4.1.1.1 Ensure access on /smit.log is configured Remove world read and write access to /smit.log : chmod o-rw /smit.log

4.1.1.2 Ensure access on /etc/group is configured Ensure correct ownership and permissions are in place for /etc/group : "chown root:security /etc/group

chmod u=rw,go=r /etc/group"

4.1.1.3 Ensure access on /etc/inetd.conf is configured Set the recommended permissions and ownership to /etc/inetd.conf : "chmod u=rw,go=r /etc/inetd.conf

chown root:system /etc/inetd.conf

trustchk -u /etc/inetd.conf mode=644"

4.1.1.4 Ensure access on /etc/motd is configured Apply the appropriate permissions to /etc/motd : "chown bin:bin /etc/motd

chmod u=rw,go=r /etc/motd"

4.1.1.5 Ensure access on /etc/passwd is configured Ensure correct ownership and permissions are in place for /etc/passwd : "chown root:security /etc/passwd

chmod u=rw,go=r /etc/passwd"

4.1.1.6 Ensure /etc/mail/submit.cf access is configured chmod u=rw,g=r,o= /etc/mail/submit.cf "Impact:

It will not impact the usability of application or system."

4.1.1.7 Ensure access to /etc/ssh/ssh\_banner is configured Run the following commands to set mode, owner, and group on /etc/ssh/ssh\_banner : "# chown root:root /etc/ssh/ssh\_banner

# chmod u=rw,go=r /etc/ssh/ssh\_banner"

4.1.1.8 Ensure access on /etc/ssh/ssh\_config is configured Change the permissions of the /etc/ssh/ssh\_config file to ensure that only the owner can read and write to the file: chmod 644 /etc/ssh/ssh\_config

4.1.1.9 Ensure access on /etc/ssh/sshd\_config is configured Change the permissions of the /etc/ssh/sshd\_config file to ensure all accounts can read the file but only the owner (root) can modify it: "chmod u=rw,go=r /etc/ssh/sshd\_config

4.1.1.11 Ensure access on /var/adm/cron/cron.allow is configured Apply the appropriate permissions to /var/adm/cron/cron.allow : "chown root:sys /var/adm/cron/cron.allow

chmod u=r,go= /var/adm/cron/cron.allow"

4.1.1.13 Ensure access on /var/ct/RMstart.log is configured Remove world read and write from /var/ct/RMstart.log : chmod o-rw /var/ct/RMstart.log

4.1.1.14 Ensure access on /var/tmp/dpid2.log is configured Remove world read and write from /var/tmp/dpid2.log : chmod o-rw /var/tmp/dpid2.log

4.1.1.15 Ensure access on /var/tmp/hostmibd.log is configured Remove world read and write from /var/tmp/hostmibd.log : chmod o-rw /var/tmp/hostmibd.log

4.1.1.16 Ensure access on /var/tmp/snmpd.log is configured Remove world read and write from /var/tmp/snmpd.log: chmod o-rw /var/tmp/snmpd.log

4.1.1.17 Ensure crontab is restricted to authorized users Ensure that all root crontab entries are owned and writable by root only. "The script below traverses up each individual directory path, ensuring that all directories are not group/world writable and that they are owned by the root or bin user:

crontab -l |egrep -v '^#' |awk '{print $6}' |grep ""^/"" |sort -u | while read DIR

do

DIR=${DIR:-$(pwd)}

while [[ -a ${DIR} ]]

do

[[ ""$(ls -ld ${DIR})"" = @(????????w? \*) ]] && print "" WARNING ${DIR} is world writable""

[[ ""$(ls -ld ${DIR})"" = @(?????w???? \*) ]] && print "" WARNING ${DIR} is group writable""

[[ ""$(ls -ld ${DIR} |awk '{print $3}')"" != @(root|bin) ]] && print "" WARNING ${DIR} is not owned by root or bin""

DIR=${DIR%/\*}

done

done

NOTE: Review the output and manually change the directories, if possible. Directories which are group and/or world writable or not owned by root are marked with ""WARNING""

To manually change permissions on the files or directories:

To remove group writable access:

chmod g-w <name>

To remove world writable access:

chmod o-w <name>

To remove both group and world writable access:

chmod go-w <name>

To change the owner of a file or directory:

chown <new user> <name>"

|  |  |  |
| --- | --- | --- |
| 4.1.2.2 Ensure Home directories access is configured | For all local accounts with UID >= 200: | #!/usr/bin/ksh -e # Provided to CIS by AIXTools # Copyright AIXTools, 2022 lsuser -R files -a id home account\_locked ALL | while read name ids homes locks rest; do uid=$(echo ${ids} | cut -f2 -d =) if [[ ${uid} -ge 200 ]]; then home=$(echo ${homes} | cut -f2 -d =) locked=$(echo ${locks} | cut -f2 -d =) if [[ ${home} == "/dev/null" || ${locked} == "true" ]]; then continue elif [[ ! -d ${home} ]]; then /usr/bin/printf "%-32s does not exist; Run appropriate CIS remediation " ${home} ${name} continue else /usr/bin/perl -e ' $user=$ARGV[0]; $hd=$ARGV[1]; $uid=$ARGV[2]; $huid=((stat $hd)[4]); if ($huid != $uid && $huid != 0) { printf("Locked Account: %s does not own %s. ", ${user},${hd}); exit(1); # triggers command after OR (||) }' ${name} ${home} ${uid} || /usr/bin/chuser -R files account\_locked=true $name fi fi done Impact: \* Locally administered accounts with HOME directories owned by a random userid will be locked. Valid users can open a ticket to get the UID of their HOME directory corrected. The risk of a malicious user modifying an accounts HOME directory is reduced. |
| 4.1.2.3 Ensure Home directory write access is restricted to owner | For all local accounts with UID >= 200: | - Remove write permission from home directories that have group or world write access: #!/usr/bin/ksh -e # home\_mode\_acl: 4.8.1.3 # Provided to CIS by AIXTools # Copyright AIXTools, 2022 typeset -i UIDCK=$1 typeset -i ret=0 if test $UIDCK == 0; then UIDCK=200 fi lsuser -R files -a id home account\_locked ALL | while read name ids homes locks rest; do uid\_check=$(echo ${ids} | cut -f2 -d =) if [[ ${uid\_check} -ge ${UIDCK} ]]; then home=$(echo ${homes} | cut -f2 -d =) locked=$(echo ${locks} | cut -f2 -d =) if [[ ${home} == "/dev/null" || ${locked} == "true" ]]; then continue elif [[ ! -d ${home} ]]; then /usr/bin/printf "%-32s does not exist; locking account named [%s] " ${home} ${name} chuser -R files account\_locked=true $name else [[ ${home} != "/" && ${home} != "/dev/null" ]] perl -e '$f=$ARGV[0]; $m=(stat $f)[2]; exit (($m & 022) + 1) if ($m & 0200000000); exit($m & 022);' $home # exit($m&amp;022 +1) if ($m & 0200000000) else exit ($m &022); ' $home ret=$? [[ $ret == 0 ]] && continue if (( $ret & 022 )); then printf "%s: had group or world write mode " $home chmod og-w ${home} fi if (($ret & 1)); then printf "%s: had ACL defined and enabled " $home rm -rf /tmp/$$/${home} mkdir -p /tmp/$$/${home} aclget /tmp/$$/${home} | aclput ${home} rm -rf /tmp/$$/${home} fi fi fi done - NOTE: The permission change is automatically applied to all accounts with a user ID ( uid ) greater or equal to 200 Also, if the HOME directory has already been defined to something special (here, /dev/null ) no change is made to the account attributes. - To automate the process for new users see Additional Information below. Impact: There should be no impact - at least as far a world permissions are concerned. There is a potential that all members in the group staff or system might see minimal impact - if their systems have, or had, a default umask of 002 when their accounts were created. Accounts created with a default umask of 022 or stricter will not be impacted, unless a user account modified their HOME directory mode bits to permit group and/or other write access. |