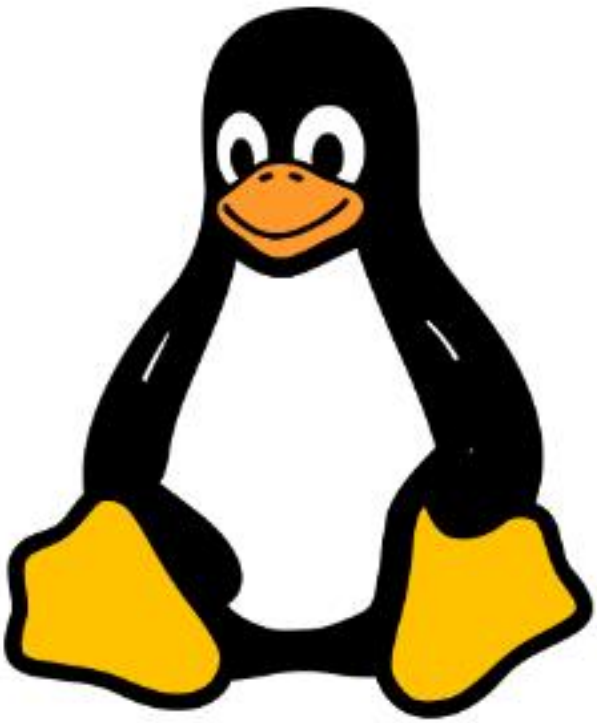
Shell Script Project



5조

김승훈 오주현 이동민 최낙원 최원석

■ 목차

1. 프로젝트 개요

1) 쉘 스크립트 운영 개요 및 목적

서버 운영의 안정화를 위해 시스템 및 보안 취약점을 주기적으로 점검해야 한다. 점검 작업을 효율적으로 하기위해 쉘 스크립트를 작성해야 하며, 쉘 스크립트를 활용하여 효율적으로 운영하기 위해서는 상황에 맞게 스크립트를 사용해야 한다.

(1) 시스템 점검 쉘 스크립트

- 월별 유지보수 시 실행
- 기간별 정기적인 실행을 통하여 지속적인 분석 가능
- 프로세스, 디스크, 파일 및 디렉터리, 서비스 별 세부적인 분할을 통한 세밀한 점검 가능

(2) 보안 점검 쉘 스크립트

- 월별 유지보수 시 실행
- 기간별 정기적인 실행 통하여 지속적인 분석 가능
- 계정 관리 및 SSH 접근, 취약점 점검 등의 보안 점검 가능

(3) 패키지 설치 쉘 스크립트

- 특정 업무 수행 시 실행
- EX) 신규 입사자 PC 세팅 / 신규 서비스 세팅

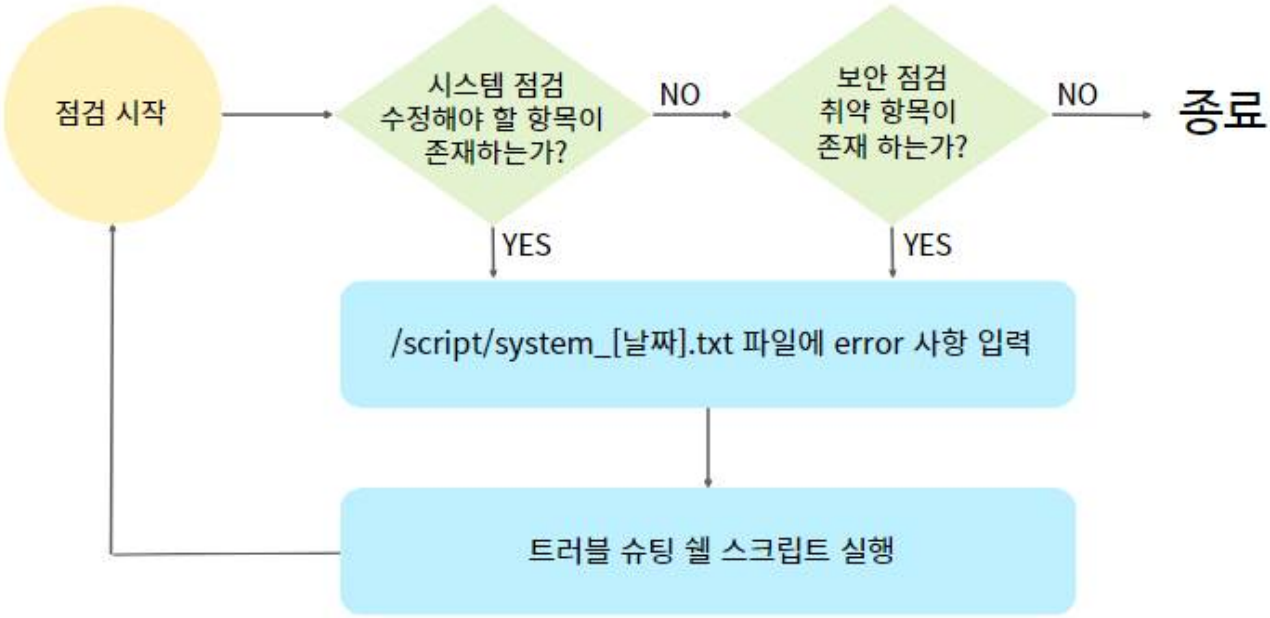
(4) 백업 쉘 스크립트

- 특정 업무 수행 시 실행
- EX) 사옥 이전 / 사용 환경 마이그레이션 / 위험도 높은 테스트 진행 전

(5) 트러블 슈팅 쉘 스크립트

- 특정 이슈 및 장애 발생 시 실행
- 특정 디렉터리 및 파일 중 변경된 접근 권한 정상화
- 특정 주요 서비스 중 비활성화 서비스 활성화
- 계정 관리 최적화

2) 쉘 스크립트 동작 흐름도



2. 시스템 점검 쉘 스크립트

1) 시스템 점검 스크립트 체크 리스트

분류	세부항목
프로세스 관리	동작 중인 프로세스
	평균 시스템 부하
	CPU 사용량 점검
	사용중인 메모리
	네트워크 세션 목록 및 상태
디스크 관리	디스크별 파티셔닝 항목
	디스크별 사용률
	디스크 통계 정보
	지정 사용률 초과 항목 체크
파일 및 디렉터리 관리	주요 파일 존재 여부 확인
	주요 파일 접근 권한 확인
	기본 허가권 설정 확인
	파일 용량 확인
서비스 관리	활성화 서비스 리스트
	비활성화 서비스 리스트
	특정 서비스 확인 - 중요 서비스 위주
	서비스 포트 리스트

2) 쉘 스크립트 코드

```
today=`date "+%Y_%m_%d"`
user=$(whoami)

function TEXT() {
if [ -f /script/${user}_${today}_system.txt ]; then
    rm -rf /script/${user}_${today}_system.txt
    touch /script/${user}_${today}_system.txt
else
    touch /script/${user}_${today}_system.txt
fi
}

function SYS_CK() {

services1=$(systemctl list-unit-files --type=service | grep enabled | awk '{print $1}')
services2=$(systemctl list-unit-files --type=service | grep disabled | awk '{print $1}')
services3=$(systemctl list-units --type=service | grep named | awk '{print $1}')
services4=$(lsof -i -nP | grep LISTEN | awk '{print $(NF-1)" "$1}' | sort -u)



echo ""
echo "-----------------------------------------------"
echo "|     월간 시스템 점검 보고서  |    ${today}    |"
echo "-----------------------------------------------"
echo "|   4조_ 김승훈 오주현 이동민 최낙원 최원석     |"
echo "-----------------------------------------------"
echo "|               | 1-1. 동작중인 프로세스          |"
echo "|               | 1-2. 평균 시스템 부하           |"
echo "| 1. 프로세스 | 1-3. CPU 코어별 사용량          |"
echo "|       관리    | 1-4. 사용중인 메모리            |"
echo "|               | 1-5. 네트워크 세션 목록 및 상태|"
echo "-----------------------------------------------"
echo "|               | 2-1. 디스크별 파티셔닝 항목     |"
echo "| 2. 디스크    | 2-2. 디스크별 사용률            |"
echo "|      관리     | 2-3. 디스크 통계 정보           |"
echo "|               | 2-4. 지정 사용률 초과 항목      |"
echo "-----------------------------------------------"
```

```
echo "|                | 3-1. 주요 파일 존재 여부 확인 및|"
echo "| 3. 파일 및      |        주요 파일 접근 권한 확인   |"
echo "|    디렉터리    | 3-2. 기본 허가권 설정 확인       |"
echo "|       관리      | 3-3. 파일 용량 확인              |"
echo "-----------------------------------------------"
echo "|                | 4-1. 활성화 서비스 리스트        |"
echo "| 4. 서비스      | 4-2. 비활성화 서비스 리스트      |"
echo "|       관리      | 4-3. 특정 서비스 확인            |"
echo "|                | 4-4. 서비스 포트 리스트          |"
echo "-----------------------------------------------"
echo ""
echo ""

echo "-----------------------------------------------"
echo "|        1. 프로세스 관리        |    ${today}    |"
echo "-----------------------------------------------"
echo ""

echo "=============1-1. 동작중인 프로세스============="
echo ""
ps -ef
echo ""

echo "==============1-2. 평균 시스템 부하=============="
echo ""
uptime
echo""

echo "=============1-3. CPU 코어별 사용량============="
echo ""
pidstat -l
echo ""

echo "==============1-4. 사용중인 메모리=============="
echo ""
free -h
echo ""

echo "========1-5. 네트워크 세션 목록 및 상태========"
```

```
echo ""
netstat -a
echo ""

echo ""
echo "-----------------------------------------------"
echo "|          2. 디스크 관리          |    ${today}    |"
echo "-----------------------------------------------"
echo ""

echo "==========2-1. 디스크별 파티셔닝 항목=========="
echo ""
fdisk -l | grep Disk | grep /dev
echo ""

echo "==============2-2. 디스크별 사용률=============="
echo ""
df -h
echo""

echo "==============2-3. 디스크 통계 정보=============="
echo ""
# 체크할 툴 입력
tool=smartctl

# 툴 존재 여부 확인 및 없으면 설치
if which $tool > /dev/null; then
  echo "$tool 이 이미 존재 합니다."
else
  echo "$tool 이 존재하지 않습니다. 설치작업을 실행합니다"
  # 툴 설치
  yum -y install smartmontools
fi
#디스크 정보 확인
disks=$(lsblk -d | awk '{print $1}' | grep -v "NAME")

# use smartctl to get information about each disk
for disk in $disks; do
  echo "Information for disk $disk:"
```

```
  smartctl -a /dev/$disk
  echo ""
done
echo ""

echo "========2-4. 지정 사용률 초과 항목 체크========"
echo ""
threshold=70

# use iostat to get disk utilization information
output=$(iostat -dx 1 2)

# check for errors
if [ $? -ne 0 ]; then
  echo "Error: iostat command failed"
  exit 1
fi

# process the output from iostat
echo "$output" | awk '
  BEGIN {
    found=0
  }
  /Device:/ {
    device=$2
  }
  /%util/ {
    if ($14 > threshold) {
      found=1
      printf "%s utilization is %.1f%% (threshold is %d%%)₩n", device, $14, threshold
    }
  }
  END {
    if (found == 0) {
      printf "No disk utilization exceeded the threshold."
    }
  }
' threshold=$threshold
echo ""
```

```
echo ""
echo "-----------------------------------------------"
echo "|  3. 파일 및 디렉터리  관리 |   ${today}    |"
echo "-----------------------------------------------"
echo ""

function file_chk() {
        sort=`ls -ld "$FILE" 2> /dev/null | awk '{print $1}'| cut -c 1` > /dev/null
        per=`ls -ld "$FILE" 2> /dev/null | awk '{print $1}'` > /dev/null
        if [ "$sort" == "d" ]; then
                        if [ "$per" != "$default" ]; then
                                echo -e  "[위험] $FILE의 Permission 변경하세요!!"
                        fi

        elif [ "$sort" == "-" ]; then
                        if [ "$per" != "default" ]; then
                                echo -e  "[위험] $FILE의 Permission 변경하세요!!"
                        fi

        else
                echo "$FILE does not exist."
        fi
}



echo "==3-1. 주요 파일 존재 여부 및 접근 권한 확인=="
echo ""
FILE=/var/log
default=drwxr-xr-x.
file_chk

FILE=/var/log/messages
default=-rw-r--r--.
file_chk

FILE=/etc/crontab
default=-rw-------.
```

```
file_chk

FILE=/var/log/wtmp
default=-rw-rw----.
file_chk

FILE=/var/log/lastlog
default=-rw--w----.
file_chk

FILE=/etc/passwd
default=-rw-r--r--.
file_chk

FILE=/etc/shadow
default=-rw-------.
file_chk

FILE=/etc/pam.d
default=drwxr-x---.
file_chk

FILE=/etc/hosts.allow
default=-rw-------.
file_chk

FILE=/etc/hosts.deny
default=-rw-------.
file_chk

FILE=/etc/securetty
default=-rw-------.
file_chk

FILE=/etc/security
default=drwx------.
file_chk

FILE=/etc/rc.d/init.d
```

```
default=drwxr-x---.
file_chk

FILE=/etc/sysconfig
default=drwxr-xr-x.
file_chk

FILE=/etc/services
default=-rw-------.
file_chk

FILE=/etc/cron.allow
default=-r--------.
file_chk

FILE=/etc/cron.deny
default=-r--------.
file_chk

FILE=/etc/ssh
default=drwxr-x---.
file_chk

FILE=/etc/sysctl.conf
default=-r--------.
file_chk



echo "============3-2. 기본 허가권 설정 확인============"
echo ""
function umask_chk() {
        cat /etc/profile | grep -i umask | awk '{print $2}' | grep 022 > /dev/null
        if [ $? -eq 0 ]; then
                echo -e "[안전] UMASK OK."
        else
                echo -e "[위험] UMASK 변경하세요."
        fi
}
```

```
umask_chk

echo ""
echo "==============3-3. 파일 용량 확인=============="
echo ""
#100MB가 넘는 파일을 출력한다.
function f_volume() {
        echo "-----------------------------100MB 이상 파일-----------------------------"
        list=`find / -size +100000k -print 2> /dev/null`
        for cnt in $list
        do
                echo "$cnt"
        done

        echo "--------------------------------------------------------------------------------"
}

f_volume

echo ""
echo "-----------------------------------------------"
echo "|          4. 서비스 관리          |    ${today}    |"
echo "-----------------------------------------------"
echo ""

echo "======4-1. 현재 활성화 되어있는 항목 출력======"
echo ""

echo -e "$services1"
echo ""

echo "=======4-2. 현재 비활성화 되어있는 항목======="
echo ""

for service in $services2; do
  if [[ "$service" == "named-chroot.service" || "$service" == "sshd.service" ]]; then
    echo -e "₩033[43;31m*현재 비활성화 = $service *₩033[0m"
  else
    echo "$service"
```

```
   fi
done
echo ""

echo "==========4-3. 필수 서비스 항목=========="
echo ""

echo -e "$services3"
echo ""

echo "==========4-4. 서비스 포트 항목=========="
echo ""

echo -e "$services4"
echo ""

}
TEXT
SYS_CK > /script/${user}_${today}_system.txt
```

3) 실행 결과

 # /script/[사용자]_[날짜]_system.txt

```
-----------------------------------------------------
|    월간 시스템 점검 보고서  |   2023_02_15  |
-----------------------------------------------------
|   4조_ 김승훈 오주현 이동민 최낙원 최원석   |
-----------------------------------------------------
|              | 1-1. 동작중인 프로세스          |
|              | 1-2. 평균 시스템 부하           |
| 1. 프로세스 | 1-3. CPU 코어별 사용량          |
|       관리   | 1-4. 사용중인 메모리            |
|              | 1-5. 네트워크 세션 목록 및 상태|
-----------------------------------------------------
|              | 2-1. 디스크별 파티셔닝 항목     |
| 2. 디스크    | 2-2. 디스크별 사용률            |
|      관리    | 2-3. 디스크 통계 정보           |
|              | 2-4. 지정 사용률 초과 항목      |
-----------------------------------------------------
|              | 3-1. 주요 파일 존재 여부 확인  |
| 3. 파일 및   | 3-2. 주요 파일 접근 권한 확인  |
|   디렉터리   | 3-3. 기본 허가권 설정 확인      |
|      관리    | 3-4. 파일 용량 확인             |
-----------------------------------------------------
|              | 4-1. 활성화 서비스 리스트       |
| 4. 서비스    | 4-2. 비활성화 서비스 리스트     |
|      관리    | 4-3. 특정 서비스 확인           |
|              | 4-4. 서비스 포트 리스트         |
-----------------------------------------------------



-----------------------------------------------------
|       1. 프로세스 관리        |    2023_02_15    |
-----------------------------------------------------

=============1-1. 동작중인 프로세스=============

UID          PID    PPID  C STIME TTY          TIME CMD
```

```
root                 1      0    0 05:46 ?        00:00:01 /usr/lib/systemd/systemd
--switched-root --system --deserialize 22
root                 2      0    0 05:46 ?        00:00:00 [kthreadd]
root                 4      2    0 05:46 ?        00:00:00 [kworker/0:0H]
root                 6      2    0 05:46 ?        00:00:00 [ksoftirqd/0]
root                 7      2    0 05:46 ?        00:00:00 [migration/0]
root                 8      2    0 05:46 ?        00:00:00 [rcu_bh]
root                 9      2    0 05:46 ?        00:00:00 [rcu_sched]
root                10      2    0 05:46 ?        00:00:00 [lru-add-drain]
root                11      2    0 05:46 ?        00:00:00 [watchdog/0]

---------------------------------------- < 생략 > ----------------------------------------

==============1-2. 평균 시스템 부하==============

 17:16:40 up 55 min,  3 users,  load average: 0.00, 0.01, 0.05

=============1-3. CPU 코어별 사용량=============

Linux 3.10.0-1160.83.1.el7.x86_64 (Linux-1)     2023년 02월 15일     _x86_64_     (  1
CPU)

                UID        PID    %usr %system  %guest   %CPU   CPU  Command
                  0          1    0.02    0.05    0.00    0.07     0
/usr/lib/systemd/systemd --switched-root --system --deserialize 22
                  0          5    0.00    0.02    0.00    0.02     0  kworker/u256:0
                  0          6    0.00    0.01    0.00    0.01     0  ksoftirqd/0
                  0          9    0.00    0.01    0.00    0.01     0  rcu_sched
                  0         11    0.00    0.00    0.00    0.00     0  watchdog/0
                  0         32    0.00    0.00    0.00    0.00     0  khugepaged
                  0        286    0.00    0.00    0.00    0.00     0  scsi_eh_1
                  0        295    0.00    0.00    0.00    0.00     0  irq/16-vmwgfx
                  0        404    0.00    0.02    0.00    0.02     0  xfsaild/dm-0
                  0        405    0.00    0.00    0.00    0.00     0  kworker/0:1H
                  0        483    0.00    0.00    0.00    0.00     0
/usr/lib/systemd/systemd-journald
                  0        512    0.00    0.00    0.00    0.00     0
/usr/lib/systemd/systemd-udevd
                  0        627    0.00    0.00    0.00    0.00     0  /sbin/auditd
```

```
                81                650          0.00          0.00          0.00          0.01             0
/usr/bin/dbus-daemon        --system        --address=systemd:        --nofork        --nopidfile
--systemd-activation
                 0                656          0.00          0.00          0.00          0.00             0
/usr/bin/VGAuthService  -s
                 0          657      0.04      0.08      0.00      0.12         0   /usr/bin/vmtoolsd
               999                658          0.00          0.00          0.00          0.00             0
/usr/lib/polkit-1/polkitd  --no-debug
                 0                659          0.00          0.00          0.00          0.00             0
/usr/lib/systemd/systemd-logind
               998          662       0.00      0.00      0.00      0.00         0   /usr/sbin/chronyd
                 0          694       0.00      0.01      0.00      0.02         0   /usr/bin/python2
-Es  /usr/sbin/firewalld  --nofork  --nopid
                 0                724          0.00          0.00          0.00          0.01             0
/usr/sbin/NetworkManager  --no-daemon
                 0         1024       0.01      0.01      0.00      0.02         0   /usr/bin/python2
-Es  /usr/sbin/tuned  -l  -P
                 0         1026       0.00      0.00      0.00      0.00         0   /usr/sbin/sshd  -D
                 0         1029       0.00      0.00      0.00      0.01         0   /usr/sbin/rsyslogd
-n
                 0              1196          0.00          0.00          0.00          0.00             0
/usr/libexec/postfix/master  -w
                 0         1338       0.00      0.00      0.00      0.00         0   /usr/sbin/smartd  -n
-q  never
                 0         1357       0.00      0.00      0.00      0.00         0   sshd:  tech1-1  [priv]
              1002              1359          0.00          0.01          0.00          0.01             0   sshd:
tech1-1@pts/1
              1002         1360       0.00      0.00      0.00      0.00         0   -bash
                 0         1379       0.00      0.00      0.00      0.00         0   sshd:  tech2-1  [priv]
              1001              1381          0.00          0.01          0.00          0.01             0   sshd:
tech2-1@pts/2
              1001         1382       0.00      0.00      0.00      0.00         0   -bash
---------------------------------------- < 생략 > ----------------------------------------

==============1-4. 사용중인 메모리==============

              total        used        free      shared  buff/cache   available
Mem:           972M        198M        265M        7.7M        508M        586M
Swap:          2.0G          0B        2.0G
```

=======1-5. 네트워크 세션 목록 및 상태========

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:ssh	0.0.0.0:*	LISTEN
tcp	0	0	localhost:smtp	0.0.0.0:*	LISTEN
tcp	0	0	Linux-1:ssh	192.168.1.1:60291	ESTABLISHED
tcp	0	0	Linux-1:ssh	192.168.1.1:59910	ESTABLISHED
tcp	0	0	Linux-1:ssh	192.168.1.1:59908	ESTABLISHED
tcp6	0	0	[::]:http	[::]:*	LISTEN
tcp6	0	0	[::]:ssh	[::]:*	LISTEN
tcp6	0	0	localhost:smtp	[::]:*	LISTEN
udp	0	0	localhost:323	0.0.0.0:*	
udp6	0	0	localhost:323	[::]:*	
raw6	0	0	[::]:ipv6-icmp	[::]:*	7

Active UNIX domain sockets (servers and established)

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ACC]	STREAM	LISTENING	17932	/var/run/vmware/guestServicePipe
unix	3	[ ]	DGRAM		8981	/run/systemd/notify
unix	2	[ ]	DGRAM		8983	/run/systemd/cgroups-agent
unix	2	[ACC]	STREAM	LISTENING	8998	/run/systemd/journal/stdout
unix	5	[ ]	DGRAM		9001	/run/systemd/journal/socket
unix	17	[ ]	DGRAM		9003	/dev/log
unix	2	[ACC]	STREAM	LISTENING	13634	/run/systemd/private
unix	2	[ ]	DGRAM		17493	/var/run/chrony/chronyd.sock
unix	2	[ACC]	SEQPACKET	LISTENING	13671	/run/udev/control
unix	2	[ACC]	STREAM	LISTENING	16746	/run/dbus/system_bus_socket

---------- < 생략 > ----------

```
-----------------------------------------------------
|          2. 디스크 관리          |    2023_02_15    |
-----------------------------------------------------

==========2-1. 디스크별 파티셔닝 항목==========

Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/sdb: 1073 MB, 1073741824 bytes, 2097152 sectors
Disk /dev/mapper/centos-root: 18.2 GB, 18249416704 bytes, 35643392 sectors
Disk /dev/mapper/centos-swap: 2147 MB, 2147483648 bytes, 4194304 sectors

==========2-2. 디스크별 사용률==========

Filesystem                Size  Used Avail Use% Mounted on
devtmpfs                  475M     0  475M   0% /dev
tmpfs                     487M     0  487M   0% /dev/shm
tmpfs                     487M  7.7M  479M   2% /run
tmpfs                     487M     0  487M   0% /sys/fs/cgroup
/dev/mapper/centos-root    17G  1.8G   16G  11% /
/dev/sda1                1014M  199M  816M  20% /boot
tmpfs                      98M     0   98M   0% /run/user/0
tmpfs                      98M     0   98M   0% /run/user/1002
tmpfs                      98M     0   98M   0% /run/user/1001

==========2-3. 디스크 통계 정보==========

smartctl 이 이미 존재 합니다.
Information for disk sda:
smartctl 7.0 2018-12-30 r4883 [x86_64-linux-3.10.0-1160.83.1.el7.x86_64] (local build)
Copyright (C) 2002-18, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Vendor:               VMware,
Product:              VMware Virtual S
Revision:             1.0
User Capacity:        21,474,836,480 bytes [21.4 GB]
Logical block size:   512 bytes
Device type:          disk
```

```
Local Time is:        Wed Feb 15 17:16:40 2023 KST
SMART support is:     Unavailable - device lacks SMART capability.

=== START OF READ SMART DATA SECTION ===
Current Drive Temperature:     0 C
Drive Trip Temperature:        0 C

Error Counter logging not supported

Device does not support Self Test logging

Information for disk sdb:
smartctl 7.0 2018-12-30 r4883 [x86_64-linux-3.10.0-1160.83.1.el7.x86_64] (local build)
Copyright (C) 2002-18, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Vendor:               VMware,
Product:              VMware Virtual S
Revision:             1.0
User Capacity:        1,073,741,824 bytes [1.07 GB]
Logical block size:   512 bytes
Device type:          disk
Local Time is:        Wed Feb 15 17:16:40 2023 KST
SMART support is:     Unavailable - device lacks SMART capability.

=== START OF READ SMART DATA SECTION ===
Current Drive Temperature:     0 C
Drive Trip Temperature:        0 C

Error Counter logging not supported

Device does not support Self Test logging

Information for disk sr0:
smartctl 7.0 2018-12-30 r4883 [x86_64-linux-3.10.0-1160.83.1.el7.x86_64] (local build)
Copyright (C) 2002-18, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Vendor:               NECVMWar
```

```
Product:                VMware IDE CDR10
Revision:               1.00
Compliance:             SPC-3
Device type:            CD/DVD
Local Time is:          Wed Feb 15 17:16:40 2023 KST
SMART support is:       Unavailable - device lacks SMART capability.

=== START OF READ SMART DATA SECTION ===
Current Drive Temperature:      0 C
Drive Trip Temperature:         0 C

Error Counter logging not supported

Device does not support Self Test logging


========2-4. 지정 사용률 초과 항목 체크========

No disk utilization exceeded the threshold.

------------------------------------------------------
|  3. 파일 및 디렉터리  관리 |   2023_02_15   |
------------------------------------------------------

==3-1. 주요 파일 존재 여부 및 접근 권한 확인==

[위험] /var/log/messages의 Permission 변경하세요!!
[위험] /etc/crontab의 Permission 변경하세요!!
[위험] /var/log/wtmp의 Permission 변경하세요!!
[위험] /var/log/lastlog의 Permission 변경하세요!!
[위험] /etc/passwd의 Permission 변경하세요!!
/etc/shadow does not exist.
[위험] /etc/pam.d의 Permission 변경하세요!!
[위험] /etc/hosts.allow의 Permission 변경하세요!!
[위험] /etc/hosts.deny의 Permission 변경하세요!!
[위험] /etc/securetty의 Permission 변경하세요!!
[위험] /etc/security의 Permission 변경하세요!!
[위험] /etc/rc.d/init.d의 Permission 변경하세요!!
[위험] /etc/services의 Permission 변경하세요!!
```

```
/etc/cron.allow does not exist.
[위험] /etc/cron.deny의 Permission 변경하세요!!
[위험] /etc/ssh의 Permission 변경하세요!!
[위험] /etc/sysctl.conf의 Permission 변경하세요!!
===========3-2. 기본 허가권 설정 확인===========

[안전] UMASK OK.

==============3-3. 파일 용량 확인==============

-------------------100MB 이상 파일--------------------
/proc/kcore
/sys/devices/pci0000:00/0000:00:0f.0/resource1_wc
/sys/devices/pci0000:00/0000:00:0f.0/resource1
/var/cache/yum/x86_64/7/updates/gen/primary_db.sqlite
/usr/lib/locale/locale-archive
------------------------------------------------------


-------------------------------------------------------
|         4. 서비스 관리         |    2023_02_15    |
-------------------------------------------------------

======4-1. 현재 활성화 되어있는 항목 출력======

auditd.service
autovt@.service
chronyd.service
crond.service
dbus-org.fedoraproject.FirewallD1.service
dbus-org.freedesktop.nm-dispatcher.service
firewalld.service
getty@.service
irqbalance.service
kdump.service
lvm2-monitor.service
microcode.service
NetworkManager-dispatcher.service
NetworkManager-wait-online.service
NetworkManager.service
```

postfix.service
rhel-autorelabel-mark.service
rhel-autorelabel.service
rhel-configure.service
rhel-dmesg.service
rhel-domainname.service
rhel-import-state.service
rhel-loadmodules.service
rhel-readonly.service
rsyslog.service
smartd.service
sshd.service
sysstat.service
systemd-readahead-collect.service
systemd-readahead-drop.service
systemd-readahead-replay.service
tuned.service
vgauthd.service
vmtoolsd.service

=======4-2. 현재 비활성화 되어있는 항목=======

arp-ethers.service
blk-availability.service
chrony-wait.service
console-getty.service
console-shell.service
cpupower.service
debug-shell.service
dhcpd.service
dhcpd6.service
dhcrelay.service
ebtables.service
httpd.service
iprdump.service
iprinit.service
iprupdate.service
plymouth-halt.service
plymouth-kexec.service

plymouth-poweroff.service
plymouth-quit-wait.service
plymouth-quit.service
plymouth-read-write.service
plymouth-reboot.service
plymouth-start.service
rdisc.service
rsyncd.service
saslauthd.service
serial-getty@.service
systemd-bootchart.service
systemd-nspawn@.service
vsftpd.service
vsftpd@.service
wpa_supplicant.service

==========4-3. 필수 서비스 항목==========

==========4-4. 서비스 포트 항목==========

*:22 sshd
*:80 httpd
127.0.0.1:25 master
[::1]:25 master

. 보안 점검 쉘 스크립트

1) 보안 점검 스크립트 체크 리스트

분류	세부항목
계정 관리	SSH 루트접근 해제
	패스워드 복잡성 알림
	계정잠금 임계값
	패스워드 최대 잠금 기한 설정
	패스워드 파일 보호
파일 관리	$PATH의 값에 취약점 검사

2) 쉘 스크립트 코드

```
#!/bin/bash

today=`date "+%Y_%m_%d"`
user=$(whoami)

function TEXT() {
mkdir /script 2> /dev/null
if [ -f /script/${user}_${today}_security.txt ]; then
    rm -rf /script/${user}_${today}_security.txt
    touch /script/${user}_${today}_security.txt
else
    touch /script/${user}_${today}_security.txt
fi
}

function SYS_CK() {




echo ""
echo "-------------------------------------------------"
echo "|  보안 점검 보고서                  |     ${today}  |"
echo "-------------------------------------------------"
echo "|    4조_ 김승훈 오주현 이동민 최낙원 최원석          |"
echo "-------------------------------------------------"
```

```
echo "| 1. SSH 루트 접근 해제                                    |"
echo "-------------------------------------------------"
echo "| 2. 패스워드 복잡성 4개 검사                              |"
echo "-------------------------------------------------"
echo "| 3. 계정잠금 임계 값 알림                                 |"
echo "-------------------------------------------------"
echo "| 4. 패스워드 최대 잠금 기한                               |"
echo "-------------------------------------------------"
echo "| 5. 패스워드 취약점 검사                                  |"
echo "-------------------------------------------------"
echo "| 6. ₩$PATH 값 취약점 검사                                   |"
echo "-------------------------------------------------"
echo ""
echo ""

#target파일에서 test(검사요소)가 존재하지 않는다면 answer을 반환함
single_filecheck() {
        if [ -z "$test" ]
        then
        echo "취약요소 발견,
$answer"
        else
        echo "해당 취약요소 발견안됨"
        fi
}
#target파일에서checklist들이  test(검사요소)에 부합하지 않는다면 answer을 반환함
multi_filecheck() {
        for check in ${checklist[@]}
        do
                test=$(cat $target | grep ${check})
                if [ -z "$test" ]
                then
                        echo "취약요소발견, $check" "$answer"
                else
                        echo "해당 취약요소 발견안됨"
                fi
        done
}
```

```
echo "-------------------------------------------------"
echo "|  1. SSH 루트 접근 해제            |    ${today}  |"
echo "-------------------------------------------------"
echo ""


#1번째 설정(SSH에 대한 루트로그인)
target=/etc/ssh/ssh_config
test=$(cat $target | grep "PermitRootLogin no")
answer=" 루트로그인을 비활성화하세요!"

single_filecheck


echo ""
echo "-------------------------------------------------"
echo "|  2. 패스워드 복잡성 4개 검사      |    ${today}   |"
echo "-------------------------------------------------"
echo ""



#2번째 설정(비밀번호 취약점 설정)
target=/etc/security/pwquality.conf
checklist=(dcredit=-1 ucredit=-1 lcredit=-1 ocredit=-1)
answer="설정 안됨, 비밀번호설정 취약"

multi_filecheck


echo ""
echo "-------------------------------------------------"
echo "|  3. 계정잠금 임계 값 알림         |    ${today}   |"
echo "-------------------------------------------------"
echo ""


#3번째 설정(계정 잠금 임계값설정)
```

```
target=/etc/pam.d/system-auth
test=`cat "$target" | grep "auth required pam_tally2.so deny=5
no_magic_root"`
answer="계정 잠금 임계값이 설정되어있지 않습니다!"

single_filecheck



echo ""
echo "-------------------------------------------------"
echo "|  4. 패스워드 최대 잠금 기한      |   ${today}   |"
echo "-------------------------------------------------"
echo ""




# 4번째 설정(비밀번호 유효기간 설정)
file="/etc/login.defs"
max_days="99999"
status=0

if [ -f "/etc/login.defs" ]; then
  # 비밀번호 유효 기간이 90 인지 확인
  max_days=$(grep "^PASS_MAX_DAYS" /etc/login.defs | awk '{print $2}')
  if [ $max_days != "90" ]; then
    sed -i 's/^PASS_MAX_DAYS.*/PASS_MAX_DAYS 90/' /etc/login.defs
    echo "비밀번호 유효기간을 90으로 설정 합니다."
  else
    echo "이미 유효기간이 90으로 설정되어 있습니다."
  fi
else
  echo "Error: /etc/login.defs does not exist."
  exit 1
fi

echo ""
echo "-------------------------------------------------"
echo "|  5. 패스워드 취약점 검사          |   ${today}   |"
echo "-------------------------------------------------"
```

```
echo ""

# 5번째 설정(비밀번호 보관 파일 설정)
file="/etc/shadow"

if [ -f $file ]; then
   echo "$file 이 존재합니다! 코드를 다시 입력해주세요!"
   status=1
else
   echo "$file 이 존재하지 않습니다. "
fi

echo ""
echo "-------------------------------------------------"
echo "|  6. ₩$PATH 값 취약점 검사          |    ${today}    |"
echo "-------------------------------------------------"
echo ""


# 6번째 설정(PATH 경로 설정)
if [[ $PATH == .:* ]]; then
   echo "PATH 값은 '.:' 로 시작되면 안됩니다."
   status=1
else
   echo "정상적인 PATH값 루트입니다"
fi

}
TEXT
SYS_CK > /script/${user}_${today}_security.txt
```

3) 실행 결과

 # /script/[사용자]_[날짜]_security.txt

```
------------------------------------------------------------
|  보안 점검 보고서                |     2023_02_15  |
------------------------------------------------------------
|    4조_ 김승훈 오주현 이동민 최낙원 최원석              |
------------------------------------------------------------
| 1. SSH 루트 접근 해제                                    |
------------------------------------------------------------
| 2. 패스워드 복잡성 4개 검사                              |
------------------------------------------------------------
| 3. 계정잠금 임계 값 알림                                 |
------------------------------------------------------------
| 4. 패스워드 최대 잠금 기한                               |
------------------------------------------------------------
| 5. 패스워드 취약점 검사                                  |
------------------------------------------------------------
| 6. $PATH 값 취약점 검사                                   |
------------------------------------------------------------



------------------------------------------------------------
|  1. SSH 루트 접근 해제           |     2023_02_15  |
------------------------------------------------------------

해당 취약요소 발견안됨


------------------------------------------------------------
|  2. 패스워드 복잡성 4개 검사      |    2023_02_15   |
------------------------------------------------------------

해당 취약요소 발견안됨
해당 취약요소 발견안됨
해당 취약요소 발견안됨
해당 취약요소 발견안됨
```

```
------------------------------------------------------------
|  3. 계정잠금 임계 값 알림          |   2023_02_15   |
------------------------------------------------------------

해당 취약요소 발견안됨

------------------------------------------------------------
|  4. 패스워드 최대 잠금 기한        |   2023_02_15   |
------------------------------------------------------------

이미 유효기간이 90으로 설정되어 있습니다.

------------------------------------------------------------
|  5. 패스워드 취약점 검사           |   2023_02_15   |
------------------------------------------------------------

/etc/shadow 이 존재하지 않습니다.

------------------------------------------------------------
|  6. $PATH 값 취약점 검사           |   2023_02_15   |
------------------------------------------------------------

정상적인 PATH값 루트입니다.
```

4. 패키지 설치 쉘 스크립트

1) 패키지 설치 항목 리스트

분류	세부항목	비고
일반 패키지	net-tools	각종 네트워크 명령어들을 관리하는 패키지
	wget	웹 서버로부터 콘텐츠를 가져오는 패키지 / HTTPS, FTP 프로토콜 지원
	bind-utils	Name server lookup 패키지
	vsftpd	파일 전송 프로토콜
	dhcp	IP 동적 할당
	httpd	문서 전송 프로토콜
서버 모니터링 패키지	iftop	네트워크 인터페이스의 트래픽 모니터링 패키지
	sysstat	리눅스 성능 측정 도구 패키지
	lsof	시스템의 열려있는 파일에 대한 정보를 출력
	traceroute	라우팅 확인
	gdb	디버깅 툴
권장 패키지	whois	도메인 소유자나 IP 주소 위치 등 이외 다른 정보를 확인할 때 사용
	epel	yum의 확장된 최신 저장소 (엔터프라이즈 리눅스 추가 패키지)
패키지 업데이트	yum update	연결가능한 리포지토리 연결 업데이트 할 수 있는 패키지 목록 출력 및 실행

2) 쉘 스크립트 코드

```
#!/bin/bash

today=`date "+%Y_%m_%d"`
user=$(whoami)



function TEXT() {
if [ -f /script/${user}_${today}_system.txt ]; then
    rm -rf /script/${user}_${today}_system.txt
    touch /script/${user}_${today}_system.txt
else
    touch /script/${user}_${today}_system.txt
fi
}
```

```
function SYS_CK() {function TEXT() {
if [ -f /script/${user}_${today}_system.txt ]; then
    rm -rf /script/${user}_${today}_system.txt
    touch /script/${user}_${today}_system.txt
else
    touch /script/${user}_${today}_system.txt
fi
}

function SYS_CK() {


# Package Install List
# epel=$(yum -y install epel*)
# net-tools=$(yum -y install net-tools)
# wget=$(yum -y install wget)
# bind-utils=$(yum -y install bind-utils)
# vsftpd=$(yum -y install vsftpd-*)
# dhcp=$(yum -y install dhcp-*)
# httpd=$(yum -y install httpd-*)
# iftop=$(yum -y install iftop)
# sysstat=$(yum -y install sysstat)
# lsof=$(yum -y install lsof)
# traceroute=$(yum -y install traceroute)
# gdb=$(yum -y install gdb)
# whois=$(yum -y install whois)

# update=$(yum -y update)


echo ""
echo "-----------------------------------------------"
echo "|     패키지 설치 보고서  |    ${today}          |"
echo "-----------------------------------------------"
echo "|    4조_ 김승훈 오주현 이동민 최낙원 최원석      |"
echo "-----------------------------------------------"
echo "|                | 1-1. net-tools                    |"
echo "|                | 1-2. wget                           |"
```

```
echo "| 1. 일반 패키지 | 1-3. bind-utils                |"
echo "|                | 1-4. vsftpd                    |"
echo "|                | 1-5. dhcp                      |"
echo "|                | 1-6. httpd                     |"
echo "-------------------------------------------------"
echo "|                | 2-1. iftop                     |"
echo "|                | 2-2. sysstat                   |"
echo "| 2. 서버 모니터링| 2-3. lsof                      |"
echo "|      패키지     | 2-4. traceroute                |"
echo "|                | 2-5. gdb                       |"
echo "-------------------------------------------------"
echo "| 3. 권장 패키지 | 3-1. whois                     |"
echo "|                | 3-2. epel                      |"
echo "-------------------------------------------------"
echo "| 4. 패키지    | 4-1. yum update                  |"
echo "|    업데이트                                     |"
echo "-------------------------------------------------"
echo ""
echo ""



echo "-------------------------------------------------"
echo "|        1. 일반 패키지        |    ${today}       |"
echo "-------------------------------------------------"
echo ""

echo "================== 1-1. net-tools ================"
echo ""

if rpm -q net-tools > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install net-tools
  echo "================ 패키지 설치 완료==============="
fi

echo ""
echo "================== 1-2. wget ====================="
```

```
echo ""

if rpm -q wget > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install wget
  echo "============== 패키지 설치 완료==============="
fi

echo ""
echo "================ 1-3. bind-utils ==============="
echo ""

if rpm -q bind-utils > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install bind-utils
  echo "============== 패키지 설치 완료==============="
fi

echo ""
echo "================ 1-4. vsftpd ==================="
echo ""

if rpm -q vsftpd > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install vsftpd-*
  echo "============== 패키지 설치 완료==============="
fi

echo ""
echo "================ 1-5. dhcp ====================="
echo ""

if rpm -q dhcp > /dev/null; then
```

```
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install dhcp-*
  echo "============== 패키지 설치 완료==============="
fi

echo ""
echo "================ 1-6. httpd ====================="
echo ""

if rpm -q httpd > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install httpd-*
  echo "============== 패키지 설치 완료==============="
fi



echo ""
echo "-----------------------------------------------"
echo "|    2. 서버 모니터링 패키지    |    ${today}    |"
echo "-----------------------------------------------"
echo ""

echo "================= 2-1. iftop ==================="
echo ""

if rpm -q iftop > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install iftop
  echo "============== 패키지 설치 완료==============="
fi

echo ""
echo "================= 2-2. sysstat ================="
```

```
echo ""


if rpm -q sysstat > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install sysstat
  echo "================ 패키지 설치 완료=============="
fi

echo ""
echo "================== 2-3. lsof ====================="
echo ""

if rpm -q lsof > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install lsof
  echo "================ 패키지 설치 완료=============="
fi

echo ""
echo "================== 2-4. traceroute =============="
echo ""

if rpm -q traceroute > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install traceroute
  echo "================ 패키지 설치 완료=============="
fi

echo ""
echo "================== 2-5. gdb ======================"
echo ""
```

```
if rpm -q gdb > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install gdb
  echo "================ 패키지 설치 완료==============="
fi

echo ""
echo "-----------------------------------------------"
echo "|    3. 권장 패키지    |    ${today}               |"
echo "-----------------------------------------------"
echo ""

echo "================== 3-1. whois ====================="
echo ""



if rpm -q whois > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install whois
  echo "================ 패키지 설치 완료==============="
fi

echo ""
echo "=================== 3-2. epel ====================="
echo ""

if rpm -q epel-release > /dev/null; then
  echo "패키지가 이미 존재 합니다."
else
  echo "패키지가 존재하지 않습니다. 설치작업을 실행합니다."
  yum -y install epel*
  echo "================ 패키지 설치 완료==============="
fi

echo ""
```

```
echo "------------------------------------------------"
echo "|         4. 패키지 업데이트     |     ${today}        |"
echo "------------------------------------------------"
echo ""

echo "================== 4-1. yum update ==============="
echo ""

yum -y update
echo "패키지가 업데이트 되었습니다."

}
TEXT
SYS_CK > /script/${user}_${today}_package.txt
```

3) 실행 결과

 # /script/[사용자]_[날짜]_package.txt

```
-----------------------------------------------------
|    패키지 설치 보고서  |   2023_02_15          |
-----------------------------------------------------
|   4조_ 김승훈 오주현 이동민 최낙원 최원석     |
-----------------------------------------------------
|                  | 1-1. net-tools                |
|                  | 1-2. wget                      |
| 1. 일반 패키지  | 1-3. bind-utils                |
|                  | 1-4. vsftpd                    |
|                  | 1-5. dhcp                      |
|                  | 1-6. httpd                    |
-----------------------------------------------------
|                  | 2-1. iftop                     |
|                  | 2-2. sysstat                  |
| 2. 서버 모니터링 | 2-3. lsof                      |
|       패키지      | 2-4. traceroute               |
|                  | 2-5. gdb                      |
-----------------------------------------------------
| 3. 권장 패키지  | 3-1. whois                     |
|                  | 3-2. epel                      |
-----------------------------------------------------
| 4. 패키지        | 4-1. yum update               |
|   업데이트                                        |
-----------------------------------------------------


-----------------------------------------------------
|       1. 일반 패키지       |   2023_02_15       |
-----------------------------------------------------

================== 1-1. net-tools ===============

패키지가 이미 존재 합니다.

================== 1-2. wget ====================

패키지가 이미 존재 합니다.
```

```
================ 1-3. bind-utils ===============

패키지가 이미 존재 합니다.

================ 1-4. vsftpd ===================

패키지가 이미 존재 합니다.

================ 1-5. dhcp ====================

패키지가 이미 존재 합니다.

================ 1-6. httpd ====================

패키지가 이미 존재 합니다.

-------------------------------------------------------
|    2. 서버 모니터링 패키지    |    2023_02_15    |
-------------------------------------------------------

================ 2-1. iftop ====================

패키지가 이미 존재 합니다.

================ 2-2. sysstat ==================

패키지가 이미 존재 합니다.

================ 2-3. lsof ====================

패키지가 이미 존재 합니다.

================ 2-4. traceroute ==============

패키지가 이미 존재 합니다.

================ 2-5. gdb ====================
```

```
패키지가 이미 존재 합니다.

------------------------------------------------------
|   3. 권장 패키지   |   2023_02_15            |
------------------------------------------------------

================== 3-1. whois ====================

패키지가 이미 존재 합니다.

================== 3-2. epel ====================

패키지가 이미 존재 합니다.

------------------------------------------------------
|       4. 패키지 업데이트   |   2023_02_15      |
------------------------------------------------------

================== 4-1. yum update ==============

Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.kakao.com
 * epel: hkg.mirror.rackspace.com
 * extras: mirror.kakao.com
 * updates: mirror.kakao.com
No packages marked for update
패키지가 업데이트 되었습니다.
```

5. 백업 쉘 스크립트

1) 로그 파일 백업 스크립트 코드

```
#!/bin/bash
## 변수설정
HOST="$(/usr/bin/hostname)"
LOG="/tmp/backup.log"
DATE="$(/usr/bin/date +%Y.%m.%d)"
#백업할 디렉터리 /파일을 지정
BAK_LIST="/var/log"
#백업 디렉토리
BAK_PATH="/mnt/BACKUP/${HOST}"
#백업 파일명
BAK_FILE="${BAK_PATH}/${DATE}_${HOST}.tgz"



## 로그파일 생성
touch "${LOG}"

##백업 디렉터리 확인
if [ -e "${BAK_PATH}" ];
then
    #백업디렉터리가 존재한다면
    echo ""
else
    mkdir -p "${BAK_PATH}"
fi

## 로그 기록 시작
{
    # 백업 시작 시간
    echo ""
    echo "=== 백업시작 시간 :"
    date
    echo ""
```

```
    ## 백업
    # p : 퍼미션 유지 P : 절대경로
    echo "=== 백업 진행 파일 :"
    tar cvzpPf "${BAK_FILE}" "${BAK_LIST}"
    echo ""

    # 백업파일 정보

    NAME="$(ls -la "${BAK_FILE}" | awk '{print $9}')"
    SIZE="$(ls -la "${BAK_FILE}" | awk '{print $5}')"
    echo "=== 백업 파일 정보 :"
    echo " | 파일명 : ${NAME}"
    echo " | 파일 크기 : ${SIZE}Byte"

    # 백업 종료 시간
    echo ""
    echo "=== 백업 종료 시간 :"
    date
    echo ""


}>>"${LOG}"
## 로그기록 끝



## 백업 로그 저장됨
```

2) DB 파일 백업 스크립트 코드

```
## 먼저 Mariadb 설치를 진행

#!/bin/bash

LOG="/tmp/backup.log"
today=`date "+%Y_%m_%d"`
BAK_FILE="/mnt/BACKUP/${HOST}/DB/user_backup_${today}.sql"
BAK_PATH="/mnt/BACKUP/${HOST}/DB"

## 로그파일 생성
```

```
touch "${LOG}"

##백업 디렉터리 확인
if [ -e "${BAK_PATH}" ];
then
    #백업디렉터리가 존재한다면
    echo ""
else
    mkdir -p "${BAK_PATH}"
fi

# user database 백업
mysqldump -u root -p --databases test > ${BAK_FILE}

## 로그 기록 시작
{
    # 백업 시작 시간
    echo ""
    echo "=== 백업시작 시간 :"
    date
    echo ""

    # 백업파일 정보

    NAME="$(ls -la "${BAK_FILE}" | awk '{print $9}')"
    SIZE="$(ls -la "${BAK_FILE}" | awk '{print $5}')"
    echo "=== 백업 파일 정보 :"
    echo " | 파일명 : ${NAME}"
    echo " | 파일 크기 : ${SIZE}Byte"

    # 백업 종료 시간
    echo ""
    echo "=== 백업 종료 시간 :"
    date
    echo ""


}>>|"${LOG}"
## 로그기록 끝
```

3) 실행 결과

```
#ls –l /mnt/BAKCUP/root
-rw-r--r-- 1 root root     383  2월 12 22:30 2023.02.12_root.tgz
-rw-r--r-- 1 root root 400827  2월 16 01:11 2023.02.16._root.tgz

#ls –l /mnt/BAKCUP/DB
-rw-r--r-- 1 root root 2068  2월 16 00:20 user_backup_2023_02_16.sql


## 백업 작업 기록을 따로 파일에 기록
#cat /tmp/backup.log
=== 백업시작 시간 :
2023. 02. 16. (목) 00:20:15 KST

=== 백업 파일 정보 :
 | 파일명 : /mnt/BACKUP//DB/user_backup_2023_02_16.sql
 | 파일 크기 : 2068Byte

=== 백업 종료 시간 :
2023. 02. 16. (목) 00:20:15 KST

=== 백업시작 시간 :
2023. 02. 16. (목) 01:18:00 KST

=== 백업 진행 파일 :
/var/log/
/var/log/tallylog
/var/log/grubby_prune_debug
/var/log/lastlog
/var/log/wtmp
/var/log/btmp
/var/log/tuned/
/var/log/tuned/tuned.log
/var/log/audit/
/var/log/audit/audit.log
/var/log/chrony/
/var/log/anaconda/
```

/var/log/anaconda/anaconda.log
/var/log/anaconda/syslog
/var/log/anaconda/X.log
/var/log/anaconda/program.log
/var/log/anaconda/packaging.log
/var/log/anaconda/storage.log
/var/log/anaconda/ifcfg.log
/var/log/anaconda/ks-script-WcmxKP.log
/var/log/anaconda/ks-script-XcuvwP.log
/var/log/anaconda/journal.log
/var/log/rhsm/
/var/log/boot.log
/var/log/vmware-vgauthsvc.log.0
/var/log/vmware-vmsvc.log
/var/log/firewalld
/var/log/yum.log
/var/log/vmware-vmtoolsd-root.log
/var/log/vmware-vmsvc-root.log
/var/log/grubby
/var/log/dmesg.old
/var/log/nginx/
/var/log/nginx/access.log
/var/log/nginx/error.log
/var/log/boot.log-20230212
/var/log/cron-20230212
/var/log/cron
/var/log/maillog-20230212
/var/log/maillog
/var/log/messages-20230212
/var/log/messages
/var/log/secure-20230212
/var/log/secure
/var/log/spooler-20230212
/var/log/spooler
/var/log/sa/
/var/log/sa/sa12
/var/log/sa/sar12
/var/log/sa/sa13
/var/log/sa/sa15

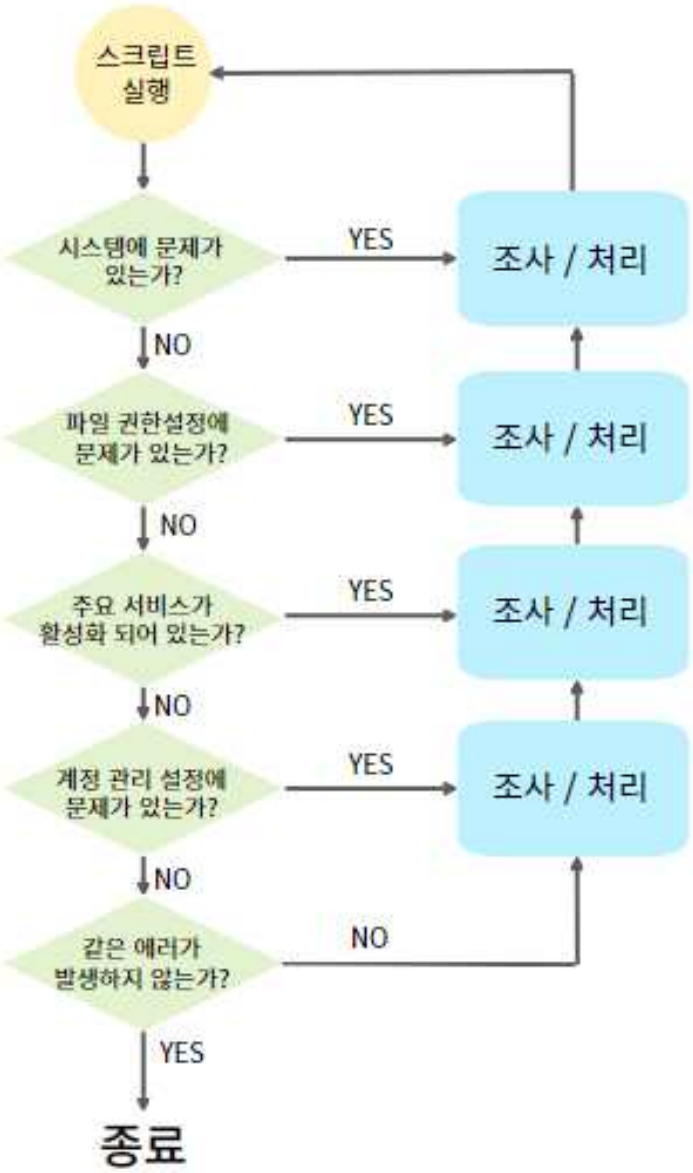
```
/var/log/sa/sar15
/var/log/sa/sa16
/var/log/dmesg
/var/log/vmware-network.6.log
/var/log/vmware-network.5.log
/var/log/vmware-network.4.log
/var/log/vmware-network.3.log
/var/log/vmware-network.2.log
/var/log/vmware-network.1.log
/var/log/vmware-network.log

=== 백업 파일 정보 :
 | 파일명 : /mnt/BACKUP/root/2023.02.16_root.tgz
 | 파일 크기 : 400845Byte

=== 백업 종료 시간 :
2023. 02. 16. (목) 01:18:00 KST
```

5. 트러블 슈팅 쉘 스크립트

1) 트러블 슈팅 쉘 스크립트 동작 흐름도



2) 쉘 스크립트 코드

```
today=`date "+%Y_%m_%d"`
user=$(whoami)

function TEXT() {
if [ -f /script/${user}_${today}_troubleshooting.txt ]; then
    rm -rf /script/${user}_${today}_troubleshooting.txt
    touch /script/${user}_${today}_troubleshooting.txt
else
    touch /script/${user}_${today}_troubleshooting.txt
fi
}

function SYS_CK() {

echo ""
echo "-----------------------------------------------"
echo "|        트러블 슈팅 보고서        |    ${today}   |"
echo "-----------------------------------------------"
echo "|    4조_ 김승훈 오주현 이동민 최낙원 최원석     |"
echo "-----------------------------------------------"
echo "|                                               |"
echo "| 1. 시스템 재점검 - 간소화                     |"
echo "|                                               |"
echo "-----------------------------------------------"
echo "|                                               |"
echo "| 2. 주요 서비스 활성화                         |"
echo "|                                               |"
echo "-----------------------------------------------"
echo "|                                               |"
echo "| 3. 주요 객체 접근 권한 복구                   |"
echo "|                                               |"
echo "-----------------------------------------------"
echo "|                                               |"
echo "| 4. 계정 관리 최적                              |"
echo "|                                               |"
echo "-----------------------------------------------"
echo ""
echo ""
```

```
echo "-----------------------------------------------"
echo "|    1. 시스템 재점검 - 간소화    |  ${today}  |"
echo "-----------------------------------------------"
echo ""


# 디스크 사용률
function check_disk_usage() {
  disk_usage=$(df -h | awk '{print $5}' | grep -v Use | sort -n | tail -1 | cut -d'%' -f1)

  if [ $disk_usage -gt 90 ]; then
    echo "[위험] 현재 디스크 사용량 90% 초과: $disk_usage%"
  else
    echo "디스크 사용량 정상: $disk_usage%"
  fi
}

# 메모리 사용률
function check_memory_usage() {
  memory_usage=$(free | awk 'NR==2{printf "%.2f₩n", $3*100/$2 }')

  if [ $(echo "$memory_usage > 80" | bc) -eq 1 ]; then
    echo "[위험] 현재 메모리 사용량 80% 초과: $memory_usage%"
  else
    echo "메모리 사용량 정상: $memory_usage%"
  fi
}

# 네트워크 연결 확인
function check_network_connectivity() {
  ping_result=$(ping -c 4 www.google.com)

  if [ $? -eq 0 ]; then
    echo "네트워크 정상 연결"
  else
    echo "[위험] 네트워크 연결 오류"
  fi
}
```

```
echo ""
check_disk_usage
echo ""
check_memory_usage
echo ""
check_network_connectivity
echo ""

echo ""
echo "-----------------------------------------------"
echo "|        2. 주요 서비스 활성화       |   ${today}   |"
echo "-----------------------------------------------"
echo ""

#주요 서비스 활성화

echo "Checking Apache service."
systemctl status httpd.service
if [ $? -ne 0 ]; then
  echo "웹서버 서비스를 활성화 하였습니다. "
  systemctl restart httpd.service
  if [ $? -eq 0 ]; then
    echo "현재 웹서버 서비스가 활성화 상태입니다."
  else
    echo "현재 웹서버 패키지 미설치, 설치 후 활성화 하겠습니다."
    yum -y install httpd-*
    systemctl enable httpd
    systemctl start httpd
    firewall-cmd --permanent --add-port=80/tcp
    firewall-cmd --permanent --add-service=http
  fi
fi

echo ""
echo "-----------------------------------------------"
echo "|    3. 주요 객체 접근 권한 복구    |   ${today} |"
echo "-----------------------------------------------"
```

```
echo ""


#주요 디렉터리 및 파일 접근 권한 정상화

echo ""

function file_chk() {
        sort=`ls -ld "$FILE" 2> /dev/null | awk '{print $1}'| cut -c 1` > /dev/null
        per=`ls -ld "$FILE" 2> /dev/null | awk '{print $1}'` > /dev/null
        if [ "$sort" == "d" ]; then
                        if [ "$per" != "$default" ]; then
                                echo -e  "[위험] $FILE의 Permission 변경하였습니다!!"
                        else
                                echo "$FILE 권한이 정상적입니다."
                        fi

        elif [ "$sort" == "-" ]; then
                        if [ "$per" != "$default" ]; then
                                echo -e  "[위험] $FILE의 Permission 변경하였습니다!!"
                        else
                                echo "$FILE 권한이 정상적입니다."
                        fi
        fi
}

echo ""
FILE=/ts.test
default=drwxr-xrwx
file_chk
chmod 757 /ts.test

FILE=/ts.test_1
default=drw-r--r--
file_chk
chmod 644 /ts.test_1

FILE=/ts.test/test.txt
default=-rw-------
```

```
file_chk
chmod 600 /ts.test/test.txt

FILE=/ts.test_1/test.txt
default=-rw-------
file_chk
chmod 600 /ts.test_1/test.txt

echo ""
echo "------------------------------------------------"
echo "|        4. 계정 관리 최적화        |   ${today}   |"
echo "------------------------------------------------"
echo ""

#1번째

echo "첫 번째 트러블슈팅. ssh를 통한 루트접근을 막음"
echo ""
rm -rf /etc/ssh/ssh_config
echo "          ForwardX11Trusted yes" >> /etc/ssh/ssh_config
echo "                  SendEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE
LC_MONETARY LC_MESSAGES" >> /etc/ssh/ssh_config
echo "SendEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT" >>
/etc/ssh/ssh_config
echo "          SendEnv LC_IDENTIFICATION LC_ALL LANGUAGE" >> /etc/ssh/ssh_config
echo "          SendEnv XMODIFIERS" >> /etc/ssh/ssh_config
echo "PermitRootLogin no" >> /etc/ssh/ssh_config


#2번째
echo "두 번째 트러블슈팅. 비밀번호 설정을 취약하지 않게 설정"
echo ""
rm -rf /etc/security/pwquality.conf
echo "dcredit=-1" >> /etc/security/pwquality.conf
echo "ucredit=-1" >> /etc/security/pwquality.conf
echo "lcredit=-1" >> /etc/security/pwquality.conf
echo "ocredit=-1" >> /etc/security/pwquality.conf
```

```
#3번째
echo "세 번째 트러블슈팅. 사용자 로그인시도를 5번 실패했을 경우 로그인이 잠기게 설정"
echo ""
echo "#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth            required        pam_env.so
auth            required        pam_tally2.so deny=5 no_magic_root
auth            sufficient      pam_unix.so nullok try_first_pass
auth            requisite       pam_succeed_if.so uid >= 1000 quiet_success
auth            required        pam_deny.so

account         required        pam_unix.so
account         sufficient      pam_localuser.so
account         sufficient      pam_succeed_if.so uid < 1000 quiet
account         required        pam_permit.so

password        requisite       pam_pwquality.so try_first_pass local_users_only retry=3
authtok_type=
password        sufficient      pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password        required        pam_deny.so

session         optional        pam_keyinit.so revoke
session         required        pam_limits.so
-session        optional        pam_systemd.so
session         [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session         required        pam_unix.so
" >> /etc/pam.d/system-auth

#4번째
echo "네 번째 트러블슈팅. 비밀번호 유효기간을 90일로 설정"
echo ""
if [ -f "/etc/login.defs" ]; then
  # 비밀번호 유효 기간이 90 인지 확인
  max_days=$(grep "^PASS_MAX_DAYS" /etc/login.defs | awk '{print $2}')
  if [ $max_days != "90" ]; then
    sed -i 's/^PASS_MAX_DAYS.*/PASS_MAX_DAYS 90/' /etc/login.defs
```

```
    fi
else
    echo "Error: /etc/login.defs does not exist."
    exit 1
fi

#5번째
echo "다섯 번째 트러블슈팅. 비밀번호 관련 보안파일 /etc/shadow를 숨기는 설정"
echo ""
if [ -f "/etc/shadow" ]; then
    # Execute the "pwunconv" command
    pwunconv
    exit 1
fi

}
TEXT
SYS_CK > /script/${user}_${today}_troubleshooting.txt
```

3) 실행 결과

 # /script/[사용자]_[날짜]_troubleshooting.txt

```
-------------------------------------------------------
|       트러블 슈팅 보고서       |    2023_02_15   |
-------------------------------------------------------
|   4조_ 김승훈 오주현 이동민 최낙원 최원석     |
-------------------------------------------------------
|                                                     |
| 1. 시스템 재점검 - 간소화                          |
|                                                     |
-------------------------------------------------------
|                                                     |
| 2. 주요 서비스 활성화                              |
|                                                     |
-------------------------------------------------------
|                                                     |
| 3. 주요 객체 접근 권한 복구                        |
|                                                     |
-------------------------------------------------------
|                                                     |
| 4. 계정 관리 최적                                  |
|                                                     |
-------------------------------------------------------


-------------------------------------------------------
|    1. 시스템 재점검 - 간소화    |  2023_02_15   |
-------------------------------------------------------

디스크 사용량 정상: 20%

메모리 사용량 정상: 20.44%

네트워크 정상 연결
```

```
-------------------------------------------------------
|        2. 주요 서비스 활성화        |  2023_02_15  |
-------------------------------------------------------

Checking Apache service.
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since 수 2023-02-15 16:36:24 KST; 40min ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 2089 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─2089 /usr/sbin/httpd -DFOREGROUND
           ├─2090 /usr/sbin/httpd -DFOREGROUND
           ├─2092 /usr/sbin/httpd -DFOREGROUND
           ├─2093 /usr/sbin/httpd -DFOREGROUND
           ├─2097 /usr/sbin/httpd -DFOREGROUND
           └─2101 /usr/sbin/httpd -DFOREGROUND

 2월 15 16:36:23 Linux-1 systemd[1]: Starting The Apache HTTP Server...
 2월 15 16:36:23 Linux-1 httpd[2089]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::f506:b910:a67e:eebd. Set the 'ServerName' directive globally to suppress this message
 2월 15 16:36:24 Linux-1 systemd[1]: Started The Apache HTTP Server.

-------------------------------------------------------
|    3. 주요 객체 접근 권한 복구    |  2023_02_15 |
-------------------------------------------------------

/ts.test 권한이 정상적입니다.
/ts.test_1 권한이 정상적입니다.
/ts.test/test.txt 권한이 정상적입니다.
/ts.test_1/test.txt 권한이 정상적입니다.
```

| 4. 계정 관리 최적화 | 2023_02_15 |

첫 번째 트러블슈팅. ssh를 통한 루트접근을 막음

두 번째 트러블슈팅. 비밀번호 설정을 취약하지 않게 설정

세 번째 트러블슈팅. 사용자 로그인시도를 5번 실패했을 경우 로그인이 잠기게 설정

네 번째 트러블슈팅. 비밀번호 유효기간을 90일로 설정

다섯 번째 트러블슈팅. 비밀번호 관련 보안파일 /etc/shadow를 숨기는 설정