



Cybersecurity

Project 1 Hardening Summary and Checklist

OS Information

Customer	Baker Street Corporation
Hostname	lp-172-22-117-233 Baker_Street_Linux_Server
OS Version	<u>Ubuntu 24.04.1 LTS (Noble)</u>
Memory information	<u>Total 914.2, 68.5 free, 802.0 used</u>
Uptime information	<u>1:28:24 up, 54 min</u>

Checklist

Completed	Activity	Script(s) used / Tasks completed / Screenshots
<input checked="" type="checkbox"/>	OS backup	sudo tar -cvpzf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc --exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run / OS backed up

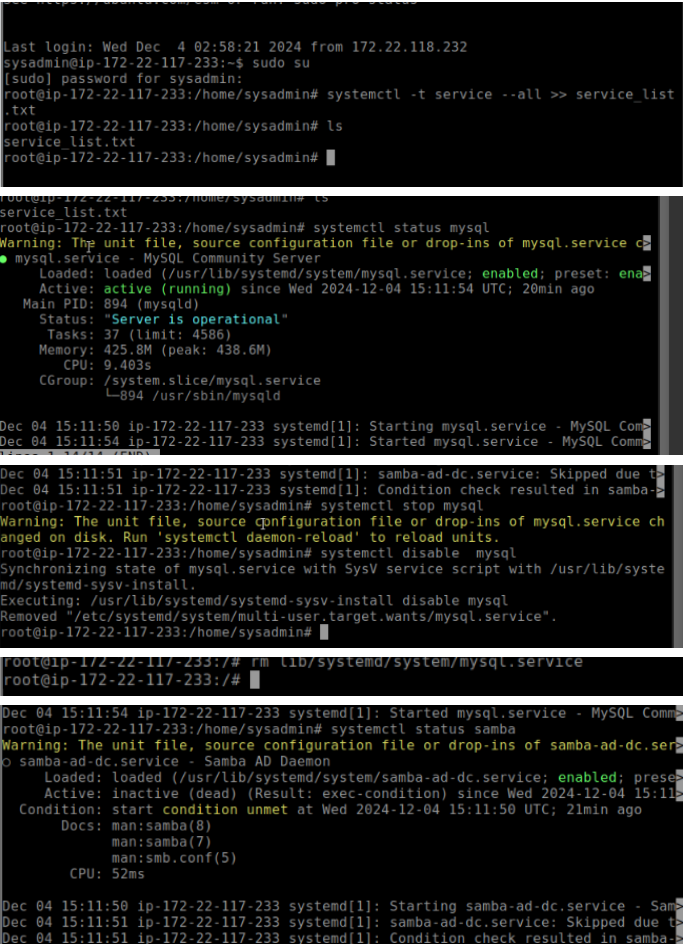
		<pre>*** System restart required *** Last login: Tue Dec 3 01:20:48 2024 from 172.22.118.117 sysadmin@ip-172-22-117-233:~\$ sudo su [sudo] password for sysadmin: root@ip-172-22-117-233:/home/sysadmin# cd / root@ip-172-22-117-233:/# ls baker_street_backup.tar.gz etc lost+found root srv bin home media run sys bin.usr-is-merged lib mnt sbin tmp boot lib.usr-is-merged opt sbin.usr-is-merged usr dev lib64 proc snap var root@ip-172-22-117-233:/#</pre>
<input checked="" type="checkbox"/>	Auditing users and groups	<p>deluser --remove-home for lestrade, irene, mary, gregson</p> <p>usermod -L for moriarty and mrs_hudson</p> <p>usermod -d to remove password for adler and toby</p> <p>Addgroup research</p> <p>usermod -G research mycroft</p> <p>delgroup marketing</p> <pre>0:99999:7::: moriarty:!!\$ys\$j9T\$8xH. 18:0:99999:7::: mycroft:\$ys\$j9T\$FGTw.R :0:99999:7::: mrs_hudson:!:20018:0:9 toby:!:20018:0:99999:7 adler:!:20018:0:99999</pre> <pre>chrony:!:19993::: ubuntu:!:20018:0:99999:7::: sysadmin:\$ys\$j9T\$y5SnyjKCr6hWMrZx999c/\$WtLVq8XjHyejw/YD3rna0wrrvkn0Q4fwDcfxWvj9wsE 8:0:99999:7::: sherlock:\$ys\$j9T\$.hQtR8196eLfFRva.Zq911snSTABLizRMcz/Ge.WyAZ4ksUA7v7LchfXsd48EveDJ. 8:0:99999:7::: watson:\$ys\$j9T\$Lxq4xg30CPuFFv9ZWjug8/\$CoTk253eWAaE/7ChaPi8m9gqoLJ14yAgJJz15tzN.U.:2 0:99999:7::: moriarty:!!\$ys\$j9T\$8xH.cxMRyMYWUfmFfQ681.\$BXMmzQWrcVdVDxJhbJjnA1kbtMw980XtJb9YazuAhC 18:0:99999:7::: mycroft:\$ys\$j9T\$FGTw.R.lTUqxnEu31T///.\$NAK6Z/d/C0shB1f52qUKdtTL/RAftEjHLTGC0m6p000: :0:99999:7::: mrs_hudson:!:20018:0:99999:7::: toby:!:20018:0:99999:7::: adler:!:20018:0:99999:7:::</pre> <pre>sysadmin@ip-172-22-117-233:~\$ sudo su [sudo] password for sysadmin: root@ip-172-22-117-233:/home/sysadmin# cd / root@ip-172-22-117-233:/# usermod -G research mycroft root@ip-172-22-117-233:/# id mycroft uid=1005(mycroft) gid=1008(mycroft) groups=1008(mycroft),1009(research) root@ip-172-22-117-233:/# delgroup marketing Removing group 'marketing' ... root@ip-172-22-117-233:/#</pre>

<input checked="" type="checkbox"/>	Updating and enforcing password policies	<p>nano /etc/pam.d/common-password</p> <p>Password requisite pam-pwquality.so retry=3 minlen=10 ucredit=1 lcredit=1 ocredit=1</p> <p>sudo usermod -p <password> <username> for toby and adler</p> <p>chage -d 0 for tony and adler</p> <pre> password [success=1 default=ignore] pam_unix.so obscure md5 here's the fallback if no module succeeds password [success=1 default=ignore] pam_unix.so obscure md5 prime the stack with a positive return value if there isn't one already; this avoids us returning an error just because nothing sets a success code since the modules above will each just jump around password [success=1 default=ignore] pam_unix.so obscure md5 and here are more per-package modules (the "Additional" block) here are the requirements for passwords password requisite pam_pwquality.so retry=3 minlen=10 ucredit=1 lcredit=1 ocredit=1 </pre> <pre> sysadmin@ip-172-22-117-233:~\$ sudo usermod -p toby123 toby sysadmin@ip-172-22-117-233:~\$ sudo passwd -S toby toby P 2024-12-03 0 99999 7 -1 sysadmin@ip-172-22-117-233:~\$ sudo usermod -p Adler123 adler [sudo] password for sysadmin: sysadmin@ip-172-22-117-233:~\$ sudo su root@ip-172-22-117-233:/home/sysadmin# chage -d 0 toby root@ip-172-22-117-233:/home/sysadmin# chage -d 0 adler root@ip-172-22-117-233:/home/sysadmin# </pre>
<input checked="" type="checkbox"/>	Updating and enforcing sudo permissions	<p>Sudo visudo</p> <p>cat /etc/sudoers</p> <pre> # See sudoers(5) for more information on "@include" directives: @includedir /etc/sudoers.d sysadmin ALL=(ALL:ALL) ALL sysadmin ALL=(ALL:ALL) ALL sherlock ALL=(ALL) NOPASSWD:ALL #watson ALL=(ALL) NOPASSWD:ALL #moriarty ALL=(ALL) NOPASSWD:ALL sysadmin ALL=(ALL:ALL) ALL #Watson and Mycroft have sudo privileges to run /var/log/logcleanup.sh watson ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh mycroft ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh #All employees in research group have sudo privileges to run research ALL=(ALL) NOPASSWD: /tmp/scripts/research_script.sh </pre>
<input checked="" type="checkbox"/>	Validating and updating permissions on files and directories	<p>chmod -R o -rwx <directory_name></p> <p>find /home/* -type f -iname "*" -exec chown :<group_name> { } +</p> <p>find /home/* -type f -iname "*" -exec chmod 070 { } \;</p> <p>Located file password file via cat command</p> <p>Ran tree command on home directory to find that file in other directories</p> <p>rm my_file.txt</p>

		<pre> root@ip-172-22-117-233:~# cd / root@ip-172-22-117-233:/# find /home/* -type f -iname "**engineering*" -exec chown :engineering {} + root@ip-172-22-117-233:/# find /home/* -type f -iname "**engineering*" -exec chmod 070 {} \; sudo: password for sysadmin: root@ip-172-22-117-233:/home/sysadmin# cd / root@ip-172-22-117-233:/# find /home/* -type f -iname "**research*" -exec chown :research {} + find: missing argument to '-exec' root@ip-172-22-117-233:/# find /home/* -type f -iname "**research*" -exec chown 4:research {} + root@ip-172-22-117-233:/# find /home/* -type f -iname "**research*" -exec chmod 070 {} \; root@ip-172-22-117-233:/# find /home/* -type f -iname "**finance*" -exec chown 4:finance {} + root@ip-172-22-117-233:/# find /home/* -type f -iname "**finance*" -exec chmod 070 {} \; root@ip-172-22-117-233:/# root@ip-172-22-117-233:/home/moriarty# cat my_file.txt user1: Password123 user2: qwerty!@# user3: letmein456 root@ip-172-22-117-233:/home/moriarty# cd root@ip-172-22-117-233:~# cd / 0 directories, 45 files root@ip-172-22-117-233:/home# cd sherlock/ root@ip-172-22-117-233:/home/sherlock# rm my_file.txt root@ip-172-22-117-233:/home/sherlock# ls deduction.doc 3.txt deduction.doc_script2.sh game_is_afoot.txt 1.txt deduction.doc_script1.sh elementary.txt 0.txt game_is_afoot.txt 2.txt root@ip-172-22-117-233:/home/sherlock# cd .. root@ip-172-22-117-233:/home# cd watson root@ip-172-22-117-233:/home/watson# ls finance_script.sh 3.txt deduction.doc 0.txt my_file.txt finance_script.sh_script1.sh deduction.doc 1.txt finance_script.sh_script2.sh deduction.doc 2.txt root@ip-172-22-117-233:/home/watson# rm my_file.txt root@ip-172-22-117-233:/home/watson# ls finance_script.sh 3.txt Finance_script.sh_script2.sh deduction.doc 1.txt finance_script.sh_script1.sh deduction.doc 0.txt deduction.doc 2.txt root@ip-172-22-117-233:/home/watson# cd root@ip-172-22-117-233:/home/moriarty# ls Finance_script.sh 0.txt game_is_afoot.txt 3.txt my_file.txt Finance_script.sh 2.txt game_is_afoot.txt_script1.sh elementary.txt 1.txt game_is_afoot.txt_script2.sh root@ip-172-22-117-233:/home/moriarty# rm my_file.txt root@ip-172-22-117-233:/home/moriarty# ls Finance_script.sh 0.txt elementary.txt 1.txt game_is_afoot.txt_script1.sh Finance_script.sh 2.txt game_is_afoot.txt 3.txt game_is_afoot.txt_script2.sh root@ip-172-22-117-233:/home/moriarty# root@ip-172-22-117-233:/home# chmod -R o-rwx mrs_hudson root@ip-172-22-117-233:/home# chmod -R o-rwx mycroft root@ip-172-22-117-233:/home# chmod -R o-rwx sherlock root@ip-172-22-117-233:/home# chmod -R o-rwx toby root@ip-172-22-117-233:/home# chmod -R o-rwx watson root@ip-172-22-117-233:/home# ls -l total 36 -rwxr-x--- 2 adler adler 4096 Oct 22 16:36 adler -rwxr-x--- 2 moriarty moriarty 4096 Oct 22 16:35 moriarty -rwxr-x--- 2 mrs_hudson mrs_hudson 4096 Oct 22 16:35 mrs_hudson -rwxr-x--- 2 mycroft mycroft 4096 Oct 22 16:35 mycroft -rwxr-x--- 2 sherlock sherlock 4096 Oct 22 16:35 sherlock -rwxr-x--- 3 sysadmin sysadmin 4096 Dec 4 16:00 sysadmin -rwxr-x--- 2 toby toby 4096 Oct 22 16:36 toby -rwxr-x--- 3 ubuntu ubuntu 4096 Oct 22 16:32 ubuntu -rwxr-x--- 2 watson watson 4096 Oct 22 16:35 watson root@ip-172-22-117-233:/home# </pre>
<input checked="" type="checkbox"/>	Optional: Updating password hashing configuration	No instructions were given for this in the Day 1 Guide

<input checked="" type="checkbox"/>	Auditing and securing SSH	<pre> Nano /etc/ssh/sshd_config # OpenSSH is to specify options with # possible, but leave them commented # default value. #Include /etc/ssh/sshd_config.d/*.conf Port 22 #AddressFamily any #ListenAddress 0.0.0.0 #ListenAddress :: #HostKey /etc/ssh/ssh_host_rsa_key #HostKey /etc/ssh/ssh_host_ecdsa_key #HostKey /etc/ssh/ssh_host_ed25519_key #Match User anoncvs # X11Forwarding no # AllowTcpForwarding no # PermitTTY no # ForceCommand cvs server Protocol 2 # Do not change #IgnoreUserKnownHosts no # Don't read the user's ~/.rhosts and #IgnoreRhosts yes # To disable tunneled clear text passwords PasswordAuthentication yes PermitEmptyPasswords no # Change to yes to enable challenge-response # some PAM modules and threads) KbdInteractiveAuthentication no # Authentication: #LoginGraceTime 2m #PermitRootLogin no #StrictModes yes #MaxAuthTries 6 #MaxSessions 10 </pre>
-------------------------------------	---------------------------	--

<input checked="" type="checkbox"/>	<p>Reviewing and updating system packages</p>	<pre>apt update</pre> <pre>apt upgrade -y</pre> <pre>apt list --installed >> package_list.txt</pre> <pre>grep telnet package_list.txt</pre> <pre>grep rsh_client package_list.txt</pre> <p>Telnet transmits data in plain text, including passwords and usernames. The data is unencrypted, uses simple authentication and is run over a well known port (23) which makes it vulnerable to attacks.</p> <p>Rsh-client also transmits unencrypted data and its authentication is weak, relying on hostnames and user ids. It is also outdated and does not check the integrity of the data being transmitted.</p> <p>Ufw helps with hardening with its firewall management, traffic filtering, logging, limiting connection attempts, etc</p> <p>Lynis helps by providing system-wide auditing, hardening suggestions, vulnerability detection, compliance testing, etc.</p> <p>Tripwire helps by enforcing policy, monitoring file integrity, alerting on unauthorized changes, logging and reporting, etc.</p> <pre># Do not change AllowUsers sherlock watson moriarty mycroft irene lestrade sy root@ip-172-22-117-233:~# apt update Get:1 http://security.ubuntu.com/ubuntu noble-security InRele Reading state information... Done 30 packages can be upgraded. Run 'apt list --upgradab root@ip-172-22-117-233:~# apt upgrade -y Reading package lists... Done Building dependency tree... Done root@ip-172-22-117-233:~# cd / root@ip-172-22-117-233:/# apt list --installed >> package_list.txt WARNING: apt does not have a stable CLI interface. Use with caution in ripts. root@ip-172-22-117-233:/# apt install ufw Reading package lists... Done Building dependency tree... Done Reading state information... Done ufw is already the newest version (0.36.2-6). ufw set to manually installed. 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded. root@ip-172-22-117-233:/#</pre>
-------------------------------------	---	--

		<pre> root@ip-172-22-117-233:/# grep telnet package list.txt inetutils-telnet/noble,now 2:2.5-3ubuntu4 amd64 [installed,automatic] telnet/noble,now 0.17+2.5-3ubuntu4 all [installed] inetutils-telnet/noble,now 2:2.5-3ubuntu4 amd64 [installed,automatic] telnet/noble,now 0.17+2.5-3ubuntu4 all [installed] root@ip-172-22-117-233:/# inetutils-telnet/noble,now 2:2.5-3ubuntu4 amd64 [installed,automatic] telnet/noble,now 0.17+2.5-3ubuntu4 all [installed] root@ip-172-22-117-233:/# apt remove telnet Reading package lists... Done Building dependency tree... Done Reading state information... Done The following packages will be REMOVED: telnet 0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded. After this operation, 48.1 kB disk space will be freed. Do you want to continue? [Y/n] </pre>
<input checked="" type="checkbox"/>	Disabling unnecessary services	<pre> systemctl -t service --all >> service_list.txt systemctl status <Service_name> systemctl stop mysql systemctl disable mysql rm /lib/systemd/system/mysql.service </pre> 

<input checked="" type="checkbox"/>	<p>Enabling and configuring logging</p>	<pre>nano /etc/systemd/journald.conf</pre> <pre>nano /etc/logrotate.conf</pre> <pre>systemctl restart systemd-journald</pre>  <pre>GNU nano 7.2 journald.conf * # This file is part of systemd. # # systemd is free software; you can redistribute it and/or modify it under the # terms of the GNU Lesser General Public License as published by the Free # Software Foundation; either version 2.1 of the License, or (at your option) # any later version. # # Entries in this file show the compile time defaults. Local configuration # should be created by either modifying this file (or a copy of it placed in # /etc/ if the original file is shipped in /usr/), or by creating "drop-ins" in # the /etc/systemd/journald.conf.d/ directory. The latter is generally # recommended. Defaults can be restored by simply deleting the main # configuration file and all drop-ins located in /etc/. # # Use 'systemd-analyze cat-config systemd/journald.conf' to display the full conf # # See journald.conf(5) for details. [Journal] Storage=persistent #Compress=yes #Seal=yes #SplitMode=uid #SyncIntervalSec=5m #RateLimitIntervalSec=30s #RateLimitBurst=10000 #SystemMaxUse=300M #SystemKeepFree= #SystemMaxFileSize= #SystemMaxFiles=100 #RuntimeMaxUse= #RuntimeKeepFree= #RuntimeMaxFileSize= #RuntimeMaxFiles=100 #MaxRetentionSec= GNU nano 7.2 logrotate.conf * # see "man logrotate" for details # global options do not affect preceding include directives # rotate log files daily daily # use the adm group by default, since this is the owning group # of /var/log/. su root adm # keep 7 days worth of backlogs rotate 7 # create new (empty) log files after rotating old ones create # use date as a suffix of the rotated file #dateext # uncomment this if you want your log files compressed #compress # packages drop log rotation information into this directory root@ip-172-22-117-233:/etc# systemctl restart systemd-journald</pre>
<input checked="" type="checkbox"/>	<p>Scripts created</p>	<pre>/home/sysadmin/hardening_script1.sh</pre> <pre>/home/sysadmin/hardening_script2.sh</pre>


```

printf "\n" >> $REPORT_FILE

# Force Sherlock, Watson, and Mycroft to change their password upon their next login
echo "Forcing Sherlock, Watson, and Mycroft users to change their password on next login"
# Placeholder for command to force password change

chage -d 0 sherlock
chage -d 0 watson
chage -d 0 mycroft

echo "Password change enforced for Sherlock, Watson, and Mycroft." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Output the sudoers file to the report
echo "Gathering sudoers file..."
# Placeholder for command to output sudoers file
echo "Sudoers file:$(cat /etc/sudoers)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Output memory information
echo "Gathering memory information..."
# Placeholder for command to get memory info
echo "Memory Information: $(free -h)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Output uptime information
echo "Gathering uptime information..."
# Placeholder for command to get uptime info
echo "Uptime Information: $(uptime)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Backup the OS
echo "Backing up the OS..."
# Placeholder for command to back up the OS

baker_street_backup.tar.gz

# Script to check for files with world permissions and update them
echo "Checking for files with world permissions..."

chmod -R o-rwx /home/*

# Placeholder for command to find and update files with world permissions
echo "World permissions have been removed from any files found." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

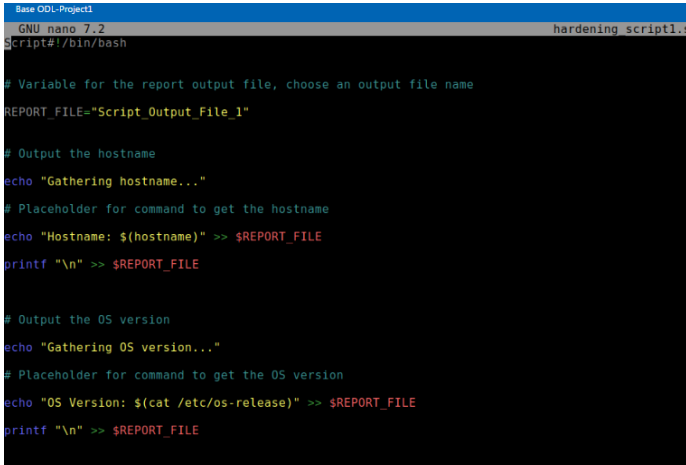
# Find specific files and update their permissions
echo "Updating permissions for specific scripts..."

# Engineering scripts - Only members of the engineering group
echo "Updating permissions for Engineering scripts."

# Placeholder for command to update permissions
find /home/* -type f -iname "**engineering*" -exec chown :engineering {} +
find /home/* -type f -iname "**engineering*" -exec chmod 070 {} \;

echo "Permissions updated for Engineering scripts." >> $REPORT_FILE

```

		<pre>printf "\n" >> \$REPORT_FILE # Research scripts - Only members of the research group echo "Updating permissions for Research scripts..." # Placeholder for command to update permissions find /home/* -type f -iname "*research*" -exec chown :research {} + find /home/* -type f -iname "*research*" -exec chmod 070 {} \; echo "Permissions updated for Research scripts" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Finance scripts - Only members of the finance group echo "Updating permissions for Finance scripts" # Placeholder for command to update permissions find /home/* -type f -iname "*finance*" -exec chown :finance {} + find /home/* -type f -iname "*finance*" -exec chmod 070 {} \; echo "Permissions updated for Finance scripts." >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE</pre>  <pre>Base ODL-Project1 GNU nano 7.2 hardening_script1.s Script#1:/bin/bash # Variable for the report output file, choose an output file name REPORT_FILE="Script_Output_File_1" # Output the hostname echo "Gathering hostname..." # Placeholder for command to get the hostname echo "Hostname: \$(hostname)" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Output the OS version echo "Gathering OS version..." # Placeholder for command to get the OS version echo "OS Version: \$(cat /etc/os-release)" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Placeholder for command to upgrade packages apt upgrade -y echo "Packages have been updated and upgraded" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Placeholder for command to list all installed packages echo "Installed Packages:\$(apt list --installed)" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE echo "Printing out logging configuration data" # Placeholder for command to display logging data echo "journald.conf file data: \$(cat etc/systemd/journald.conf)" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Placeholder for command to display logrotate data echo "logrotate.conf file data:\$(cat /etc/logrotate.conf)" >> \$REPORT_FILE</pre>
--	--	--

		<pre>#!/bin/bash # Variable for the report output file, choose a NEW output file name REPORT_FILE="Script_Output_File_2" # Output the sshd configuration file echo "Gathering details from sshd configuration file" # Placeholder for command to get the sshd configuration file echo "sshd configuration file:\$(cat /etc/ssh/sshd_config)" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Update packages and services Echo "Updating packages and services" # Placeholder for command to update packages apt update</pre>
<input checked="" type="checkbox"/>	Scripts scheduled with cron	<p>Crontab -e (Syntax generated via crontab-generator.org)</p> <pre># Edit this file to introduce tasks to be run by cron. # # Each task to run has to be defined through a single line # indicating with different fields when the task will be run # and what command to run for the task # # To define the time you can provide concrete values for # minute (m), hour (h), day of month (dom), month (mon), # and day of week (dow) or use '*' in these fields (for 'any'). # # Notice that tasks will be started based on the cron's system # daemon's notion of time and timezones. # # Output of the crontab jobs (including errors) is sent through # email to the user the crontab file belongs to (unless redirected). # # For example, you can run a backup of all your user accounts # at 5 a.m every week with: # 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/ # # For more information see the manual pages of crontab(5) and cron(8) # # m h dom mon dow command 0 0 1 * * /home/sysadmin/hardening_script1.sh >/dev/null 2>&1 0 0 1 * 1 /home/sysadmin/hardening_script2.sh >/dev/null 2>&1</pre>