МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп’ютерних систем та інфокумунікацій

Кафедра комп’ютерних систем, мереж і кібербезпеки

Лабораторна робота №6

З диципліни: «Технології DevOps»

Виконав:

студент 5 курсу групи №555 ім

Напряму підготовки

125 Кібербезпека та захист інформації

ст. Орлов Станіслав Валерійович

Прийняв:

к.т.н., доцент

Узун Дмитро Дмитрович

Харків, 2023

Постановка завдання:

A. Create a script that uses the following keys:

1. When starting without parameters, it will display a list of possible keys and their description.

2. The --all key displays the IP addresses and symbolic names of all hosts in the current subnet

3. The --target key displays a list of open system TCP ports.

The code that performs the functionality of each of the subtasks must be placed in a separate function

B. Using Apache log example create a script to answer the following questions:

1. From which ip were the most requests?

2. What is the most requested page?

3. How many requests were there from each ip?

4. What non-existent pages were clients referred to?

5. What time did site get the most requests?

6. What search bots have accessed the site? (UA + IP)

C. Create a data backup script that takes the following data as parameters:

1. Path to the syncing directory.

2. The path to the directory where the copies of the files will be stored.

In case of adding new or deleting old files, the script must add a corresponding entry to the log file indicating the time, type of operation and file name. [The command to run the script must be added to crontab with a run frequency of one minute]

Вирішення завдання

A. **Create a script that uses the following keys:**

1. When starting without parameters, it will display a list of possible keys and their description.

2. The --all key displays the IP addresses and symbolic names of all hosts in the current subnet

3. The --target key displays a list of open system TCP ports.

Скрипт виконуючий завдання наведено нижче:

```
#!/usr/bin/env sh
set -e

main() {
  parse_args "$@"
}

parse_args() {
  if [ $# -eq 0 ]; then
    print_help
  fi
  INVOKE_ALL=false
  INVOKE_TARGET=false
  CIDR=""
  TARGET=""
  while [ -n "$1" ]; do
    case "$1" in
      --help)
        print_help
        shift;;
      --cidr)
        if [ -z "$2" ]; then
          echo "error: key does not have arg" >&2
          exit 1
        fi
        export CIDR="$2"
        shift 2
        ;;
      --target)
        if [ -z "$2" ]; then
          echo "error: key does not have arg" >&2
          exit 1
        fi
        export TARGET="$2"
        INVOKE_TARGET=true
        shift 2
        ;;
      --all)
        INVOKE_ALL=true
        shift;;
      *)
        print_help
        shift;;
    esac
  done
  if [ "${INVOKE_ALL}" = "true" ] && [ "${INVOKE_TARGET}" = "true" ]; then
    echo "error: you cannot use both --all and --target at the same time" >&2
    exit 1
  elif [ "${INVOKE_ALL}" = "true" ]; then
    print_all_hosts "${CIDR}"
  elif [ "${INVOKE_TARGET}" = "true" ]; then
    scan_target "${TARGET}"
  fi
}

print_help() {
  cat <<EOF
# ┬─┐┬─┐┐ ┬┌─┐┬─┐┐─┐
# │ │├─│┌┘│ ││─┘└─┐
# ┘─┘┴─┘└┘ ┘─┘┘ ──┘

--all - key displays the IP addresses and symbolic names of all hosts in the current subnet.
--cidr <CIDR> - CIDR subnetwork to scan.
--target <IP_OR_HOST> - key displays a list of open system TCP ports.
EOF
  exit 0
}

print_all_hosts() {
  CIDR="${1}"
  if [ -z "${CIDR}" ]; then
    echo "error: CIDR is empty, make sure you have passed --cidr flag" >&2
    exit 2
  fi
  echo "info: Scanning for hosts in the local network ${CIDR}"
  nmap -sn "${CIDR}" | awk '/Nmap scan report for/{print $5}'
}

scan_target() {
  TARGET="${1}"
  if [ -z "${TARGET}" ]; then
    echo "error: TARGET is empty, make sure you have passed --target flag" >&2
    exit 2
  fi
  echo "info: Scanning target ${TARGET} for open TCP ports"
  nmap -sT Normal -np 0-65535 "${TARGET}"
}

main "$@"
```

B. Using Apache log example create a script to answer the following questions:

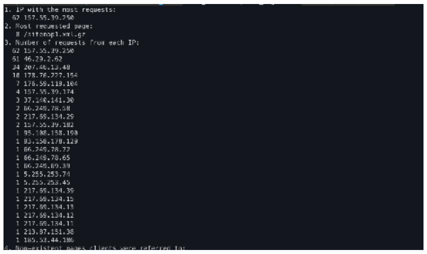
1. From which ip were the most requests?

176.120.103.194

2. What is the most requested page?

/sitemap1.xml.gz

3. How many requests were there from each ip?



```
1. IP with the most requests:
  62 157.55.39.250
2. Most requested page:
   8 /sitemap1.xml.gz
3. Number of requests from each IP:
  62 157.55.39.250
  61 46.29.2.62
  34 207.46.13.48
  18 178.76.227.154
   7 178.59.119.104
   4 157.55.39.174
   3 37.140.141.30
   2 66.249.78.58
   2 217.69.134.29
   2 157.55.39.182
   1 95.108.158.190
   1 93.158.178.129
   1 66.249.78.72
   1 66.249.78.65
   1 66.249.69.39
   1 5.255.253.74
   1 5.255.253.45
   1 217.69.134.39
   1 217.69.134.15
   1 217.69.134.13
   1 217.69.134.12
   1 217.69.134.11
   1 213.87.151.38
   1 185.53.44.186
4. Non-existent pages clients were referred to:
```

4. What non-existent pages were clients referred to?

Таких сторінок немає

5. What time did site get the most requests?

Усі порівну

```
5 [30/Sep/2015:02:10:46
5 [30/Sep/2015:00:42:46
5 [30/Sep/2015:02:25:50
5 [30/Sep/2015:02:25:52
5 [30/Sep/2015:02:25:53
5 [30/Sep/2015:02:25:54
5 [30/Sep/2015:02:26:23
5 [30/Sep/2015:02:26:24
5 [30/Sep/2015:02:26:51
5 [30/Sep/2015:02:26:52
```

6. What search bots have accessed the site? (UA + IP)

Скрипт:

```
#!/usr/bin/env sh
set -ex

awk '{print $1}' apache.log | uniq -c | sort -k1nr | head -10
grep -oE 'GET [^ ]+' apache.log | awk '{print $2}' | sort | uniq -c | sort -k1nr | head -10
awk '{print $1}' apache.log | sort | uniq -c | sort -k1nr
awk '$9~/404/{print $7}' apache.log
awk '{print $4}' apache.log | sort | uniq -c | sort -k1nr | head -10

cut -d '"' -f 6 apache.log | grep -i 'bot' | sort -u | \
while IFS= read -r UA; do
  grep -F "${UA}" apache.log
done

awk -F '"' '$6~/bot/{print $0}' apache.log
```

Результат виконання:

```
Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/Fast/2.0; +http://go.mail.ru/help/robots)
Mozilla/5.0 (compatible; XoviBot/2.0; +http://www.xovibot.net/)
Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)
Mozilla/5.0 (iPhone; CPU iPhone OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Version/7.0 Mobile/11A465 Safari/9537.53
(compatible; bingbot/2.0;  http://www.bing.com/bingbot.htm)
```

Mozilla/5.0 (iPhone; CPU iPhone OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Version/7.0 Mobile/11A465 Safari/9537.53 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_1 like Mac OS X; en-us) AppleWebKit/532.9 (KHTML, like Gecko) Version/4.0.5 Mobile/8B117 Safari/6531.22.7 (compatible; Mediapartners-Google/2.1; +http://www.google.com/bot.html)

C. Create a data backup script that takes the following data as parameters:

1. Path to the syncing directory.

2. The path to the directory where the copies of the files will be stored.

Скрипппт:

```
#!/usr/bin/env sh
set -e

export TEMP="$(mktemp -d)"
export SRC="${TEMP}/src"
export DEST="${TEMP}/dest"

main() {
  init
  sync
  modify
  sync
  clean
}

init() {
  cat <<EOF
#  o┌┐┐o┌┐┐
#  ||||| |
#  ┘┘ └┘┘ ┘
EOF
  mkdir -p "${SRC}" "${DEST}"
  cd "${SRC}"
  seq 1 20 | xargs -I{} sh -c 'date -Ins > {}'
}

sync () {
  cat <<EOF
# ┐─┐┐ ┬┌┐┐┌─┐
#  └─┐└┌┘||||
# ──┘ ┘ ┘ └┘ └─┘
EOF
  cd "${SRC}"
  find ./ -type f | \
  while IFS= read -r SRC_FILE; do
    SRC_CHECKSUM="$(md5sum "${SRC_FILE}" | awk '{print $1}' | tr -d '[:space:]')"
    DEST_FILE="${DEST}/${SRC_FILE}"
    DEST_CHECKSUM="$(md5sum "${SRC_FILE}" | awk '{print $1}' | tr -d '[:space:]')"
    if [ ! -f "${DEST_FILE}" ] || [ "${SRC_CHECKSUM}" != "${DEST_CHECKSUM}" ]; then
      echo "info: copying ${SRC_FILE}"
      cp -pf "${SRC_FILE}" "${DEST_FILE}"
    else
      echo "skip: ${SRC_FILE}"
    fi
  done

  cd "${DEST}"
  find ./ -type f | \
  while IFS= read -r DEST_FILE; do
```

```
    SRC_FILE="${SRC}/${DEST_FILE}"
    if [ ! -f "${SRC_FILE}" ]; then
      rm -vf "${DEST_FILE}"
    fi
  done
}

modify() {
  cat <<EOF
#  ┌┐ ┌─┐┬─┐o┬─┐┐ ┬
#  ││││ ││ ││├─ └┌┘
# ┘ ┘┘─┘┘─┘┘┘  ┘
EOF
  cd "${SRC}"
  ls | sort -R | head -10 | sort -n | xargs rm -rvf
}

clean() {
  cat <<EOF
#  ┌─┐┬ ┬─┐┬─┐┌┐┐
#  │ │ ├─ │─┤│││
#  └─┘┘─┘┴─┘┘ ┘┘└┘
EOF
  rm -rvf "${TEMP}"
}

Main
```

Результат виконання:

```
info: Copying files from /tmp/tmp.dsDkecGXch/src to /tmp/tmp.dsDkecGXch/dest
info: copying ./20
info: copying ./19
info: copying ./18
info: copying ./17
info: copying ./16
info: copying ./15
info: copying ./14
info: copying ./13
info: copying ./12
info: copying ./11
info: copying ./10
info: copying ./9
info: copying ./8
info: copying ./7
info: copying ./6
info: copying ./5
info: copying ./4
info: copying ./3
info: copying ./2
info: copying ./1
removed '1'
removed '2'
removed '3'
removed '4'
removed '5'
removed '8'
removed '9'
removed '13'
removed '17'
removed '18'
skip: ./20
skip: ./19
skip: ./16
skip: ./15
```

```
skip: ./14
skip: ./12
skip: ./11
skip: ./10
skip: ./7
skip: ./6
removed './1'
removed './2'
removed './3'
removed './4'
removed './5'
removed './8'
removed './9'
removed './13'
removed './17'
removed './18'
removed '/tmp/tmp.dsDkecGXch/dest/6'
removed '/tmp/tmp.dsDkecGXch/dest/7'
removed '/tmp/tmp.dsDkecGXch/dest/10'
removed '/tmp/tmp.dsDkecGXch/dest/11'
removed '/tmp/tmp.dsDkecGXch/dest/12'
removed '/tmp/tmp.dsDkecGXch/dest/14'
removed '/tmp/tmp.dsDkecGXch/dest/15'
removed '/tmp/tmp.dsDkecGXch/dest/16'
removed '/tmp/tmp.dsDkecGXch/dest/19'
removed '/tmp/tmp.dsDkecGXch/dest/20'
removed directory '/tmp/tmp.dsDkecGXch/dest'
removed '/tmp/tmp.dsDkecGXch/src/20'
```

Висновки

У ході виконання лабораторної роботи оволодів основними навичками роботи з мовою виконная скриптів BASH та використав їх для вирішення типов задач роботи в мережою та файловою системою.