Global Information Assurance Certification Paper

Label Controlled File Transfer Server – Case Study

SANS GSEC Practical v.1.4b, Option 2

Don C. Weber

January 13, 2003

Abstract:

I had recently been assigned to a project whose main objective was to create a secure file transfer system on a secure operating system which will be used to provide users the capability of uploading and downloading label controlled data. My specific task was the file transfer potion of the system. I was told the secure operating system that the project would be utilizing, the file transfer program I would be installing, and a suggested direction of implementation. As I had never worked with any of the three, I knew that I had my work cut out for me.

The following is the process that I used to configure my portion of the label controlled file transfer system. I will touch on Trusted Solaris (TSOL), the secure operating system, Washington University File Transfer Protocol Daemon (wu-ftpd), the file transfer program, and a chroot jail, the suggested direction of implementation. By the end of the project I had configured a file transfer system that displayed an acceptable amount of security for my project leader (What more can you ask?).

Before:

Glossary of terms:

chroot – Unix command used to run command or interactive shell with special root directory.
Chroot Jail – Area of a file system used to isolate programs from the rest of the operating system.
Discretionary Access Control – Also referred to as DAC. An access control mechanism that enables the owner of a file or directory to grant or deny access to other users.
Ftpaccess – ftpd configuration file
ldd – Unix command used to print shared library dependencies.
Least Sensitive – Also referred to as LS. MAC label for low security files.
Mandatory Access Control - Also referred to as MAC. A system-enforced access control mechanism that uses clearances and sensitivity labels to enforce security policy.
Most Sensitive – Also referred to as MS. MAC label for high security files.
Pam Modules – Pluggable Authentication Modules for Unix
Solaris Management Console – Also referred to as SMC. An application launcher used to access GUI-based administration tools in the Trusted Solaris operating environment.
runpd – Trusted Solaris command used to determine the privileges required to run programs.
Trusted Solaris 8 – Also referred to as TSOL. Sun secure operating system version 8.
wu-ftpd – Popular file transfer protocol program.

From the 'get go', I was told I would be working on a project involving Sun's Trusted Solaris operating system. My duty would be to install the wu-ftpd servers capable of downloading and uploading files from two different networks, a Most Sensitive (MS) network and a Least Sensitive (LS) network. I was also told that the wu-ftpd server had been set up before on a server running Sun's Solaris 9 and that I would have some project documentation that I could reference to see how it was a protected within a chroot jail. (I am unable to provide that documentation or author with this practical but I will refer to it as the jail documentation from this point on.)

The project leader expressed to me that his main concern was that a malicious user would be able to gain root access, at some point, due to the fact that the service is require to run as root to enable it to bind to a port lower than 1024 (wu-ftpd can be set up to utilize ephemeral ports but configuration limitations did not let me use this option). Although wu-ftpd and chroot are not without their faults, these would be a part of the multiple layers of defense used to prevent that malicious user from doing what he/she wanted whenever he/she wanted. The risk is that the data stored on the ftp server might be compromised or that the system itself might be compromised and used to compromise other systems on the network to compromise data in the future.

So, armed with the names of the tools I would be utilizing, I set out to find some documentation on them. I started with the obvious, Sun's Trusted Solaris operating system. The first thing that I found was the on-line manual for TSOL 8 (see references for url). Here is a quote from the user guide, it is rather large but I feel that it will describe TSOL a little better than I can.

"The Trusted Solaris software package is an enhanced version of the Solaris operating environment (including the Common Desktop Environment (CDE)), with special security features. The Trusted Solaris environment enables an organization to define and implement a security policy for a single Sun workstation or a network of Sun workstations. A security policy is the set of rules and practices that help protect information and other resources (such as computer hardware) in your system. Typically, rules deal with such items as who has access to which information or who is allowed to write files to tape; practices are recommended procedures for performing tasks". (docs.sun.com)

I also found the TSOL forum to which I could search on and/or post any questions about TSOL (see references for url). I did further searches on TSOL 8 but these were the best resources I found and it does not seem to be widely documented openly. The overall project took a few days to set up the TSOL boxes and get them running in the manner we needed to implement. This worked to my advantage because it gave me a look within the operating system and showed me a few of the pitfalls that we going to challenge me during the installation of the jailed wu-ftpd. TSOL controls everything through permissions. There are normal file permissions (DAC), the added file label permissions (MAC), and the "kernel permissions" that controls the actions a executable can perform

(Please Note: I came up with the term “kernel permission” on my own to give them a degree of separation from MAC and DAC privileges for this explanation, the Sun documentation refers to all of them as permissions. I will do the same for the rest of this text.).

TSOL control the operating system by splitting the administrative responsibilities. The biggest, and at times most frustrating change, is that they have removed the “Superuser” capability. No longer can the administrator run around as “root” and change anything and everything. This might be good for security, and it is the desired effect of a secure operating system, but it makes system administration, at times, a nightmare. One piece of Sun documentation that became invaluable was “To Find Out Which Privileges a Program Needs.” (see reference for url). The command “runpd” has been added to TSOL to assist the administrator in finding which permissions a program needs to run. Now, there are two directions that this command can take an administrator. Not only does it list the permissions the program needs to run successfully, it runs the program, giving it the necessary permissions automatically. (I can see the light bulbs flashing.) A lazy administrator might take the very tempting route of including “runpd” in every script. Beware of this procedure. Child processes will inherit the privileges given the parent. If “runpd” was utilized, a malicious user might be able to gain these permissions and run amuck. The other path, however, will save the administrator countless hours of searching, applying, and stripping permissions. This involves only utilizing “runpd” to perform common administrative routines. This will become more apparent as I describe the wu-ftpd setup. With the operating system up and running I was ready to move on.

Next I did research on the ftp server. First of all, TSOL does come, standard, with an ftp daemon. My project lead was fairly sure that it was a Sun modified version of wu-ftpd but he had several security concerns. First of all we had no clue what version of wu-ftpd they had implemented. The current version is 2.6.2 (wu-ftpd.org) but if Sun implemented a later version then there would be vulnerabilities. Also, Sun’s version runs through the inet services (which we were disabling for security) and this brought up questions of how the permissions were implemented and if a malicious user were to break out, he was not 100 percent sure of what that user would be capable of accomplishing since the ftp daemon has to run as root to connect to the proper ports. This left us with replacing the ftp daemon with a brand new copy of wu-ftpd. But I was not sure exactly what wu-ftpd was until I went to wu-ftpd.org and found this statement.

“Wuarchive-ftpd, more affectionately known as WU-FTPD, is a replacement ftp daemon for Unix systems developed at Washington University (.wustl.edu) by Chris Myers and later by Bryan D. O'Connor (who are no longer working on it or supporting it!). WU-FTPD is the most popular ftp daemon on the Internet, used on many anonymous ftp sites all around the world.” (wu-ftpd.org)*

Wu-ftpd is completely documented at the server’s home page www.wu-ftpd.org. Downloads and documentation, jackpot! With a little more looking I came across the Landfield Group. “*The Landfield Group maintains an archive of publicly*

available material," (landfield.com) to include, but not limited, to wu-ftpd. This proved to be a great find because it is here that I found one of the most valuable documents for this project, "How to setup WU_FTPD Guest Accounts." As I looked at this tutorial and compared it to the jail documentation I figured I had enough information on the wu-ftpd server to move onto finding out what chroot was and how I could use it to secure wu-ftpd.

Chroot would be easy, or so I thought. Chroot is a unix command with man pages and plenty of other documentation for configuration and utilization. Overall, chroot is described very well by Steve Friedl in his article for the December 2002 issue of Linux Magazine.

"Think of chroot() as a kind or reality distorter. Once a running process executes chroot("/home/jail"), /home/jail becomes "/," and for all intents and purposes, every file and directory outside of /home/jail (including the true root directory and true /home directory) no longer exist."

This might have confused me at first but now that I have a better understanding I realize that the chroot jail basically hides the rest of the file system from not only the user but the process as well. This means that anything a program is going to need to run must be within the chroot jail. Static libraries, scripts, binaries, file structure, all must be implemented in the exact same manner that the program is expecting or it will "blow up." In the long run, I discovered that the only real hard part about implementing chroot is making sure that all the boxes are checked. I would pick this up the hard way later, but for now I had the chroot man pages, I had the jail documentation provided to me by my project leader. Unfortunately, I did not have Mr. Friedl's great article on chroot until after I had fought and banged my way through all the configuration of a chrooted, or jailed, wu-ftpd server. Satisfied that the majority of my research was completed I was able to move onto setting up the label controlled file transfer daemon.

During:

The following is the final result of the setup for the jailed wu-ftpd daemon on a TSOL server. It is a step-by-step tutorial and I have injected comments throughout this tutorial for a better description of what is taking place in certain steps. Comments are shown in the following fashion, "(## COMMENT ##)."

1) Download current wu-ftpd.
(## Currently using version wu-ftpd 2.6.2(1). ##)

2) Install wu-ftpd:

a) The wu-ftpd package should be configured and made on a Solaris 8 [SOL8] box, with the exception of the final step ('make install'). The

package should then be transfered over to the Trusted Solaris [TSOL] box.

(## If the system has the current version of gcc installed then wu-ftpd can be compiled and made on this system. However, TSOL does not come with a default version of gcc and it might go against project parameters to have this installed for obvious security reasons. ##)

b) There may be some commands that are needed to complete the install. Locate these commands and add them to the PATH. I.E. For 'ranlib' add /usr/ccs/bin to the PATH. Don't forget to export the PATH variable.

c) 'runpd make install' will then install wu-ftpd on the TSOL box.

3) Run the following commands to create the directory structure needed for the jail.

a) 'mkdir -p /jail/serv'
'cd /jail/serv'
'mkdir -p dev lost+found sbin scripts usr/lib/security etc/lib/sparcv9'
'mkdir -p usr/lib/secpolicy/chroot usr/lib/secpolicy/passwd var/ld'
'mkdir -p usr/platform/SUNW,Ultra-250/lib usr/share/lib/zoneinfo/US'
'mkdir -p home/bin home/dev home/usr/lib'
'mkdir -p home/leastsensitive/bin home/leastsensitive/dev'
'mkdir -p home/mostsensitive home/leastsensitive/usr/lib'
'mkdir -p home/usr/platform/SUNW,Ultra-250/lib'
'mkdir -p home/leastsensitive/usr/platform/SUNW,Ultra-250/lib'
'mkdir -M var/run' (## This is a multilevel directory. ##)
'mkdir -M etc/ftpconfig' (## This is a multilevel directory. ##)

(## Mulitlevel directories are directories that can contain files with different MAC permissions and are controlled by the operating system. A user at one level will not see files with any other label because these files are in a completely different directory that is hidden by TSOL. I.E. Users at the Most Sensitive (MS) level will not be able to even see any files labeled Least Sensitive (LS). ##)

(## Final directory structure will look as follows. All are ADMIN_LOW unless otherwise stated. ##)

(## ADMIN_LOW is a TSOL label that system administrators operate at to perform their duties. This does give them general access to be able to perform their functions but even when operating at this level administrators are still limited by all the other permissions. There is an ADMIN_HIGH label that administrators, given this privilege, can operate with which enables them to perform some higher level functions, such as viewing system log files. ##)

/jail/serv

```
/dev
/lost+found
/sbin
/scripts
/etc
        /ftpconfig (## multilevel dir ##)
        /lib
                /sparcv9
/home (## LS labeled ##)
        /bin
        /dev
        /leastsensitive (## LS labeled ##)
                /bin
                /dev
                /usr
                        /lib
                        /platform
                                / SUNW,Ultra-250
                                        /lib

        /mostsensitive (## MS labeled)
        /usr
                /lib
                /platform
                        / SUNW,Ultra-250
                                /lib
/var
        /ld
        /run (## multilevel dir ##)
/usr
        /lib
                /secpolicy
                        /chroot
                        /passwd
        /platform
                /SUNW,Ultra-250
                        /lib
        /share
                /lib
                        /zoneinfo
                                /US
```

4) Make the device files.

a) First you have to get the major and minor of the devices.
‘ls -Ll /dev/zero’

'ls -Ll /dev/conslog'
'ls -Ll /dev/null'
'ls -Ll /dev/tty'

The output should look like:
'crw-rw-rw- 1 root sys 21, 0 Aug XX xx:xx /dev/conslog'
The major is 21 and the minor is 0.

b) Make the devices from /jail/serv.
'runpd mknod dev/conslog c major minor'
'runpd mknod dev/null c major minor'
'runpd mknod dev/tty c major minor'
'runpd mknod dev/zero c major minor'
'runpd mknod home/dev/null c major minor'
'runpd mknod home/dev/zero c major minor'
'runpd mknod home/leastsensitive/dev/null c major minor'
'runpd mknod home/leastsensitive/dev/zero c major minor'

5) Create separate server and chroot binaries and give them the proper permissions.

a) Copy the binary located at '/usr/sbin/in.ftpd' to '/jail/serv/sbin/ls.ftpd' [ls.ftpd] (LS server) and '/jail/serv/sbin/ms.ftpd' [ts.ftpd] (MS server). You may delete the original if you wish.

b) Change the allowed privileges on both new binaries to allow ONLY 'net_privaddr', 'proc_chroot', 'proc_setid'.

c) In a terminal with the same label, set the label of ls.ftpd to "[LEAST SENSITIVE] and ms.ftpd to "[MOST SENSITIVE]".

d) Copy the binary located at '/usr/sbin/chroot' to '/usr/sbin/chroot.ls' and '/usr/sbin/chroot.ms'. Use the 'cp -p' option to preserve the file permissions. DO NOT delete the original.

e) In a terminal with the same label, set the label of chroot.ls to "[LEAST SENSITIVE]" and chroot.ms to "[MOST SENSITIVE]".

(## The binaries need to be renamed for two reasons. First, it makes the individual process more readily recognizable to the administrator. Second, and more important, as I stated before, wu-ftpd replaces the default install of in.ftpd that comes with TSOL. In order for the original in.ftpd to function properly under TSOL it is given a lot of privileges we deemed unnecessary and a security risk. The new binaries are disassociated from these privileges and only run with the ones we provide above. ##)

6) Modify /etc/ftpaccess file to run dual servers at different levels.

(## The ftpaccess file is one of the most important ways to control every aspect of the wu-ftpd daemon. Through documentation is provided via the man pages or at wu-ftpd.org. ##)

(## Ensuring that the terminal label is correct is a VERY IMPORTANT part of this step! ##)

a) Make an ftpaccess file to model the ftpaccess.txt included with this setup procedure (Appendix A). Be sure to change the information to match the current network configuration, user directories (from step 5), and ftpgroup information.

b) Copy the ftpaccess to etc/ftpaccess.ls and edit the file for the IP address and file directory of the LS server. Then change the label to [LEAST SENSITIVE].

c) Follow the step above (b) to make etc/ftpaccess.ms labeled [MOST SENSITIVE].

d) In an LS terminal 'cp etc/ftpaccess.ls to etc/ftpconfig/ftpaccess'

e) In an MS terminal 'cp etc/ftpaccess.ms to etc/ftpconfig/ftpaccess'

g) Make a simbolic link from ftpaccess to etc/ftpconfig/ftpaccess with the command 'ln -s ftpconfig/ftpaccess ftpaccess'

(## This lets us utilize two separate ftpaccess files without modifing the wu-ftpd source code or creating a script to change ftpaccess for the different level servers. Depending on what level process tries to access the file will determine which file it utilizes. ##)

7) Copy the following files to these directories.

(## This list of files was obtained utilizing the "ldd" command to determine which libraries the wu-ftpd server required. The files required by the pam modules for password verification, however, were not listed by this command. Fortunately, the jail documentation listed these files for me. The command for listing files is included because the "ls" command within the wu-ftpd server is broken and cannot handle long listings. By including these files here files can be long listed. Performing the "ldd" command on "ls" also gave the libraries required by this command. Please note, because the wu-ftpd server is performing its own chroot, a complete directory structure must be constructed at the new root level. The null and zero devices must be included, as well as the required libraries, for the "ls" command to function correctly. This might seem tedious but life without long listing is very unrewarding. ##)

/etc/nsswitch.conf => etc/nsswitch.conf
/etc/lib/ld.so.1 => etc/lib/ld.so.1
/etc/lib/libdl.so.1 => etc/lib/libdl.so.1
/etc/lib/libsecpolicy.so.1 => etc/lib/libsecpolicy.so.1
/etc/lib/nss_files.so.1 => etc/lib/nss_files.so.1

```
        /etc/lib/sparcv9/libsecpolicy.so.1  => etc/lib/sparcv9/libsecpolicy.so.1
        /usr/lib/ld.so.1                => usr/lib/ld.so.1
        /usr/lib/ld.so.1                => home/usr/lib/ld.so.1
        /usr/lib/ld.so.1                => home/leastsensitive/usr/lib/ld.so.1
        /usr/lib/libbsm.so.1  => usr/lib/libbsm.so.1
        /usr/lib/libc.so.1      => usr/lib/libc.so.1
        /usr/lib/libc.so.1      => home/usr/lib/libc.so.1
        /usr/lib/libc.so.1      => home/leastsensitive/usr/lib/libc.so.1
        /usr/lib/libcmd.so.1  => usr/lib/libcmd.so.1
        /usr/lib/libcrypt_i.so.1=> usr/lib/libcrypt_i.so.1
        /usr/lib/libdl.so.1     => usr/lib/libdl.so.1
        /usr/lib/libdl.so.1     => home/usr/lib/libdl.so.1
        /usr/lib/libdl.so.1     => home/leastsensitive/usr/lib/libdl.so.1
        /usr/lib/libdoor.so.1  => usr/lib/libdoor.so.1
        /usr/lib/libgen.so.1   => usr/lib/libgen.so.1
        /usr/lib/libldap.so.4  => usr/lib/libldap.so.4
        /usr/lib/libmd5.so.1  => usr/lib/libmd5.so.1
        /usr/lib/libmp.so.2    => usr/lib/libmp.so.2
        /usr/lib/libnsl.so.1    => usr/lib/libnsl.so.1
        /usr/lib/libpam.so.1  => usr/lib/libpam.so.1
        /usr/lib/libproject.so.1=> usr/lib/libproject.so.1
        /usr/lib/libresolv.so.2         => usr/lib/libresolv.so.2
        /usr/lib/libsecdb.so.1          => usr/lib/libsecdb.so.1
        /usr/lib/libsldap.so.1=> usr/lib/libsldap.so.1
        /usr/lib/libsocket.so.1         => usr/lib/libsocket.so.1
        /usr/lib/libt6.so.1     => usr/lib/libt6.so.1
        /usr/lib/libtsol.so.1   => usr/lib/libtsol.so.1
        /usr/lib/nss_files.so.1         => usr/lib/nss_files.so.1
        /usr/lib/secpolicy/chroot/chroot.kpolicy-tsol.so
                                => usr/lib/secpolicy/chroot/chroot.kpolicy-tsol.so
        /usr/lib/secpolicy/passwd/passwd.kpolicy-tsol.so
                                => usr/lib/secpolicy/passwd/passwd.kpolicy-tsol.so
        /usr/lib/security/pam_projects.so.1         =>
usr/lib/security/pam_projects.so.1
        /usr/lib/security/pam_roles.so.1    => usr/lib/security/pam_roles.so.1
        /usr/lib/security/pam_unix.so.1             => usr/lib/security/pam_unix.so.1
        /usr/platform/SUNW,Ultra-250/lib/libc_psr.so.1
                        => usr/platform/SUNW,Ultra-250/lib/libc_psr.so.1
        /usr/platform/SUNW,Ultra-250/lib/libc_psr.so.1
                        => home/usr/platform/SUNW,Ultra-250/lib/libc_psr.so.1
        /usr/platform/SUNW,Ultra-250/lib/libc_psr.so.1
                => home/leastsensitive/usr/platform/SUNW,Ultra-
250/lib/libc_psr.so.1
        /usr/platform/SUNW,Ultra-250/lib/libmd5_psr.so.1
                                => usr/platform/SUNW,Ultra-250/lib/libmd5_psr.so.1
        /usr/share/lib/zoneinfo/US/Central=> usr/share/lib/zoneinfo/US/Central
```

/var/ld/ld.config => var/ld/ld.config
/bin/false => bin/false
/usr/bin/ls => home/bin/ls
/usr/bin/ls => home/leastsensitive/bin/ls

8) Make these files with these commands.

a) 'cat /etc/netconfig | egrep '^(tcp) ' >etc/netconfig'
b) 'cat /etc/project | grep default >etc/project'
c) 'echo "ftp-data\t20/tcp\nftp\t\t21/tcp" >etc/services'
d) 'ksh' <enter> 'result=`grep daemon.info /etc/syslog.conf` ; if [[-z $result]] ; then print "\ndaemon.info\t/var/adm/messages" >>/etc/syslog.conf ; fi;'
e) 'echo /bin/false >etc/shells'
f) 'echo "ftp\tauth\trequisite\t/usr/lib/security/\$ISA/pam_unix.so.1" >etc/pam.conf'
g) Copy or type the banner to be display on connection to 'etc/banner'
h) Copy or type the message to be displayed, after a MS user logs in, to 'home/.ftpmessage.msg'.
i) Copy or type the message to be displayed, after a LS user logs in, to 'home/leastsensitive/.ftpwelcome.msg'.
j) 'echo "root\nftp\nanonymous\n" >etc/ftpusers

9) Change file permissions, ownership, and labels using the following commands.

(## You might have to use "runpd". ##)

(## Changing the file permission and ownerships are very important. The permissions and ownerships on the home directories give us complete control of what the user can do when connected by ftp. This includes uploading and downloading from certain directories that we can't hide but need for functionality. Although this can also be controlled through the utilization of the "ftpaccess" file not all of the features can be used, or are even wanted, in this configuration. One notable feature is the upload control capability. Although this feature completely controls what and where a user can upload, to include the creation of directories, it also changes the uploaded files ownership to one specified within the file. This means that we cannot, directly through file ownership, determine who placed a file. This can be determined through the examination of logs but it still leaves us with a serious vulnerability. Changing ownership of a file would require that we give the parent process the permissions within TSOL to perform this action. Any malicious user that was able to break out of wu-ftpd and the jail would have this capability. And that would be very bad indeed. ##)

a) 'chown 0:ftpguest /jail'
b) 'chmod 710 /jail'
c) 'chown -R 0:0 /jail/serv'

d) ‘chmod -R 700 /jail/serv’
e) ‘chown 0:ftpguest /jail/serv’
f) ‘chown 0:ftpguest /jail/serv/home’
g) ‘chown ftpguest:ftpguest /jail/serv/home/*’
h) ‘chmod 710 /jail/serv’
i) ‘chmod 710 /jail/serv/home’
k) ‘chmod -R 770 /jail/serv/home/*’
l) ‘chmod 444 /jail/serv/home/.ftpwelcome.msg /jail/serv/home/leastsensitive/.ftpwelcome.msg’
m) ‘chown root:root /jail/serv/home/.ftpwelcome.msg /jail/serv/home/leastsensitive/.ftpwelcome.msg’
(## For the following commands you must be in the properly labeled terminal. ##)
n) ‘setlabel [LEAST SENSITIVE] /jail/serv/home /jail/serv/home/leastsensitive /jail/serv/home/leastsensitive/.ftpwelcome.msg’
o) ‘setlabel “[MOST SENSITIVE]” /jail/serv/home/mostsensitive /jail/serv/home/.ftpwelcome.msg’

10) Modify ‘boot’ profile.

(## This step allows wu-ftpd to start on boot-up with only the permissions that are given to it here. ##)

a) Using the Solaris Management Console (SMC) change to ‘Trusted Solaris Configuration/Users/Rights’ then locate and open the properties for the ‘boot’.
b) Under the ‘commands’ tab move the /jail/serv/sbin/ls.ftpd and ms.ftpd servers to the ‘Commands Permitted:’ list.
c) Highlight ls.ftpd and click the ‘Security Attributes:’ edit button.
d) Under ‘Ownership’ select user->root, group->root and select ‘Real’ for both
e) Under ‘Extended Attributes’ select “[LEAST SENSITIVE]” for ‘Label’ and ‘Clearence’ and give ‘net_privaddr’, ‘proc_chroot’, ‘proc_setid’ to the ‘Privileges’.
f) Click ‘Apply’ then ‘Close’.
g) Do c, d, e, and f for ms.ftpd only set the ‘Label’ and ‘Clearence’ to “[MOST SENSITIVE]”.
h) Under the ‘commands’ tab move the /sbin/chroot.ls and chroot.ms to the ‘Commands Permitted:’ list.
i) Do c, e, and f for both chroot.ls and chroot.ms setting the ‘Label’ and ‘Clearence’ to “[LEAST SENSITIVE]” and “[MOST SENSITIVE]” respectively. You do not need to change the ‘Ownership’ fields.

11) Configure the ftp servers to start on boot-up.

a) Create a script that will start both wu-ftpd servers with the "-aSW" options and copy it to "/etc/rc2.d/S99wu_ftpd".

(## Because of the changes main to the boot-up right these process will be started at the proper security levels. ##)

12) Create script to copy the user names and passwords from the web server user file to the passwd and shadow file.

a) Copy the 'passwdport.doc' to /jail/serv/scripts/passwdport.sh .
b) Change the allowed privileges on the script to allow ONLY 'file_mac_write'
c) Using SMC locate the 'Custom Root' rights and select passwdport.sh in the commands tab.
d) Assign the command root for 'user' and 'group'.
e) Give the script the 'file_mac_write' privilege.
f) Run the script every time users need to be changed.

(## This script must be run in a LS terminal and a MS terminal to insert users in the LS password and MS password file. ##)

(## I have not included this script as a part of this paper for obvious security reasons, but it is an important part of the setup so I decided to retain this step. The first alternative was to include the "passwd" and "useradd" scripts within the jail but this posed a SERIOUS security problem due to all the permission and extra files involved, not to mention the scripts themselves. Other methods to create password and shadow files include, but are not limited to, copying the files directly and removing anything that is not wanted or needed, or to utilize the "htpasswd" script provided by the apache web server. ##)

13) Tar your work.

a) tar cvfT jailMMDDCCYY.tar /jail

(## The T option preserves the multilevel directories. This will require 'runpd.' ##)

b) Make a directory called wujail.bk and place the back up here.

14) Insure the networks and computers are added properly added and labeled on this machine.

(## TSOL needs to know every single machine that will be connecting to the server or it will deny the connection automatically. COOL!! ##)

After:

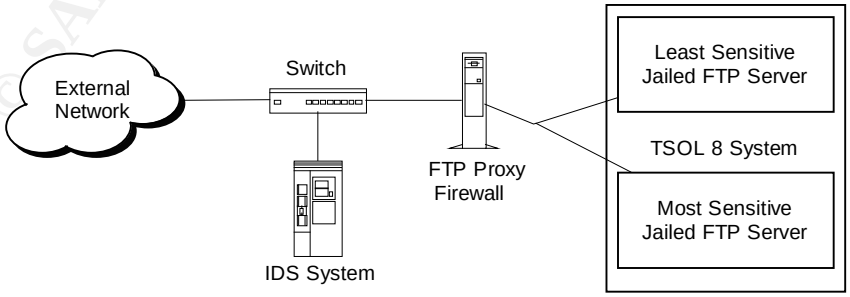
Let me recap what I was actually trying to accomplish with this project. My team leader came to me and instructed me to configure a working file transfer system utilizing Sun's Trusted Solaris operating system and a wu-ftpd server

within a chroot jail. He wanted to be able to store labeled data that could only be accessed by specific users. His main concerns were data and system integrity. I believed that I had accomplished all of these parameters to the best of my abilities. The only real way to be sure would be to test the system.

Basic testing was accomplished as I was setting up the wu-ftpd server in a "hit and miss" method. The wu-ftpd server was tested by manipulating data and user input. This was done repeatedly until the proper configurations were determined. Chroot was tested by including and then removing files that were questionable to the configuration. Using this approach with the wu-ftpd and chroot consumed the most time in this project. But, because of the importance of this portion of the project, it was critical that I check as many aspects as possible. A formal test procedure is currently in the making and once it is completed and approved the complete system will be tested in its entirety. Testing is generally handled in a much different way, but on this project, these procedures will be formed by the developers.

Of course both the wu-ftpd server and chroot itself are not risk free. Wu-ftpd has had several security issues come to light over the years. These have all been documented at wu-ftpd.org. As of the current version there are no current issues that have come to my attention. Chroot, however, is a slightly different story. There are ways to break out of a jailed process, Steve Friedl covered one in his article but Simes actually shows some code in his article, "How to break out of a chroot() jail." However, these methods require perl or c compilers within the chrooted area and are generally associated with jailed web servers. Secondly, because the ftpaccess file assigns set permissions to uploaded files through the umask feature and because the user is denied the command "chmod," uploaded files cannot be executed. Additionally, TSOL requires that select permissions be assigned to all executable programs before they are able to run, further limiting executable code.

The only real risk presented to the system now is a malicious user gaining access to the computer through means other than the jail wu-ftpd server. This issue, although not a part of my specific assignment, is addressed by the other members of the team. Switches, IDS systems, proxy firewalls, secure operating systems, and controlled environments all contribute to controlling any access to the data and systems. The following is a basic diagram of the completed system.



It seems that, as long as the system administrators properly monitor the complete system, this multiple defensive layering should be sufficient to protect the data.

Reference:

Sun Documents, “Trusted Solaris 8 Answer Book”, http://docs.sun.com/db/coll/175.4

Sun Discussion Forums, “security.trustedsolaris”, http://supportforum.sun.com/cgi-bin/WebX.cgi?folder@@/security.trustedsolaris

WU-FTPD, “WU-FTPD Development Group”, http://www.wu-ftpd.org

Kent Landfield, “The Landfield Group: Wu-ftpd Resource Center”, http://www.landfield.com/wu-ftpd

Michael Brennen, 09151995, “How to setup WU_FTPD Guest Accounts”, http://www.landfield.com/wu-ftpd/docs/guest-howto.html

Steve Friedl, December 2002, “Go Directly to Jail, Secure Untrusted Applications with Chroot”, Linux Magazine (www.linuxmagazine.com)

Sun Documents, “To Find Out Which Privileges a Program Needs”, http://docs.sun.com/db/doc/816-1048/6m7gaddjm?a=view#manageprogramprivileges-42

WU-FTPD, “Current version”, ftp://ftp.wu-ftpd.org/pub/wu-ftpd/wu-ftpd-current.tar.gz

WU-FTPD, “ftpaccess”, http://www.wu-ftpd.org/man/ftpaccess.html

imes, 05122002, “How to break out of a chroot() jail”, http://www.bpfh.net/simes/computing/chroot-break.html

Appendix A:

```
##########################################################################
##
## TSOL WU-FTP Server Configuration File
##      ** Please read the ftpaccess man page for information concerning
##         configuration
##
##      ###################################
##      ##      [x] Least Sensitive Server  ##
##      ##      [ ] Most Sensitive Server    ##
##      ###################################
##
## Date:        xxxx20xx
## Contact:   Don C. Weber
## Notes:      Please date and initial
##########################################################################

#
# Set up guest class
#
class ftpguest guest *

#
# Set up the IP address for this server to listen to for FTP connections
#
daemonaddress xxx.xxx.xxx.xxx          #Least Sensitive Server
#daemonaddress xxx.xxx.xxx.xxx         #Most Sensitive Server

#
# Limit the failed login attempts
#
loginfails 2

#
# Set the default umask for an uploaded file
#
defumask 0127

#
# Set timeout values (in seconds)          Default – off
#
#timeout accept 120
#timeout data 1200
#timeout idle 900
#timeout maxidle 1200
```

```
#
# Set the hostname displayed on successful connections
#
hostname LeastSensitive                 #Least Sensitive Server
#hostname MostSensitive                 #Most Sensitive Server

#
# Set the pre-authentication greeting so that no useful info is diplayed
#
greeting terse

#
# Set the path for the security banner file
#
banner /etc/banner

#
#  Set the login message
#
message /.ftpwelcome.msg login

#
# Force every user logging in to be a guest user
#
guestuser *

#
# Limit the command that a user can successfully execute
#
path-filter ftpguest /etc/pathmsg ^[-A-Za-z0-9_\.]*$ &\. ^-

#
# Set up the separate chroot()'ed directories for each guest class
#
guest-root /home/leastsensitive         #Least Sensitive Server
#guest-root /home                       #Most Sensitive Server

#
# Allow selective downloads (GETs)
#
#noretrieve *
noretrieve .ftpwelcome.msg

#
# Allow/disallow all other configurable commands       Default – disallow
```

```
#
compress        no      guest
tar             no      guest
chmod           no      guest
delete          no      guest
overwrite       no      guest
rename          no      guest
umask           no      guest

#
# Make logging very verbose
#
log commands guest,anonymous,real
log transfers guest,anonymous,real inbound,outbound
log security guest,anonymous,real
log syslog                          #redirects the logging of transfers to syslog

#
# Set email address of ftp server administrator
#
#email <ftpmanager>@<hostname>
```