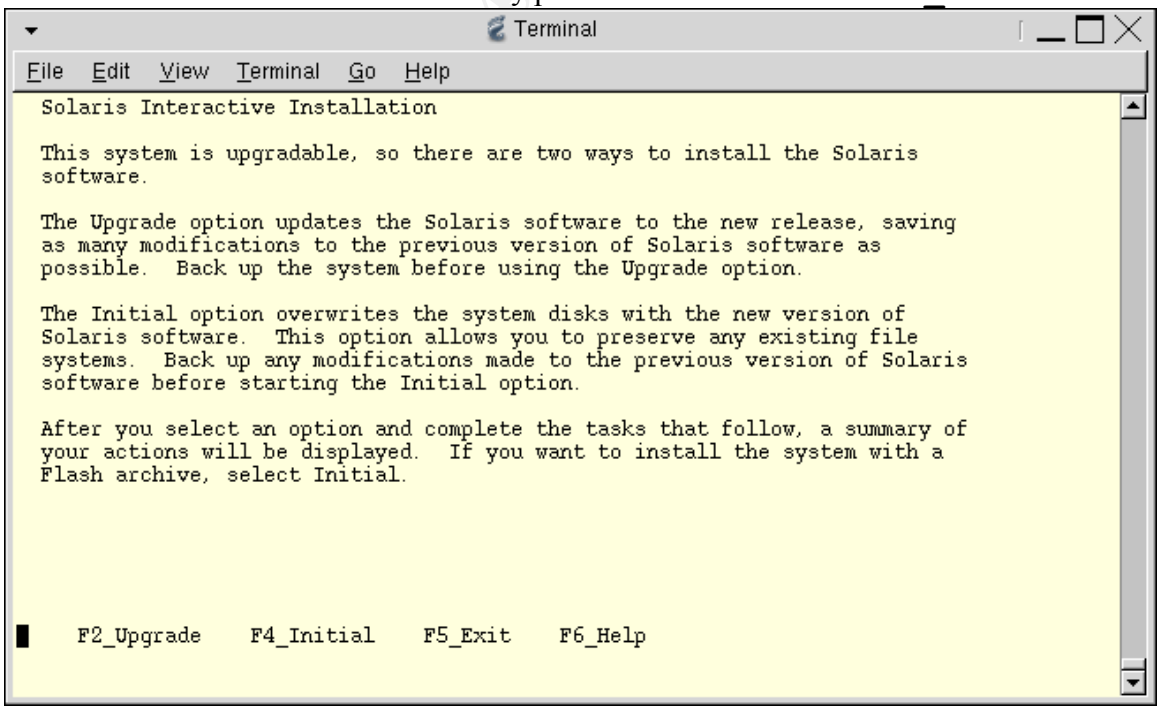
25.Set the year and date, and **`F2_Continue`**.



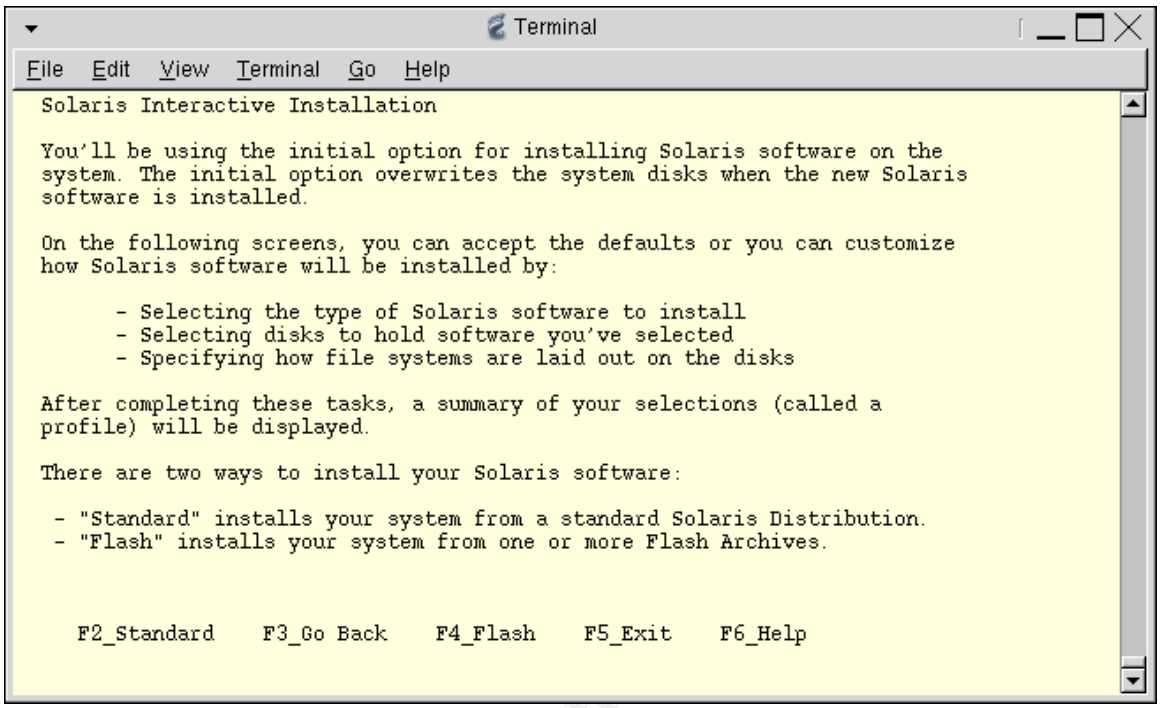
26.Confirm the Time Zone and Date/Time information, and **`F2_Continue`**.



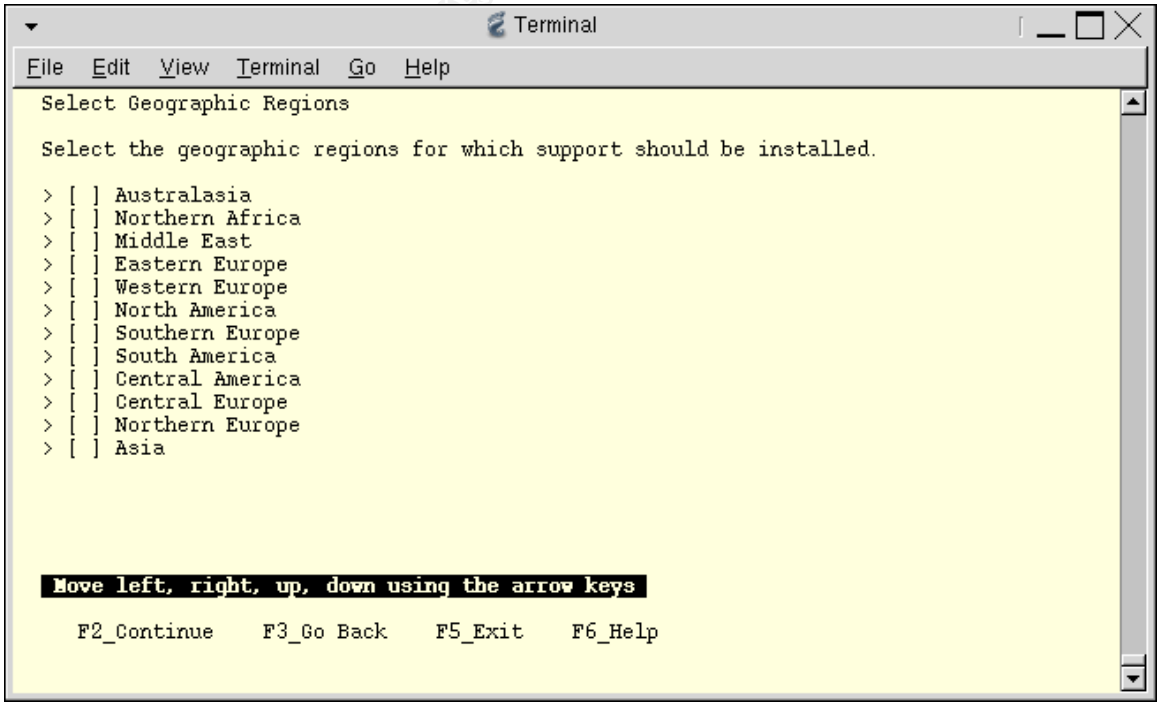
27.Most Sun systems come from the factory with the operating system pre-installed. We want to perform an "Initial" install to overwrite any previous installation. Select **`F4_Initial`**.



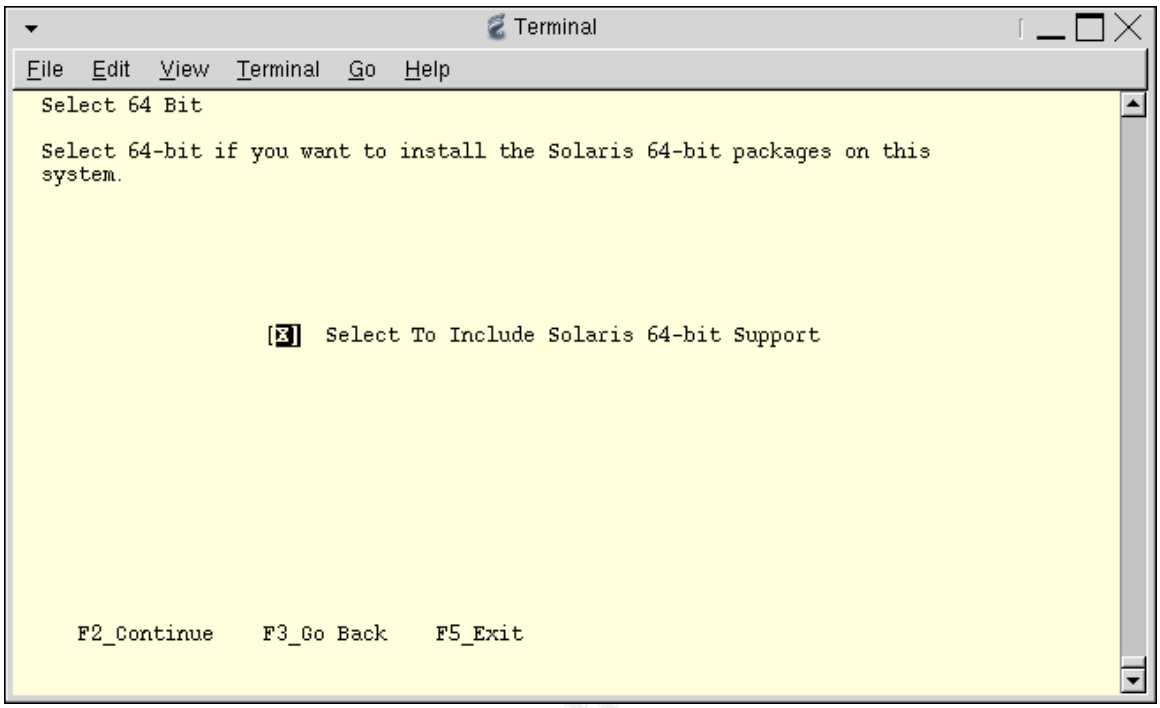
28.Select **F2_Standard** installation.



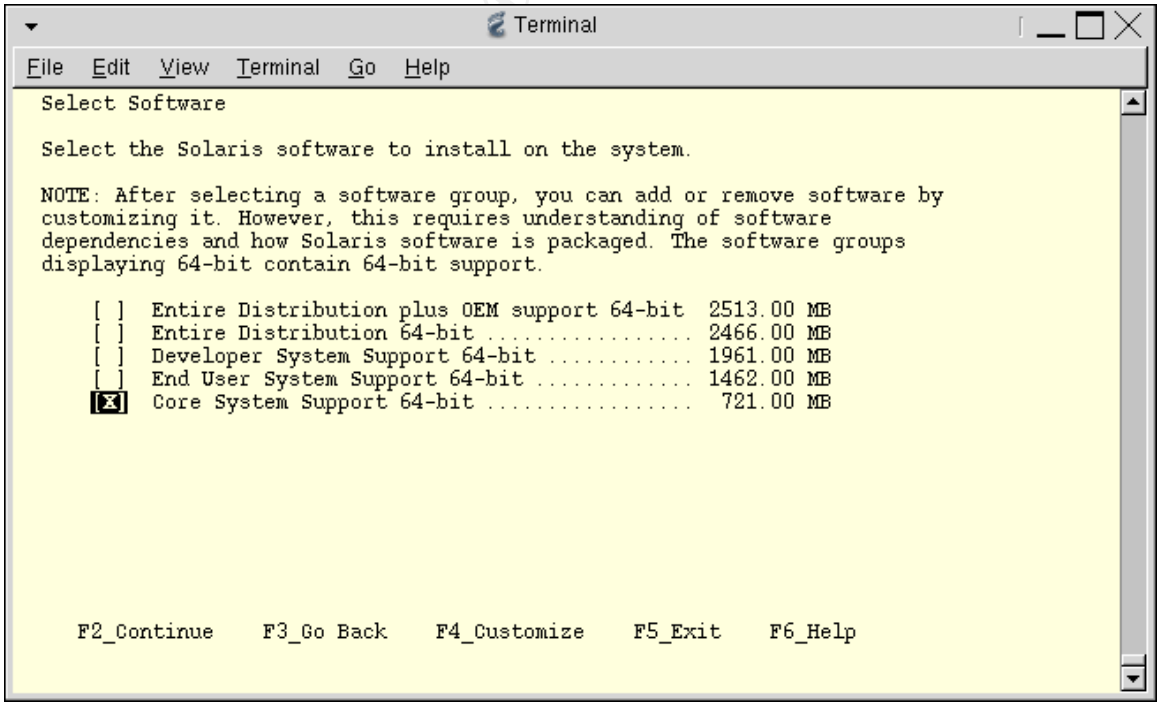
29.Select any additional Geographic Regions for which support should be installed. Usually, none are necessary. Select **F2_Continue**.



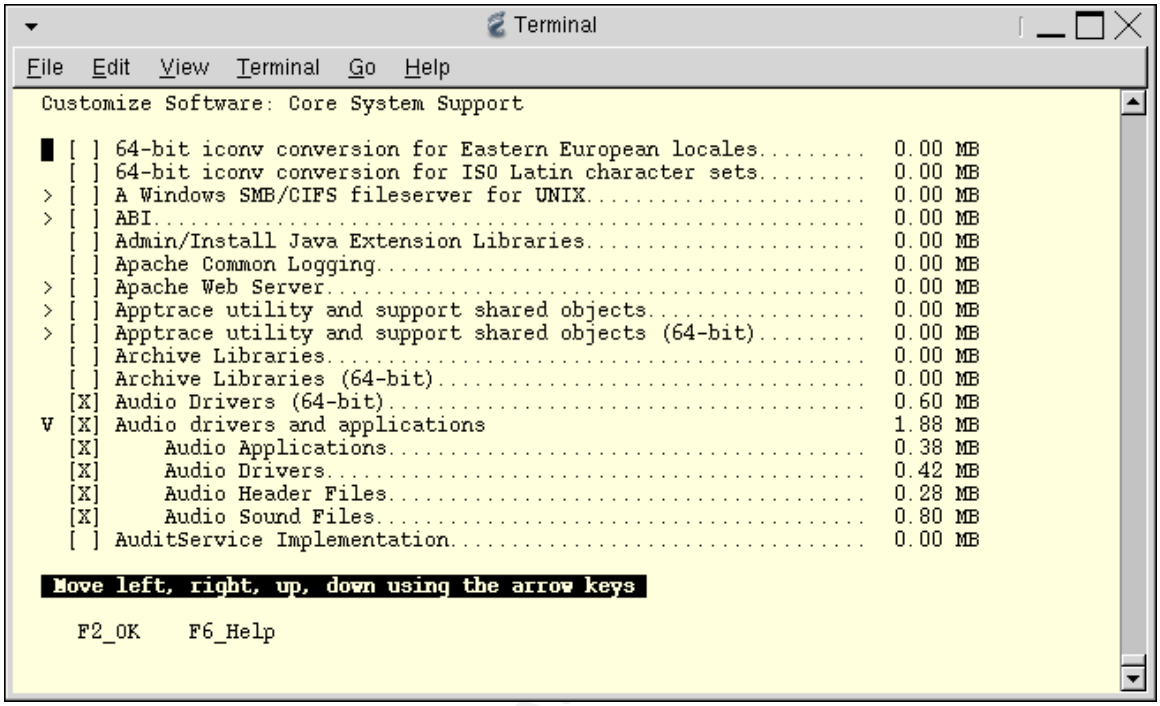
30.Select 64-bit Support and **`F2_Continue`**.



31.Select "Core System Support 64-bit" and **`F4_Customize`**.



32. The Customize Software Screen allows you to add and remove individual software components of the "Core System Support" software group. The installer displays an ASCII menu with four columns (see screen capture).



The first column works like an inverted tree. A ">" represents an unexpanded cluster of packages, a "V" represents an expanded cluster of packages, and no symbol represents a single package. You can expand and collapse a cluster by using the arrow keys to place the cursor over the ">" or "V" respectively and pressing <Enter>.

The second column shows which clusters or packages are selected. No symbol means that no cluster or package is selected. An "X" shows that a cluster or package is selected. A "!" indicates that the cluster or package is required, and may not be removed. Finally, a "/" indicates a partially selected cluster. Clusters and packages may be selected or unselected by placing the cursor over the brackets and pressing <Enter>.

The third column is the cluster or package name. Highlighting the name and pressing <Enter> will display a detailed description.

The fourth column displays the size of the cluster or package. A size of 0.00 MB indicates the cluster or package is not selected.

Because we have selected the "Core System Support" software group, which does not include the Java Virtual Machine, the WebStart Installer cannot automatically install the Solaris Software 2/2 CDROM. So, we need not concern ourselves with selecting any clusters or packages that are part of the second disk. We will install those packages manually after the initial install is complete.

Now we can start to customize our installation by making the following changes:

Cluster/Package Additions

SUNWzlib	- Zip Compression Library (required by SSH)
SUNWnptr	- NTP Daemon (root)
SUNWntpu	- NTP (var)
SUNWmdr	- Solaris Volume Manager (root)
SUNWmdu	- Solaris Volume Manager (usr)
SUNWmdx	- Solaris Volume Manager Drivers
SUNWlibC	- SunWorkShop compilers libC required by system tools
SUNWadmfw	- System and Network Administrative Framework
SUNWadmc	- System Administration Core Libraries
SUNWtcpd	- TCPD access control (TCP Wrappers)

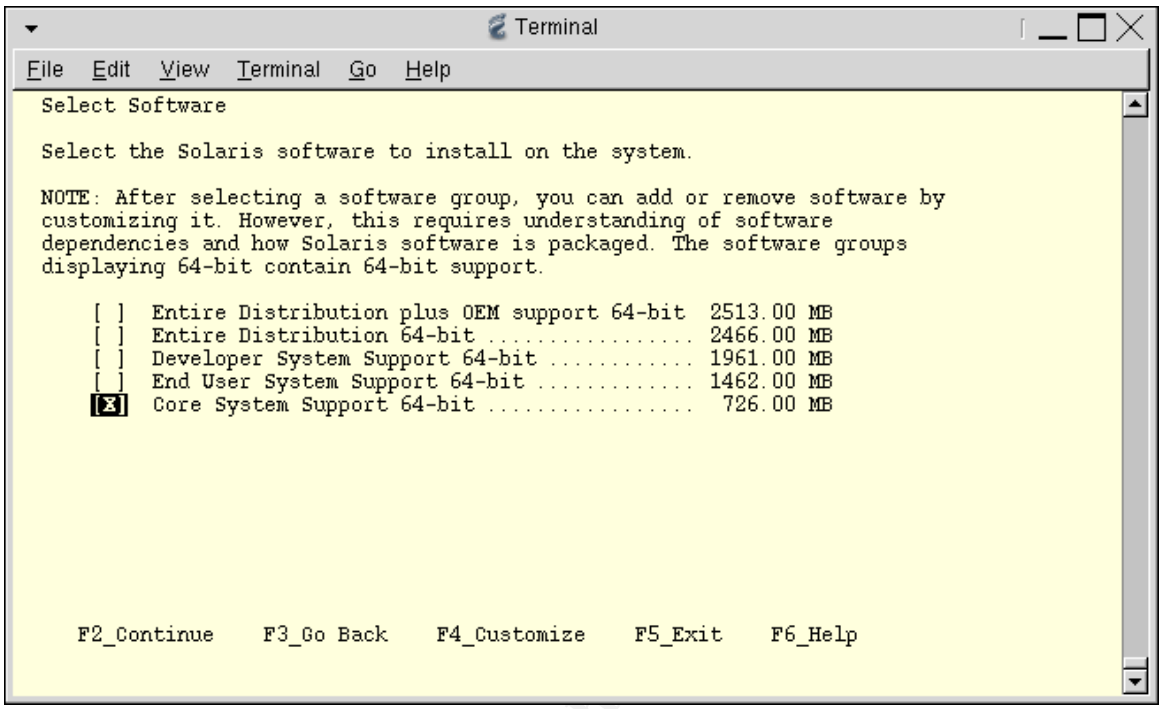
Cluster/Package Deletions

SUNWatfsr	- AutoFS (root)
SUNWatfsu	- AutoFS (usr)
SUNWftpr	- FTP Server Configuration Files
SUNWftpu	- FTP Server and Utilities
SUNWkrbr	- Kerberos Support (root)
SUNWkrbu	- Kerberos Support (usr)
SUNWlldap	- LDAP development libraries
SUNWCsndm	- Sendmail Support
SUNWnisr	- NIS Client (root)
SUNWnisu	- NIS Client (usr)
SUNWdtcor	- Solaris Desktop /usr/dt File System
SUNWinamd	- Internet Domain Name Server
SUNWrcmdr	- Remote Network Server Commands (root)
SUNWrcmdu	- Remote Network Server Commands (usr)
SUNWtnetr	- Telnet Server (root)
SUNWtnetu	- Telnet Server (usr)
SUNWtnamr	- Trivial Name Server (root)
SUNWtnamd	- Trivial Name Server (usr)

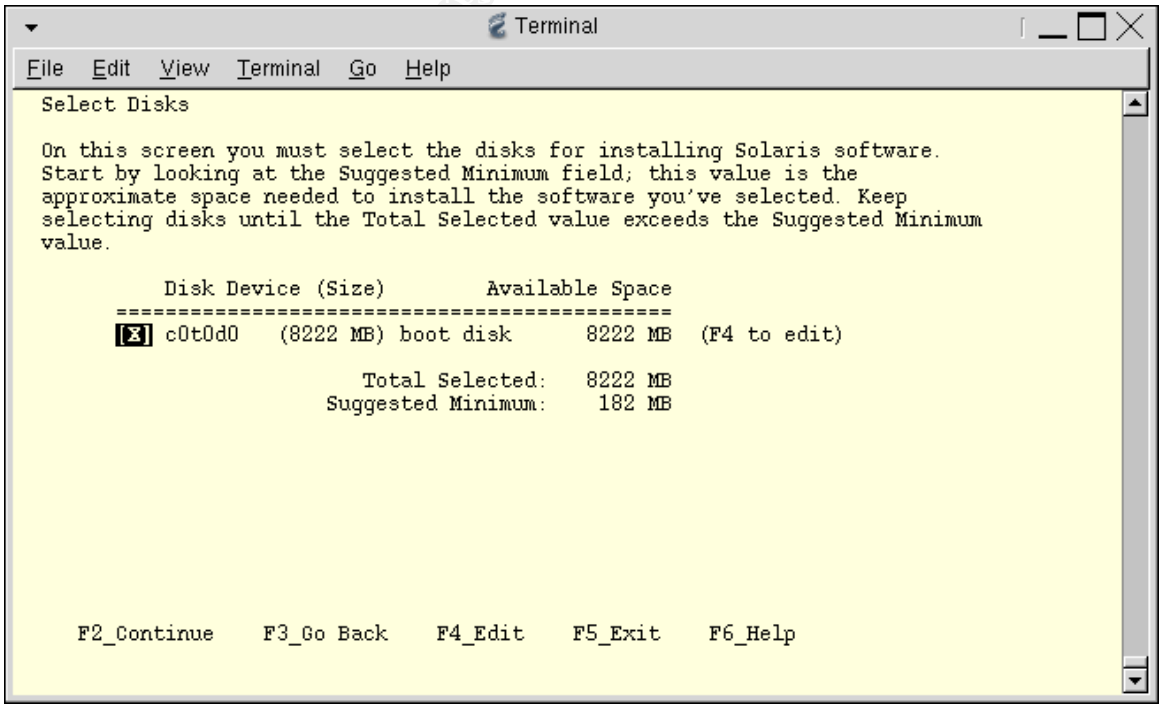
As many packages have been removed as possible. The above lists should be fairly self-explanatory. Some security tools have been added, such as Zlib for SSH, TCPD for access control of network services, and NTP for network time synchronization. Deletions include vulnerable network services such as FTP, Telnet, AutoFS, NIS, and LDAP. Note that some hardware platforms may require additional clusters or patches.

Select **F2_OK** to exit the customization screen and return to the "Select Software" screen.

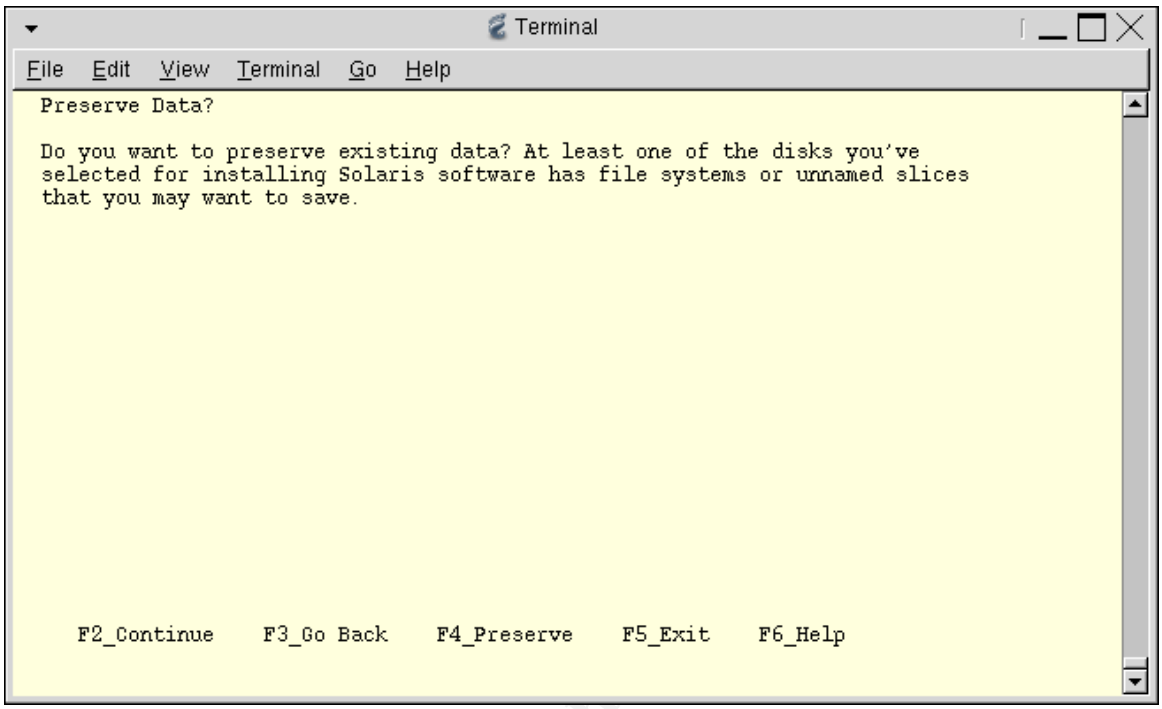
33. At the “Select Software” screen, choose **`F2_Continue`**.



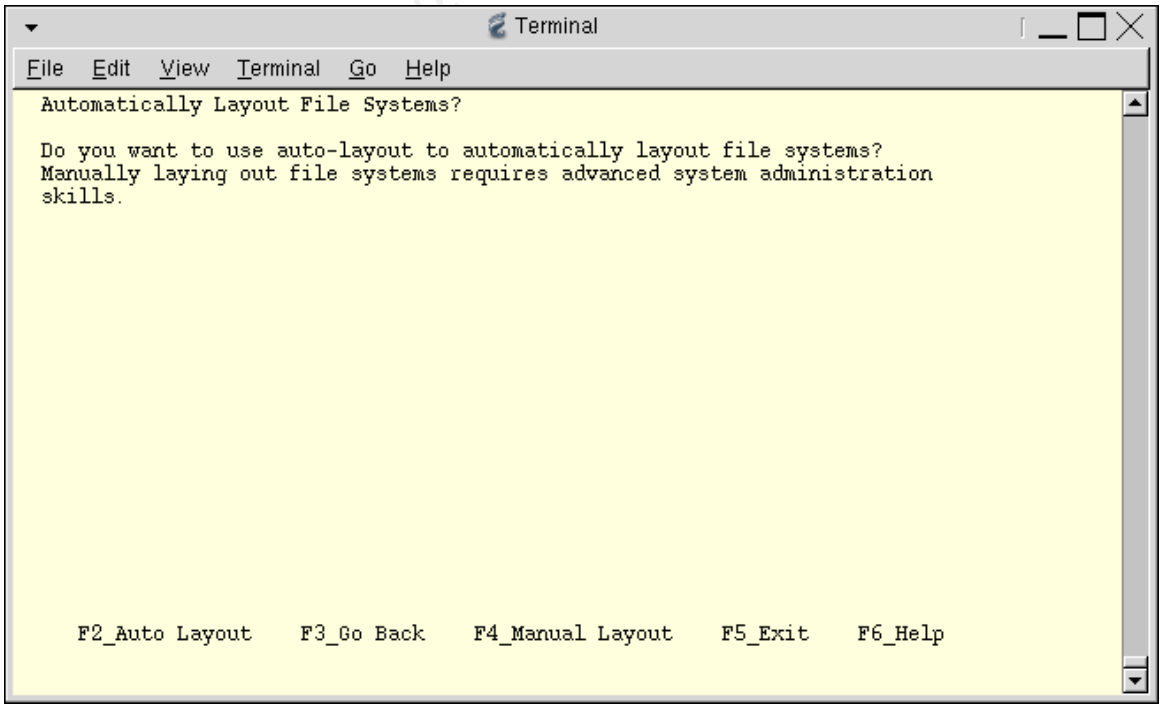
34. Select the disks that will be for the installation. More than one disk may be selected. In our case, only one disk is present. Select **`F2_Continue`**.



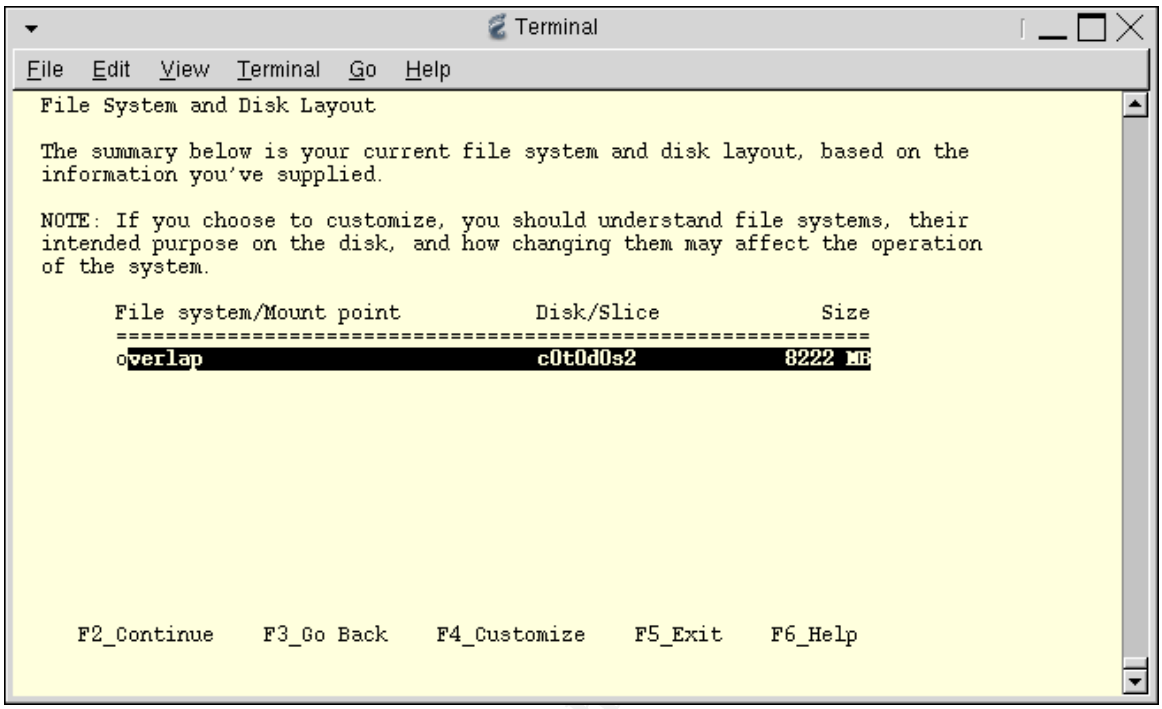
35. No existing data (partitions) should be preserved. Select **`F2_Continue`**.



36. We have come to one of the greatest debates in UNIX Administration, disk partitioning. Most administrators prefer greater control over file system layout and will want to select **`F4_Manual Layout`**.



37.Select the appropriate disk (if there is more than one) and **`F4_Customize`**.

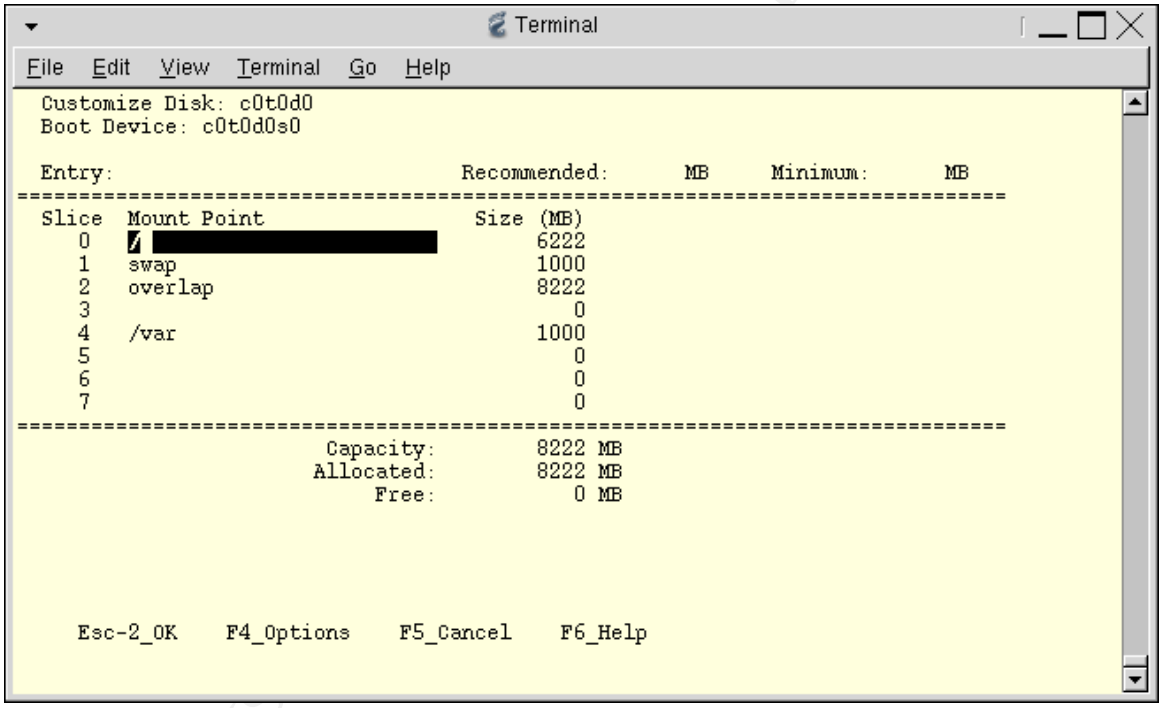


38. By using the arrow keys, you can navigate through the ASCII menu to layout the file system. We prefer to keep all of the core partitions under / and only break out /var separately. However, it is certainly worth noting that many administrators may not want to do this.

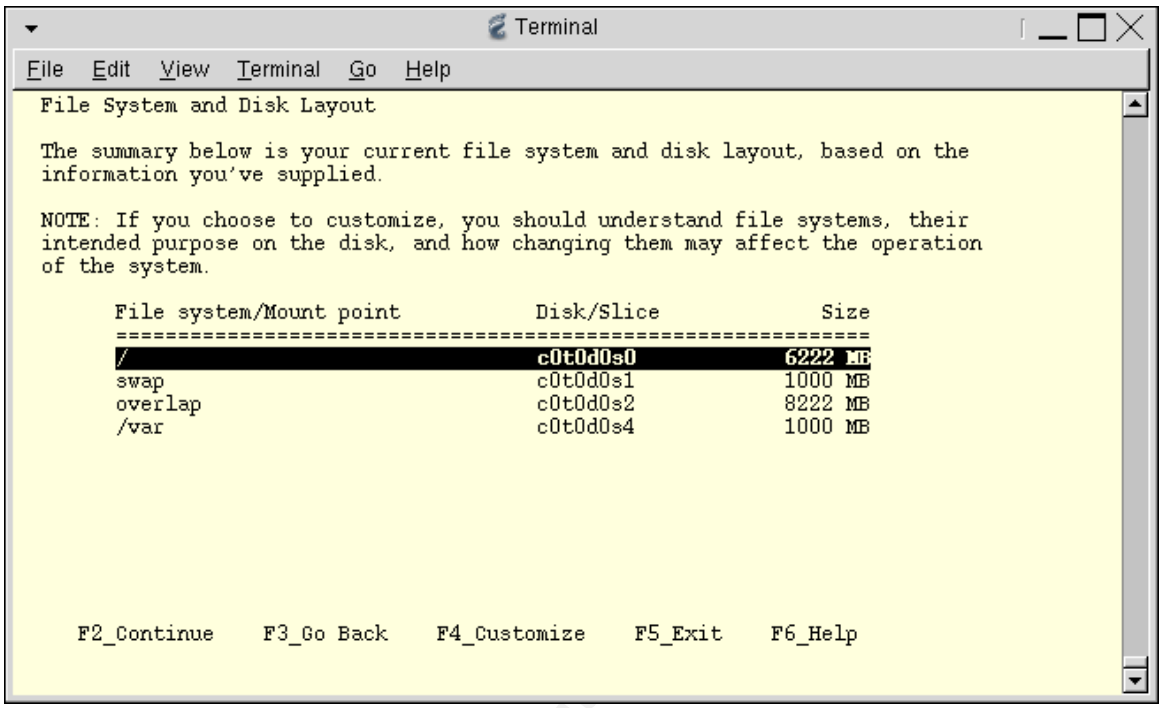
    A good balance between security and convenience must be obtained. And since our servers are for internal use only, convenience outweighed security. Conversely, it is often recommended that /usr and /opt be separate partitions so that they may be mounted read-only, especially when providing external facing services. Bottom line, site-specific needs and personal preference go a long way here, so make sure you consider your partitioning scheme very carefully.

    No matter what you decide, try to pick a standard scheme that can be used for most configurations. This will make setting up JumpStart much easier.

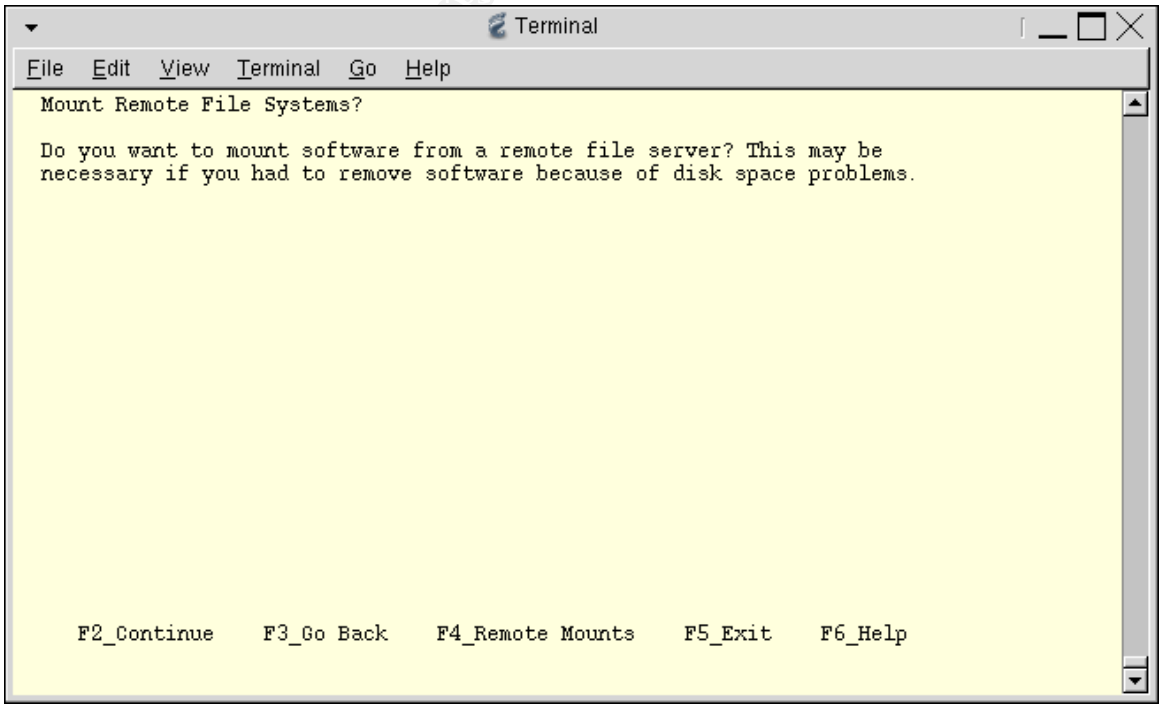
    Use **`Esc-2_OK`** to exit.



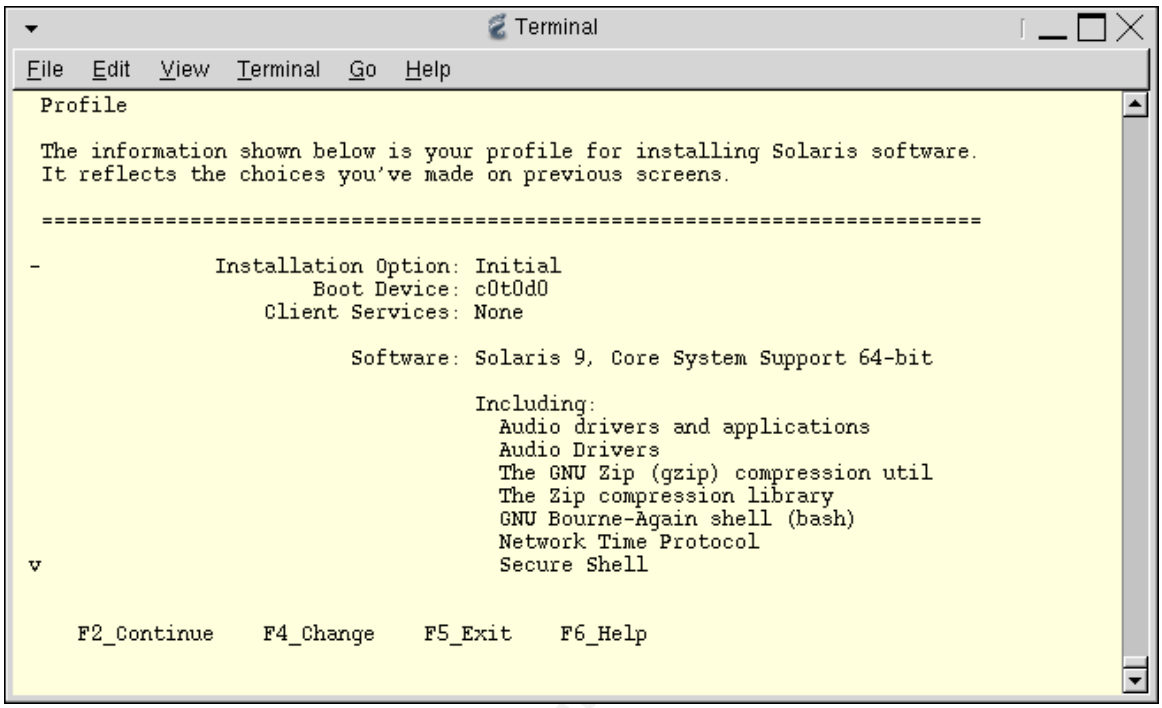
39. The "File System and Disk Layout" screen provides a summary of the current disk and file system layouts. Select **`F2_Continue`**.



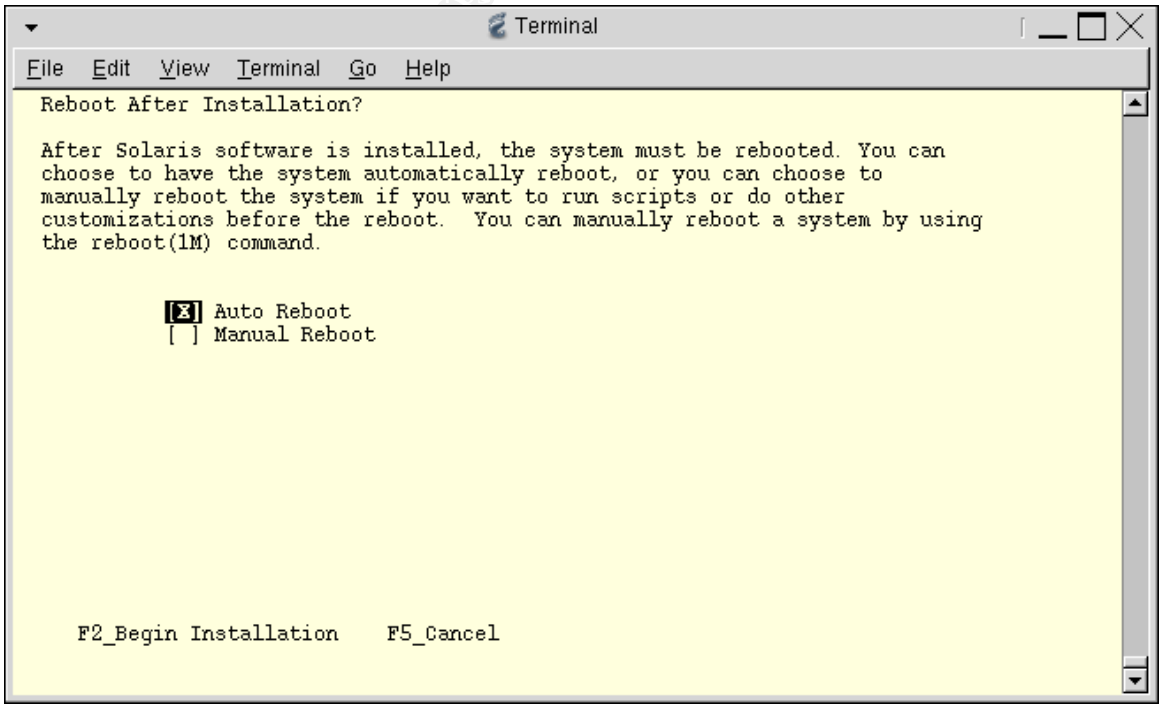
40. We do not want to mount any remote (NFS) file systems. Select **`F2_Continue`**.



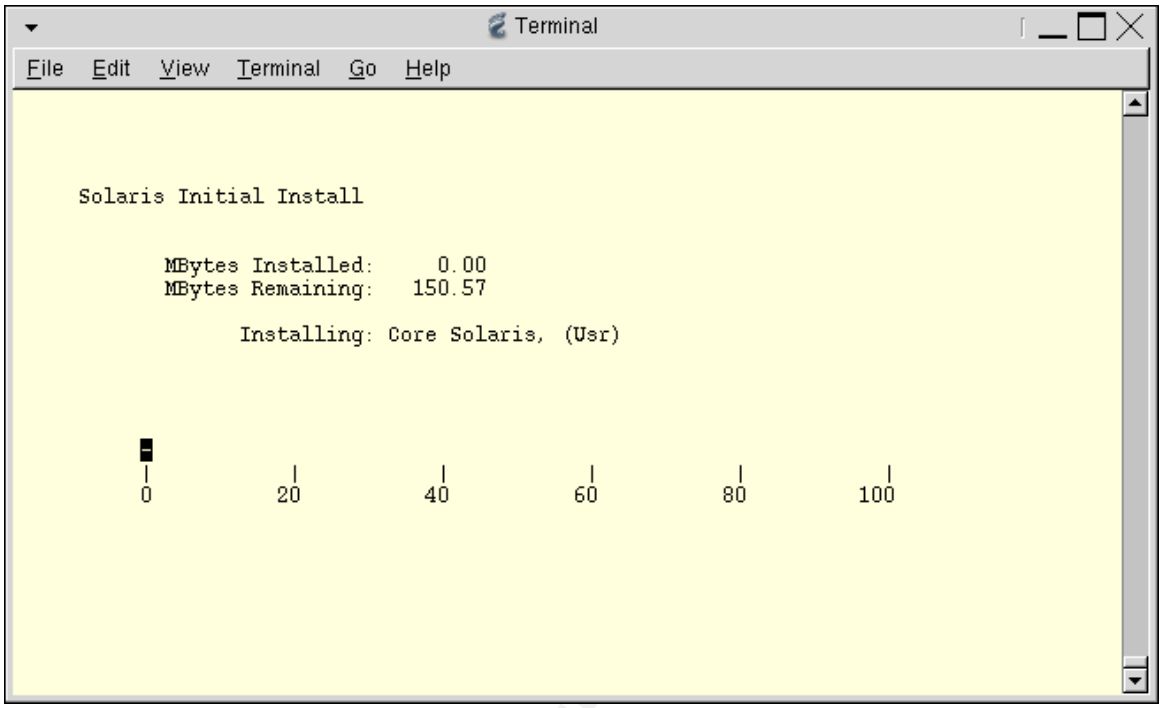
41. The “Profile” screen summarizes your installation profile. Verify for correctness, and **`F2_Continue`**.



42. Select “Auto Reboot” after the installation, and **`F2_Continue`**.



43. The installation will begin and its progress will be displayed. The system will reboot once the installation of the Solaris 9 Software 1/2 CDROM is complete.



44. Because the WebStart Installer cannot run, the root password will not be set. Once the JumpStart server has rebooted, log in as root and set the root password using the **passwd** command.

45. Now we can proceed with the package installation from the Solaris 9 Software 2/2 CDROM. Start by ejecting the Solaris 9 Software 1/2 CDROM from the CDROM drive:

```
# eject cdrom
```

Load the Solaris 9 Software 2/2 CDROM into the CDROM drive and mount the it using the following command (Note: In most cases the CDROM drive will be /dev/dsk/c0t6d0s0. However IDE systems may be different):

```
# mount -F hsfs -o ro /dev/dsk/c0t2d0s0 /mnt
```

Change to the package directory:

```
# cd /mnt/Solaris_9/Product
```

Install the following additional packages, using the command:

```
# pkgadd -d . package1 [package2 package3 ...]
```

Additional Packages

SUNWaccr - System Accounting (root)
SUNWaccu - System Accounting (usr)
SUNWbash - GNU Bash Shell (required for Gzip)
SUNWgcmn - Common GNU Packages (required for GNU software)

SUNWgtar	- GNU Tar Utility
SUNWgzip	- Gzip Compress Utility
SUNWsshcu	- SSH Common Utilities
SUNWsshr	- SSH Client and Utilities (root)
SUNWsshu	- SSH Client and Utilities (usr)
SUNWsshdr	- SSH Server (root)
SUNWsshdu	- SSH Server (usr)
SUNWzip	- Zip Compression Utility
SUNWzlibx	- Zlib Compression Library (64-bit)

After all of the above packages have been added, the Solaris Software 2/2 CDROM can be unmounted:

```
# cd /
# umount /mnt
```

At this point the CDROM may be safely removed, and the system should be rebooted:

```
# init 6
```

46. Next, the latest Solaris 9 patch cluster should be downloaded from http://sunsolve.sun.com to a secure intermediate location and burned to a CDROM. The CDROM can then be mounted on the JumpStart server to install the patches:

```
# cp /mnt/9_Recommended.zip /tmp
# cd /tmp
# unzip 9_Recommended.zip
# cd 9_Recommended
# ./install_cluster
```

Since this is a minimal installation, you will see many errors during the patch process. Most of the errors should be "return code 8", which indicates that the target packages are not installed, and can safely be ignored. "return code 2" indicates the patch has already been applied and can be ignored as well. After the patch cluster has been successfully installed, the CDROM can be unmounted and removed, then server should be rebooted.

47. First, /etc/inetd.conf needs to be modified to remove any unnecessary services. Make a copy of the original inetd.conf:

```
# cp /etc/inetd.conf /etc/inetd.conf.orig
```

Remove all services except TFTP:

```
# echo "Only Services for JumpStart" > /etc/inetd.conf
# echo "tftp dgram udp6 wait root /usr/sbin/in.tftpd
in.tftpd -s /tftpboot" >> /etc/inetd.conf
```

48. By default, TCP Wrappers are not enabled. Will enable them with these commands:

```
# echo "ENABLE_CONNECTION_LOGGING=YES" \
> >> /etc/default/inetd
# echo "ENABLE_TCPWRAPPERS=YES" >> /etc/default/inetd
```

Now, we configure TCP Wrappers to accept TFP and SSH connections on the private network by

creating the /etc/hosts.allow file with the following commands:

```
# echo "in.tftpd: 192.168.0.0" > /etc/hosts.allow
# echo "in.sshd: 192.168.0.0" >> /etc/hosts.allow
```

Next, we create the /etc/hosts.deny file to block all other services from any host except the localhost:

```
# echo "ALL: ALL EXCEPT localhost : DENY" > /etc/hosts.deny
```

49. Logging of all failed authentication attempts should be enabled by the commands:

```
# echo "SYSLOG_FAILED_LOGINS=0" >> /etc/default/login
# echo "auth.notice    /var/log/authlog" >> \
> /etc/syslog.conf
# touch /var/log/authlog
# chmod 600 /var/log/authlog
```

Logging of all SU attempts should be enabled by the commands:

```
# touch /var/log/sulog
# chmod 600 /var/log/sulog
```

50. A default banner message should be configured using the command:

```
# echo "Authorized Uses Only." > /etc/issue
```

Next, make sure SSH displays the banner message with the following command:

```
# echo "Banner  /etc/issue" >> /etc/ssh/sshd_config
```

51. NTP should be configured by creating the following /etc/inet/ntp.conf file and creating the drift file.

```
# cat > /etc/inet/ntp.conf <<EOF
restrict default nomodify
restrict 192.168.0.5
restrict 192.168.0.6
restrict 127.0.0.1
server 192.168.0.5
server 192.168.0.6
driftfile /var/ntp/ntp.drift
EOF
# touch /var/ntp/ntp.drift
```

Start NTP by using the following command:

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

52. We need to disable any unnecessary services that are started at boot time in the /etc/rc2.d directory. (Note: RPC is required for bootp, a necessary component of the JumpStart server. However, RPC will be disabled on all clients.)

```
# cd /etc/rc2.d
# rm S71ldap.client S73cachefs.daemon S76nscd \
> S89PRESERVE S93cacheos.finish
```

53. Finally, we create an administrator user account that can used to connect via SSH and if necessary, SU to root:

```
# useradd -g 10 -d /export/home/admin -m -c "Admin User" \
> -s /usr/bin/bash admin
```

Be sure and set a password for the admin user:

```
# passwd admin
New Password:
Re-enter new Password:
passwd: password successfully changed for admin
```

54. Last, reboot the server so that all the above changes will take effect.

```
# init 6
```

Installation (JumpStart Server)

JumpStart consists of three core components, installation, identification, and configuration. The installation component contains all of the installation media. The identification component holds individual host client information, such as: IP address, host name, etc. The configuration component mandates how JumpStart behaves. All of these most components are shared over NFS so that JumpStart clients can access this information during installation. We will organize each of the components into a separate directory.

Installation Component:
```
# mkdir /export/install
```

Identification Component:
```
# mkdir /export/sysidcfg
```

Configuration Component:
```
# mkdir /export/jumpstart
```

To share these components, we add the following line to the /etc/dfs/dfstab. The "-o ro=*@IP address"* option restricts NFS to be read-only to a specific IP address or range of addresses. (Note: The anon=0 is required for JumpStart to operate properly).

```
# cat > /etc/dfs/dfstab <<EOF
share -F nfs -o ro=@192.168.0.0,anon=0 /export/install
share -F nfs -o ro=@192.168.0.0,anon=0 /export/jumpstart
share -F nfs -o ro=@192.168.0.0,anon=0 /export/sysidcfg
EOF
```

The following command will start sharing the directories:
```
# shareall
```

In case we decide to add additional Solaris versions in the future, we will create a sparc9 directory to hold the installation files in /export/install, and a sparc9 directory to hold the configuration files in /export/jumpstart.
```
# mkdir /export/jumpstart/sparc9 /export/install/sparc9
```

The next step is to install the installation media on the JumpStart server in the /export/install/sparc9 directory. Insert and mount the Solaris 9 Software1/2 CDROM, as we have previously.
```
# cd /mnt/Solaris_9/Tools
# ./setup_install_server /export/install/sparc9
```

Next, copy the JumpStart configuration check tool and sample JumpStart rules files to the configuration directory. The check tool verifies any rules that are created to define a JumpStart session, and the rules file contains the JumpStart session definitions.
```
# cd ../Misc/jumpstart_sample
# cp check /export/jumpstart
# cp rules /export/jumpstart
```

Now, unmount and eject the Solaris 9 Software 1/2 CDROM, and insert and mount the Solaris 9

Software 2/2 CDROM.

```
# cd /mnt/Solaris_9/Tools
# ./add_to_install_server /export/install/sparc9
```

Once the operation is complete, unmount and eject the Solaris 9 Software 2/2 CDROM.

Now, we need to set up some basic configuration files for JumpStart. These configuration files are specified in the /export/jumpstart/rules file (see Appendix A for more information the rules file). There are three types of configuration files that can used in the rules file: begin scripts, profiles, and finish scripts.

- Begin Scripts – An optional Bourne shell script that is executed before installation begins. This can be useful for copying configuration files to the client before installation, but in reality, it is seldom used. Our implementation of JumpStart does not require the use of begin scripts.
- Profiles – A text file that is used as a template by JumpStart, which defines how to install the Solaris operating system.
- Finish Script – An optional Bourne script that is executed after installation is complete. This is useful for installing patches, third-party software, and updating configuration files.

Because different Sun hardware may have different sized hard drives we will create three different profiles (Appendix B), one profile for drives < 18 GB (small.profile), one profile for drives > 18 GB (default.profile), and one for drives > 36 GB (large.profile). Each profile specifies the:

- install_type – upgrade, flash, or initial
- geo – geographic location
- locale – language set
- cluster – software cluster
- partitioning – defines whether or not to use auto-sizing
- filesys – defines each partition's location and size
- package – adds or removes additional packages (we use the same additional packages we specified for the JumpStart server, except for any components that are unique requirements for JumpStart).

Now that we have defined the basic installation, we need to add a finish script to perform the hardening and configuration. But first, we need to add some directories to the /export/jumpstart/sparc9 directory to help keep everything organized. We will create a “files” directory to hold JumpStart client configuration files, a “software” directory to hold third-party applications, and a “patches” directory to place an uncompressed copy of the latest Solaris 9 patch cluster.

```
# mkdir /export/jumpstart/files \
> /export/jumpstart/software \
> /export/jumpstart/patches
```

The finish script (see Appendix C for detailed information) requires some trial and error to get perfect. The most import item to remember is that when the finish script is executed after the installation, the root of the newly installed drive is mounted on /a not /. So, all modifications, package

installations, and patch installation must be modified to use /a as the base directory. We created two different finish scripts. One performs disk mirroring (default-end.sh), and the other does not (nomirror-end.sh).

Installation (JumpStart Client)

Each JumpStart client requires a "sysidcfg" file. This tells JumpStart about the basic identification information needed to install a client. JumpStart looks for a file named "sysidcfg" in a location that you provide when setting up the JumpStart client. We have found the best way to handle this is to make a new directory for each JumpStart client in the /export/sysidcfg directory. The new directory is named after the *hostname* of the client, and contains a single sysidcfg file for that specific client. sysidcfg files contain the following information (an example can be found in Appendix D):

- system_local – language set
- name_service – type of naming service to use
- network_interface – network parameters
- security_policy – security policy type
- root_password – the root password in hash form
- timezone – default timezone
- timehost – where to set the from
- terminal – terminal type

The next steps may be performed in variety of ways. We have found the setup of jumpstart clients to be a manual and tedious procedure. So, we created a script called "/export/jumpstart/setup-client" (Appendix E) to automate the process. It is possible to do the set up manually, using the /export/install/sparc9/Solaris_9/Tools/add_install_client script (see the MAN page for detailed instructions on its usage). To help you understand the process, here is the procedure that the setup-client script performs:

1. setup-client accepts a number of arguments to determine what type of installation should be performed. The syntax is as follows:

   setup-client [-n] [-s | -l] <hostname> <IP_Addr> <MAC_Addr>
   -n = nomirror: Turns off automatic root disk mirroring (optional)
   -s | -l = small | large: Disk size/layout
   hostname = Desired hostname of the new client
   IP_Addr = Desired IP address of the new client (must be 192.168.0.x)
   MAC_Addr = MAC address for the primary interface of the new client

2. First, setup-client checks to see if any of the disk parameters have been entered, if not then defaults of "mirroring enabled" and "default disk size" are taken.
3. Then setup-client determines if this JumpStart client has any previous configuration entries that might conflict with its operation. If so, setup-client issues an error requesting that the problem be fixed. Otherwise, execution continues.
4. setup-client adds an entry to /etc/hosts for the JumpStart client.
5. setup-client adds an entry to /etc/ethers for the JumpStart client.
6. Now, setup-client creates a sysidcfg file for the JumpStart client in /export/sysidcfg/*hostname*/sysidcfg.
7. Next, setup-client adds a rule to the /export/jumpstart/rules file based on the disk layout parameters specified.
8. setup-client uses /export/jumpstart/check to verify and commit the rule file.

9. Then, setup-client performs the /export/install/sparc9/Solaris_9/Tools/add_install_client command to set up the /tftpboot network boot image and the /etc/bootparams file.
10. Finally, setup-client logs the successful completion and date to /export/jumpstart/client.log

Now, we can use the setup-client tool to add our JumpStart client to the server.

```
# cd /export/jumpstart
# ./setup-client -n -s server1 192.168.0.20 0:3:ba:27:b5:6f
```

Finally, The JumpStart client is ready for installation. To proceed with the installation, we attach the client to the private JumpStart network. Then we power on the client. After the initial banner message, we issue a break using **stop-a** and at the OK prompt we issue the following command:

```
ok boot net – install
```

The installation will now proceed unattended through completion. Installation times vary from about thirty minutes to several hours, depending on the speed of the client.

Testing

Before the JumpStart server can be used safely, we need to conduct some tests to verify its security.

1. SSH as the "admin" user to the JumpStart server from a host on the private network (the only network allowed by TCP Wrappers to SSH to our server). We should be able to successfully connect.

```
# ssh admin@192.168.0.10
admin's password:
Authentication Successful.
...
```

2. SSH as the "admin" user to the JumpStart server from a host not on the private network. TCP Wrappers should not allow you to connect.

```
# ssh admin@192.168.0.10
warning: Authentication failed.
Disconnected (local); connection lost (Connection
closed by remote host.).
```

3. SSH as "root" to the JumpStart server from a host on the private network. Root log in is disallowed; you should not be able to connect.

```
# ssh root@192.168.0.10
admin's password:
admin's password:
admin's password:
Disconnected; no more authentication methods available
(No further authentication methods available.).
```

4. Make several botched log in attempts, then examine the /var/log/authlog to ensure that failed log ins are being logged. For example:

```
Aug 3 08:20:55 192.168.0.10 sshd[5801]: [ID 702911
auth.warning] Wrong password given for user 'admin'.
```

5. SU from the "admin" user to "root". Then, SU from the "admin" user to "root", and intentionally botch the password. Check the /var/log/sulog to ensure that both successful and unsuccessful SU attempts are logged.
   Successful:

```
SU 08/03 08:24 + pts/5 admin-root
```

   Unsuccessful:

```
SU 08/03 08:25 - pts/5 admin-root
```

6. Examine the output of the **mount** command to ensure that all files systems are logging. Look for the "logging" attribute for each mounted filesystem partition. For example:

```
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/
largefiles/logging/xattr...
```

7. Perform a **ps -ef** and examine all running processes. Be sure that nothing is out of place or unaccounted for. Look for the following processes, and investigate discrepancies:

```
sched
/etc/init -
pageout
fsflush
/usr/lib/saf/sac -t 300
/usr/sbin/auditd
/usr/sbin/cron
/usr/lib/sysevent/syseventd
/usr/lib/picl/picld
/usr/sbin/rpcbind
/usr/sbin/inetd -s
/usr/lib/inet/xntpd
/usr/lib/nfs/lockd
/usr/lib/nfs/statd
/usr/sbin/syslogd -t
/usr/sbin/nscd
/usr/lib/utmpd
/usr/sbin/in.rarpd -a
/usr/lib/saf/ttymon
/usr/lib/nfs/nfsd
/usr/sbin/rpc.bootparamd
/usr/lib/nfs/mountd
/usr/local/sbin/sshd2
```

8. Use NMAP to scan the JumpStart server for open ports. Any ports other than those listed below need to be investigated and secured.

```
# nmap 192.168.0.10

Starting nmap 3.50
Interesting ports on jumpstart (192.168.0.10):
The 1657 ports scanned but not shown below are in
state: closed)
PORT     STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
2049/tcp  open  nfs
4045/tcp  open  lockd
32771/tcp open  sometimes-rpc5
32778/tcp open  sometimes-rpc19
```

9. Use Nessus to perform more detailed vulnerability scans. After scanning our JumpStart server, there were no unexpected abnormalities. RPC and NFS are of the greatest risk. But, our exposure is limited by the firewall and private network.

Host	Port/Service	Issue
192.168.0.10	ssh (22/tcp)	Security Notes
192.168.0.10	rpc (111/tcp)	Security Warning
192.168.0.10	nfs (2049/tcp)	Security Warning
192.168.0.10	general icmp	Security Warning
192.168.0.10	general tcp	Security Notes
192.168.0.10	general tcp	Security Warning

10. Test that NFS is restricted to the private network. First, use a host on the private network to mount a NFS share. This should be successful for read-only mounts. Then try to mount a NFS share from a host not on the private network. You should not be able to mount a share.
    Private:

```
# mount 192.168.0.10:/export/install /mnt
#
```

    Non-Private:

```
# mount 192.168.0.10:/export/install /mnt
nfs mount: 192.168.0.10:/export/install: Permission
denied
```

11. Last, verify that NTP is working correctly by executing the following command (the output should be similar that which is displayed below):

```
# ntptrace
localhost: stratum 16, offset 0.000073, synch distance
1.00066
192.168.0.5: stratum 3, offset 0.000026, synch
distance 1.00086
192.168.0.6: stratum 3, offset 0.000079, synch
distance 1.00122
```

Ongoing Maintenance

1. Backups

   As mentioned earlier, backups will not be performed on a regularly scheduled basis. However, backups will be performed whenever changes are made to the JumpStart server, such as, after applying patches or making configuration changes. We will make an initial backup of each mounted UFS filesystem using TAR, and burn the the tar files to a CDROM. Thereafter, the same procedure for backups will be followed after any patching or configuration changes.

   ```
   # tar cvf root.`date`.tar /
   # tar cvf var.`date`.tar /var
   ```

2. Log Rotation

   Log rotation is an important part of administration. Log files must be rotated to conserve disk space; ever growing log files can consume a partition. Rotating log files also makes it easier to search through the log files when you are looking for something specific. Log rotation in Solaris 9 is much improved over previous versions. It is now accomplished by a tool called LOGADM. LOGADM should already be set up to run from cron nightly as part of the default Solaris installation. By default, the configuration file, /etc/logadm.conf, contains entries for syslog, messages, cron, lpsched, and process accounting. You will probably want to make additions to these defaults. For example, to rotate /var/log/authlog:

   ```
   # echo "/var/log/authlog -C 8 -P '`date`' -a 'kill -HUP
   `cat /var/run/syslog.pid`'"
   ```

3. Log Monitoring

   Log monitoring can be accomplished in a multitude of ways, from manual observations to complex scripts and utilities. Programs such as LogCheck http://sourceforge.net/projects/sentrytools/ can monitor log files and email an administrator when anything suspicious is noted. Also, many administrators prefer to send all syslog entries to a remote syslog server, so they can monitor everything centrally. While others, prefer to keep all the logging local. There are pros and cons to all of these options.

   For simplicity, we will keep all the logging local to the JumpStart server. Weekly, we will manually review the syslog, messages, and authlog files for any anomalies.

4. Patching

   Patching is one of the single most important proactive things that you can do to help prevent problems and attacks. There are several methods available for patching Solaris servers. First, Patch Manager, from Sun, can examine your system, find out what patches are available / needed, and automatically apply the patches. Second, PatchDiag, from Sun, can also identify which patches are available / needed, but will not automatically install them. Third, Sun provides patch clusters of the most commonly needed patches and all critical security patches for download from http://sunsolve.sun.com.

   Again, for simplicity, we will download the newest patch cluster on the first of each month and install it manually. Installation instructions can be found in the README file of the patch cluster.

5. Password Maintenance

   Passwords are one of the weakest links in the security chain. A good password scheme is dependent on choosing adequately strong passwords, keeping passwords secrete, as well as changing passwords often.

   Choosing strong passwords is not difficult, but often over-looked. Secure passwords can be generated by a random password generator, such as http://www.multicians.org/thvv/gpw.html. Or, a simple recipe, such as:

   A. Take a phrase such as: I have a yellow dog named sam.
   B. Take the first letter of each word: Ihaydns
   C. Change some of the letters to numbers and mix capitalization: !hAydn5

   We will use a password generator to create our strong passwords.

   Passwords must be kept secrete to remain effective. That means that only trusted administrators should have access to the root password. Do not send passwords in email, unless encrypted. Do not write passwords down... no sticky notes taped to the underside of your keyboard!

   Passwords must be changed regularly. No matter how hard you try, passwords will "leak" out. We will change our passwords monthly. Making password maintenance part of your patching routine is a good way to remember.

6. Permissions

   Many systems files have permissions that are not strict enough, possibly allowing an attacker to modify critical system files, or insert malicious code. Tools, such as Fix-Modes http://mrtg.stanford.edu/SECURITY/Scripts/fix-modes/, should be used regularly to insure that file permissions are kept secure. Be sure to run Fix-Modes after any patches applied, as the patches may reset file permissions.

   Additional tools, such as TripWire http://www.tripwire.com/products/tripwire_asr/, can be used with cron to constantly monitor for any file changes and send an email if changes occur. Initially, TripWire creates a baseline MD5 checksum of each file. Each time TripWire is run, it recomputes the checksum of each and compares it with the baseline checksum. If they do not match, an alert is sent via email.

7. General Monitoring

   General monitoring is useful for keeps tabs on things like disk utilization, CPU utilization, swap file usage, memory utilization, etc. We use a proprietary software package from Concord http://www.concord.com to monitor our systems over the network via SNMP. It's installation is beyond the scope of this document. However, there are free solutions as well. SRS Net Connect http://srsnetconnect.sun.com from Sun provides basic system monitoring and email alerts at no charge. If you have Sun support, it will also analyze your systems on a monthly basis for security and configuration vulnerabilities.

Appendix A - rules

```
#
#       @(#)rules 1.12 94/07/27 SMI
#
# The rules file is a text file used to create the rules.ok file for
# a custom JumpStart installation. The rules file is a lookup table
# consisting of one or more rules that define matches between system
# attributes and profiles.
#
# This example rules file contains:
#   o syntax of a rule used in the rules file
#   o rule_keyword and rule_value descriptions
#   o rule examples
#
# See the installation manual for a complete description of the rules file.
#
#
##############################################################################
#
# RULE SYNTAX:
#
# [!]rule_keyword rule_value [&& [!]rule_keyword rule_value]... begin profile finish
#
#    "[ ]"  indicates an optional expression or field
#    "..."  indicates the preceding expression may be repeated
#     "&&"  used to "logically AND" rule_keyword and rule_value pairs together
#      "!"  indicates negation of the following rule_keyword
#
#  rule_keyword   a predefined keyword that describes a general system
#                 attribute. It is used with the rule_value to match a
#                 system with the same attribute to a profile.
#
#  rule_value     a value that provides the specific system attribute
#                 for the corresponding rule_keyword. A rule_value can
#                 be text or a range of values (NN-MM).
#                 To match a range of values, a system's value must be
#                 greater than or equal to NN and less than or equal to MM.
#
#  begin          a file name of an optional Bourne shell script
#                 that will be executed before the installation begins.
#                 If no begin script exists, you must enter a minus sign (-)
#                 in this field.
#
#  profile        a file name of a text file used as a template by the
#                 custom JumpStart installation software that defines how
#                 to install Solaris on a system.
#
```

```
#  finish        a file name of an optional Bourne shell script
#                that will be executed after the installation completes.
#                If no finish script exists, you must enter a minus sign (-)
#                in this field.
#
# Notes:
# 1. You can add comments after the pound sign (#) anywhere on a line.
# 2. Rules are matched in descending order: first rule through the last rule.
# 3. Rules can be continued to a new line by using the backslash (\) before
#    the carriage return.
# 4. Don't use the "*" character or other shell wildcards, because the rules
#    file is interpreted by a Bourne shell script.
#
#
##############################################################################
#
# RULE_KEYWORD AND RULE_VALUE DESCRIPTIONS
#
#
# rule_keyword    rule_value Type       rule_value Description
# ------------    ---------------       ----------------------
#  any            minus sign (-)        always matches
#  arch           text                  system's architecture type
#  domainname     text                  system's domain name
#  disksize       text range            system's disk size
#                                       disk device name (text)
#                                       disk size (MBytes range)
#  hostname       text                  system's host name
#  installed      text text             system's installed version of Solaris
#                                       disk device name (text)
#                                       OS release (text)
#  karch          text                  system's kernel architecture
#  memsize        range                 system's memory size (MBytes range)
#  model          'text'                system's model number
#  network        text                  system's IP address
#  totaldisk      range                 system's total disk size (MBytes range)
#
#
##############################################################################
#
# RULE EXAMPLES
#
# The following rule matches only one system:
#
#hostname sample_host    -       host_class      set_root_pw
#
# The following rule matches any system that is on the 924.222.43.0 network
# and has the sun4c kernel architecture:
```

```
#    Note: The backslash (\) is used to continue the rule to a new line.
#
#network 924.222.43.0 && \
#       karch sun4c     -       net924_sun4c     -
#
# The following rule matches any sparc system with a c0t3d0 disk that is
# between 400 to 600 MBytes and has Solaris 2.1 installed on it:
#
#arch sparc && \
#         disksize c0t3d0 400-600 && \
#         installed c0t3d0s0 solaris_2.1 - upgrade  -
#
# The following rule matches any system:
#
#any -   -   any_machine  -
#
##############################################################################
#
# BEGIN ACTUAL RULES
hostname server1   -   ./sparc9/small.profile   ./sparc9/nomirror-end.sh
hostname server2   -   ./sparc9/large.profile   ./sparc9/default-end.sh
hostname server3   -   ./sparc9/default.profile   ./sparc9/default-end.sh
```

Appendix B – small.profile

```
install_type    initial_install
geo             N_America
locale          C
cluster         SUNWCcore
partitioning    explicit
filesys         rootdisk.s0     free    /       logging
filesys         rootdisk.s1     1025    swap
filesys         rootdisk.s4     1025    /var    logging
filesys         rootdisk.s5     12      unnamed
package             SUNWzlib    add
package             SUNWzlibx   add
package             SUNWntpr    add
package             SUNWntpu    add
package             SUNWmdr     add
package             SUNWmdu     add
package             SUNWlibC    add
package             SUNWadmfw   add
package             SUNWadmc    add
package             SUNWtcpd    add
package             SUNWaccr    add
package             SUNWaccu    add
package             SUNWbash    add
package             SUNWgcmn    add
package             SUNWgtar    add
package             SUNWgzip    add
package             SUNWzip     add
package             SUNWsshdr   add
package             SUNWsshdu   add
package             SUNWsshr    add
package             SUNWsshu    add
package             SUNWatfsr   delete
package             SUNWatfsu   delete
package             SUNWftpr    delete
package             SUNWkrbr    delete
package             SUNWkrbu    delete
package             SUNWlldap   delete
package             SUNWCsndm   delete
package             SUNWnisr    delete
package             SUNWnisu    delete
package             SUNWdtcor   delete
package             SUNWinamd   delete
package             SUNWrcmdr   delete
package             SUNWrcmdu   delete
package             SUNWtnetr   delete
package             SUNWtnetu   delete
package             SUNWtnamr   delete
```

```
package       SUNWtnamd  delete
package       SUNWtftp   delete
package       SUNWtftpr  delete
```

Appendix B (cont.) – default.profile

```
install_type    initial_install
geo             N_America
locale          C
cluster         SUNWCcore
partitioning    explicit
filesys         rootdisk.s0     free    /       logging
filesys         rootdisk.s1     4100    swap
filesys         rootdisk.s4     4100    /var    logging
filesys         rootdisk.s5     12      unnamed
package             SUNWzlib    add
package             SUNWzlibx   add
package             SUNWntpr    add
package             SUNWntpu    add
package             SUNWmdr     add
package             SUNWmdu     add
package             SUNWlibC    add
package             SUNWadmfw   add
package             SUNWadmc    add
package             SUNWtcpd    add
package             SUNWaccr    add
package             SUNWaccu    add
package             SUNWbash    add
package             SUNWgcmn    add
package             SUNWgtar    add
package             SUNWgzip    add
package             SUNWzip     add
package             SUNWsshdr   add
package             SUNWsshdu   add
package             SUNWsshr    add
package             SUNWsshu    add
package             SUNWatfsr   delete
package             SUNWatfsu   delete
package             SUNWftpr    delete
package             SUNWkrbr    delete
package             SUNWkrbu    delete
package             SUNWlldap   delete
package             SUNWCsndm   delete
package             SUNWnisr    delete
package             SUNWnisu    delete
package             SUNWdtcor   delete
package             SUNWinamd   delete
package             SUNWrcmdr   delete
package             SUNWrcmdu   delete
package             SUNWtnetr   delete
package             SUNWtnetu   delete
package             SUNWtnamr   delete
```

```
package       SUNWtnamd  delete
package       SUNWtftp   delete
package       SUNWtftpr  delete
```

Appendix B (cont.) – large.profile

```
install_type    initial_install
geo             N_America
locale          C
cluster         SUNWCcore
partitioning    explicit
filesys         rootdisk.s0     free    /       logging
filesys         rootdisk.s1     8200    swap
filesys         rootdisk.s4     8200    /var    logging
filesys         rootdisk.s5     12      unnamed
package             SUNWzlib    add
package             SUNWzlibx   add
package             SUNWntpr    add
package             SUNWntpu    add
package             SUNWmdr     add
package             SUNWmdu     add
package             SUNWlibC    add
package             SUNWadmfw   add
package             SUNWadmc    add
package             SUNWtcpd    add
package             SUNWaccr    add
package             SUNWaccu    add
package             SUNWbash    add
package             SUNWgcmn    add
package             SUNWgtar    add
package             SUNWgzip    add
package             SUNWzip     add
package             SUNWsshdr   add
package             SUNWsshdu   add
package             SUNWsshr    add
package             SUNWsshu    add
package             SUNWatfsr   delete
package             SUNWatfsu   delete
package             SUNWftpr    delete
package             SUNWkrbr    delete
package             SUNWkrbu    delete
package             SUNWlldap   delete
package             SUNWCsndm   delete
package             SUNWnisr    delete
package             SUNWnisu    delete
package             SUNWdtcor   delete
package             SUNWinamd   delete
package             SUNWrcmdr   delete
package             SUNWrcmdu   delete
package             SUNWtnetr   delete
package             SUNWtnetu   delete
package             SUNWtnamr   delete
```

```
package       SUNWtnamd  delete
package       SUNWtftp   delete
package       SUNWtftpr  delete
```

Appendix C – default-end.sh

```
#!/sbin/sh

# Declare Variables
#
BASE=/a
MNT=/a/mnt
ADMIN_FILE=/a/tmp/admin
mount -f nfs 192.168.0.10:/export/jumpstart/sparc9 ${MNT}


# Copy, change, or create all those files for which settings need to be customized
#
echo "*** Copying / Modifying configuration files ***"

# Backup and remove all entries from /etc/inetd.conf
cp ${BASE}/etc/inetd.conf                    ${BASE}/etc/inetd.conf.orig
cp /dev/null                                 ${BASE}/etc/inetd.conf

# Install TCP Wrappers
echo “ENABLE_CONNECTION_LOGGING=YES” >> ${BASE}/etc/default/inetd
echo “ENABLE_TCPWRAPPERS=YES” >> ${BASE}/etc/default/inetd
echo “in.tftpd: 192.168.0.0” > ${BASE}/etc/hosts.allow
echo “in.sshd: 192.168.0.0” >> ${BASE}/etc/hosts.allow
echo “ALL: ALL EXCEPT localhost : DENY” > ${BASE}/etc/hosts.deny

# Log failed authentications and SU attempts
echo “SYSLOG_FAILED_LOGINS=0” >> ${BASE}/etc/default/login
echo “auth.notice       /var/log/authlog” >> /etc/syslog.conf
touch ${BASE} /var/log/authlog
chmod 600 ${BASE} /var/log/authlog
touch ${BASE}/var/log/sulog
chmod 600 ${BASE}/var/log/sulog

# Set default banner message
echo “Authorized Uses Only.” > ${BASE}/etc/issue

# Set default banner for SSH
echo “Banner  /etc/issue” >> ${BASE}/etc/ssh/sshd_config

# Set up NTP Client
cat > ${BASE}/etc/inet/ntp.conf <<EOF
restrict default nomodify
restrict 192.168.0.5
restrict 192.168.0.6
restrict 127.0.0.1
server 192.168.0.5
```

```
server 192.168.0.6
driftfile /var/ntp/ntp.drift
EOF
touch ${BASE}/var/ntp/ntp.drift

# Disable Unnecessary service in /etc/rc2.d
rm ${BASE}/etc/rc2.d/S71ldap.client  ${BASE}/etc/rc2.d/S71rpc \
${BASE}/etc/rc2.d/S73cachefs.daemon ${BASE}/etc/rc2.d/S76nscd \
${BASE}/etc/rc2.d/S89PRESERVE ${BASE}/etc/rc2.d/S93cacheos.finish

#Install the mirror script from the “files” directory (see Appendix F for the mirror script)
cp ${SI_CONFIG_DIR}/sparc9/files/mirror        ${BASE}/etc/rc2.d/S99mirror
chmod +x ${BASE}/etc/rc2.d/S99mirror

## EXAMPLE - Add Third party packages / software from the "software" directory
## Enable by removing first column of comments
##
#echo "*** Adding third party packages and software ***"
#
#mkdir ${BASE}/usr/local
#cat >${ADMIN_FILE} <<EOF
#mail=root
#instance=overwrite
#partial=nocheck
#runlevel=nocheck
#idepend=nocheck
#rdepend=nocheck
#space=ask
#setuid=nocheck
#conflict=nocheck
#action=nocheck
#basedir=default
#EOF
#
#echo "  ...installing packages."
#
#/usr/sbin/pkgadd -a ${ADMIN_FILE} -n -d ${MNT}/software/coreutils-4.5.4-sol9-sparc-local \
#-R ${BASE} <<EOF
#all
#EOF

# Install recommended patch cluster
# Get the patches from the “patch” directory
echo "*** Installing recommended patch cluster ***"

cp /etc/mnttab ${BASE}/etc/mnntab
cd ${MNT}/patches/9_Recommended; /usr/sbin/patchadd -R ${BASE} -M . ./patch_order -u; \
cd ${BASE}
```

```
sleep 30

umount ${MNT}
#
#
# End script
```

Appendix C (cont.) – nomirror-end.sh

```
#!/sbin/sh

# Declare Variables
#
BASE=/a
MNT=/a/mnt
ADMIN_FILE=/a/tmp/admin
mount -f nfs 192.168.0.10:/export/jumpstart/sparc9 ${MNT}


# Copy, change, or create all those files for which settings need to be customized
#
echo "*** Copying / Modifying configuration files ***"

# Backup and remove all entries from /etc/inetd.conf
cp ${BASE}/etc/inetd.conf                    ${BASE}/etc/inetd.conf.orig
cp /dev/null                                 ${BASE}/etc/inetd.conf

# Install TCP Wrappers
echo “ENABLE_CONNECTION_LOGGING=YES” >> ${BASE}/etc/default/inetd
echo “ENABLE_TCPWRAPPERS=YES” >> ${BASE}/etc/default/inetd
echo “in.tftpd: 192.168.0.0” > ${BASE}/etc/hosts.allow
echo “in.sshd: 192.168.0.0” >> ${BASE}/etc/hosts.allow
echo “ALL: ALL EXCEPT localhost : DENY” > ${BASE}/etc/hosts.deny

# Log failed authentications and SU attempts
echo “SYSLOG_FAILED_LOGINS=0” >> ${BASE}/etc/default/login
echo “auth.notice       /var/log/authlog” >> /etc/syslog.conf
touch ${BASE} /var/log/authlog
chmod 600 ${BASE} /var/log/authlog
touch ${BASE}/var/log/sulog
chmod 600 ${BASE}/var/log/sulog

# Set default banner message
echo “Authorized Uses Only.” > ${BASE}/etc/issue

# Set default banner for SSH
echo “Banner  /etc/issue” >> ${BASE}/etc/ssh/sshd_config

# Set up NTP Client
cat > ${BASE}/etc/inet/ntp.conf <<EOF
restrict default nomodify
restrict 192.168.0.5
restrict 192.168.0.6
restrict 127.0.0.1
server 192.168.0.5
```

```
server 192.168.0.6
driftfile /var/ntp/ntp.drift
EOF
touch ${BASE}/var/ntp/ntp.drift

# Disable Unnecessary service in /etc/rc2.d
rm ${BASE}/etc/rc2.d/S71ldap.client  ${BASE}/etc/rc2.d/S71rpc \
${BASE}/etc/rc2.d/S73cachefs.daemon ${BASE}/etc/rc2.d/S76nscd \
${BASE}/etc/rc2.d/S89PRESERVE ${BASE}/etc/rc2.d/S93cacheos.finish

# DO NOT need to copy a mirroring script. This is the nomirror script.
# See the “default-end-sh” (Appendix C) script and the
# “mirror” script (Appendix F) for more information.

## EXAMPLE - Add Third party packages / software from the "software" directory
## Enable by removing first column of comments
##
#echo "*** Adding third party packages and software ***"
#
#mkdir ${BASE}/usr/local
#cat >${ADMIN_FILE} <<EOF
#mail=root
#instance=overwrite
#partial=nocheck
#runlevel=nocheck
#idepend=nocheck
#rdepend=nocheck
#space=ask
#setuid=nocheck
#conflict=nocheck
#action=nocheck
#basedir=default
#EOF
#
#echo "  ...installing packages."
#
#/usr/sbin/pkgadd -a ${ADMIN_FILE} -n -d ${MNT}/software/coreutils-4.5.4-sol9-sparc-local \
#-R ${BASE} <<EOF
#all
#EOF

# Install recommended patch cluster
# Get the patches from the “patch” directory
echo "*** Installing recommended patch cluster ***"

cp /etc/mnttab ${BASE}/etc/mnntab
cd ${MNT}/patches/9_Recommended; /usr/sbin/patchadd -R ${BASE} -M . ./patch_order -u; \
cd ${BASE}
```

```
sleep 30

umount ${MNT}
#
#
# End script
```

Appendix D (cont.) – sysidcfg

```
system_locale=C
name_service=NONE
network_interface=primary {hostname=server1
                ip_address=192.168.0.20
                netmask=255.255.255.0
                protocol_ipv6=no
                default_route=192.168.0.1}
security_policy=NONE
root_password=H5o.sHX3wB6
timezone=US/Central
timeserver=localhost
terminal=vt100
```

Appendix E – setup-client

```
#!/sbin/sh

Usage() {
        echo "
        echo 'This is a small utility to setup new Solaris 9 Jumpstart clients.  This works'
        echo 'for 99% of clients.  Known limitations include: mirroring only works on'
        echo 'clients with scsi/fiber disks on c0 or c1, and x86 clients are not supported.'
        echo "
        echo 'Usage:  setup-client [-n] [-s | -l] <8 | 9> <hostname> <IP_Addr> <MAC_Addr>'
        echo '   -n       = nomirror:  Turns off automatic root disk mirroring (optional)'
        echo '   -s | -l  = small | large:  Disk size/layout                  (optional)'
        echo '   8 | 9    = Solaris Version'
        echo '   hostname = Desired hostname of the new client'
        echo '   IP_Addr  = Desired IP address of the new client (must be 192.168.0.x)'
        echo '   MAC_Addr = MAC address for the primary interface of the new client'
        echo "
        echo 'Example: setup-client       9 foo 10.10.10.10 0:3:ba:2e:99:1d'
        echo 'Example: setup-client -n -s 8 foo 10.10.10.10 0:3:ba:2e:99:1d'
        echo "
        exit 1
}

NOMIRROR="default"
DISK="default"

while getopts 'lns' c
do
        case $c in
                l) DISK="large" ;;
                n) NOMIRROR="nomirror" ;;
                s) DISK="small" ;;
                *) Usage ;;
        esac
done
shift `expr $OPTIND - 1`

if [ ${NOMIRROR} != "default" ]
then
        echo "Root disk mirroring has been set to ${NOMIRROR}"
fi

if [ ${DISK} != "default" ]
then
        echo "Disk size/layout has been set to ${DISK}"
fi
```

```
if [ $# != 4 ]
then
        Usage
        exit 1
fi

# echo "exiting test... DISK=${DISK}, NOMIRROR=${NOMIRROR}"
# exit 0

VERSION=9
HOST_NAME=$2
IP_ADDR=$3
MAC_ADDR=$4
JMP_SERVER=jumpstart
INST_CLIENT_BASE=
PROFILE=
END_SCRIPT=


fgrep -s "$HOST_NAME" /etc/hosts
if [ $? != 1 ]
then
        echo "!!! $HOST_NAME is already present in /etc/hosts. Please correct and re-run
setup-client !!!"
        exit 1
fi

fgrep -s "$HOST_NAME" /etc/ethers
if [ $? != 1 ]
then
        echo "!!! $HOST_NAME is already present in /etc/ethers. Please correct and re-run
setup-client !!!"
        exit 1
fi

fgrep -s "$HOST_NAME" /export/jumpstart/rules
if [ $? != 1 ]
then
        echo "!!! $HOST_NAME is already present in /export/jumpstart/rules. Please correct
and re-run setup-client !!!"
        exit 1
fi

if [ -d /export/sysidcfg/$HOST_NAME ]
then
        echo "!!! $HOST_NAME already has a sysidcfg file. Please correct and re-run
setup-client !!!"
        exit 1
```

```
else
        mkdir /export/sysidcfg/$HOST_NAME
fi

echo "$IP_ADDR      $HOST_NAME        #Added by setup-client for Jumpstart" >> /etc/hosts
echo "$MAC_ADDR $HOST_NAME        #Added by setup-client for Jumpstart" >> /etc/ethers
echo "system_locale=C
name_service=NONE
network_interface=primary {hostname=$HOST_NAME
                    ip_address=$IP_ADDR
                    netmask=255.255.255.0
                    protocol_ipv6=no
                    default_route=192.168.0.1}
security_policy=NONE
root_password=H5o.sHX3wB6
timezone=US/Central
timeserver=localhost
terminal=vt100" > /export/sysidcfg/$HOST_NAME/sysidcfg

echo "hostname $HOST_NAME   -   ./sparc${VERSION}/${DISK}.profile   ./sparc${VERSION}
${NOMIRROR}-end.sh" >> /export/jumpstart/rules
cd /export/jumpstart; ./check

/export/install/sparc${VERSION}/Solaris_${VERSION}/Tools/add_install_client -c
$JMP_SERVER:/export/jumpstart -p $JMP_SERVER:/export/sysidcfg/$HOST_NAME -s
$JMP_SERVER:/export/install/sparc${VERSION} $HOST_NAME sun4u

echo "Added $HOST_NAME to Jumpstart on `date`" >> /export/jumpstart/client.log
#end script
```

Appendix F – mirror

```
#!/sbin/sh

#This mirror script will work for 99% of the configurations, but you may
#have trouble with weird things like ATA drives.
#
#Set up variable for disks
DISK1=notfound
DISK2=notfound

#First, see what type of disk we're mirroring
fgrep -s "c0t0d0s0" /etc/vfstab
if [ $? != 1 ]
then
        echo " Found root disk. Trying to mirror disks on controller-0."
        #Now we need to be sure there is a second drive to mirror
        DISK1=c0t0d0s

        fgrep -s "c0t1d0s0" /var/sadm/system/data/vfstab.unselected
        if [ $? != 1 ]
        then
                DISK2=c0t1d0s
        else
                echo "There is no second drive on controller-0 to mirror."
        fi
else
        echo "Cannot find any disks on controller-0... Trying controller-1..."
fi

fgrep -s "c1t0d0s0" /etc/vfstab
if [ $? != 1 ]
then
      echo " Found root disk. Trying to mirror disks on controller-1."
      #Now we need to be sure there is a second drive to mirror
      DISK1=c1t0d0
         fgrep -s "c1td0s0" /var/sadm/system/data/vfstab.unselected
         if [ $? != 1 ]
         then
                DISK2=c1t1d0
         else
            echo "There is no second drive on controller-1 to mirror."
         fi
else
        echo "Cannot find any disks on controller-0 or controller-1. Please mirror disks manually."
fi

if [ "$DISK1" != "notfound" ] && [ "$DISK2" != "notfound" ]
```

```
then
        /usr/sbin/prtvtoc /dev/rdsk/${DISK1}s2 | /usr/sbin/fmthard -s - /dev/rdsk/${DISK2}s2

        /usr/sbin/metadb -a -f -c 2 c${DISK1}s5 ${DISK2}s5

        /usr/sbin/metainit -f d1  1 1 ${DISK1}s0
        /usr/sbin/metainit -f d11 1 1 ${DISK1}s1
        /usr/sbin/metainit -f d21 1 1 ${DISK1}s4

        /usr/sbin/metainit -f d2  1 1 ${DISK2}s0
        /usr/sbin/metainit -f d12 1 1 ${DISK2}s1
        /usr/sbin/metainit -f d22 1 1 ${DISK2}s4

        /usr/sbin/metainit d0  -m d1
        /usr/sbin/metainit d10 -m d11
        /usr/sbin/metainit d20 -m d21

        /usr/sbin/metaroot d0

        cp /etc/vfstab /etc/vfstab.orig

        cat > /etc/vfstab <<EOF
#device                 device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
fd      -       /dev/fd         fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/md/dsk/d10         -       -       swap    -       no      -
/dev/md/dsk/d0          /dev/md/rdsk/d0         /       ufs     1       no      logging
/dev/md/dsk/d20         /dev/md/rdsk/d20        /var    ufs     1       no      logging
swap    -       /tmp    tmpfs   -       yes     -
EOF

        cat > /etc/rc3.d/S99mirror2 <<EOF
#!/sbin/sh

/usr/sbin/metattach d0 d2
/usr/sbin/metattach d10 d12
/usr/sbin/metattach d20 d22

/usr/sbin/dumpadm -d /dev/md/dsk/d10

/bin/rm /etc/rc3.d/S99mirror2

# End script
EOF

        chmod +x /etc/rc3.d/S99mirror2
```

```
fi

/bin/rm /etc/rc2.d/S99mirror

sleep 30

/usr/sbin/reboot

# End script
```