如某项主机安全要求涉及到系统配置和服务状态的更改，则需要依次执行如下操作

1□查看和记录相关服务的初始状态

2□备份相关的系统配置

3□更改系统配置和服务状态

4）生成恢复系统配置和服务状态的命令或shell脚本。

操作之前先备份系统的如下文件：

/etc/login.defs
/etc/passwd
/etc/shadow
/etc/pam.d/system-auth
/etc/ssh/sshd_config

1.1 身份鉴别

1.1.1 a) 是否对登录操作系统和数据库系统的用户进行身份标识和鉴别 登录操作系统和数据库系统，均需要通过用户名和密码进行验证

1.1.2 b) 操作系统和数据库系统管理用户身份标识是否具有不易被冒用的特点，口令是否有复杂度要求并定期更换

1. 口令复杂度

口令必须具备采用3种以上字符、长度不少于8位并定期更换；

#vi /etc/pam.d/system_auth
password requisite pam_cracklib.so minlen=8 ucredit=1 lcredit=1 dcredit=1 ocredit=1

意思为最少有1个大写字母，1个小写字符，1个数字，1个符号

2. 口令有效期
vi /etc/login.defs PASS_MAX_DAYS 60

1.1.3 c) 是否启用登录失败处理功能，可采取结束会话、限制非法登录次数和自动退出等措施

设置6次登陆失败后锁定帐户，锁定时间3000秒

vi /etc/pam.d/system-auth
auth required pam_tally.so onerr=fail deny=6 unlock_time=3000
(放在system-

auth文件的第一行，若对root用户起作用，加上even_deny_root root_unlock_time=3

000)

解锁用户 faillog -u <用户名》-r

1.1.4 d) 当对服务器进行远程管理时，是否采取必要措施，防止鉴别信息在网络

传输过程中被窃听

远程管理时应启用SSH等管理方式，加密管理数据，防止被网络窃听。

查看sshd服务状态:

service sshd status

1.1.5 e) 是否为操作系统和数据库系统的不同用户分配不同的用户名，确保用户

名具有唯一性

操作系统的管理员用户为root

Oracle数据库系统的用户为oracle，Oracle grid用户为grid

1.2 访问控制

1.2.1 a) 是否启用访问控制功能，依据安全策略控制用户对资源的访问 制定严格

的访问控制安全策略，根据策略控制用户对应用系统的访问，特别是文件操作、

数据库访问等，控制粒度主体为用户级、客体为文件或数据库表级。

1.2.2 b)是否根据管理用户的角色分配权限，实现管理用户的权限分离，仅授予管

理用户所需的最小权限 操作系统的管理员用户为root（组为root），Oracle数据库

系统的用户为oracle（组为oinstall等），cluster用户为grid 设置umask，系统缺省值

为022。

查看用户的umask：# umask

如需修改，在/etc/profile或用户home目录下的.profile中加入如下一行内容（以027为例）：

umask 027

1.2.3 c) 是否实现操作系统和数据库系统特权用户的权限分离 区分root和oracle用户权限

1.2.4 d) 是否限制默认帐户的访问权限，重命名系统默认帐户，修改这些帐户的默认口令

为每一个系统默认帐户重设口令。

1.2.5 e) 是否及时删除多余的、过期的帐户，避免共享帐户的存在 检查多余和过期的帐户并删除。以test用户为例：# rmuser test

1.3 安全审计

1.3.1 a) 审计范围是否覆盖到服务器和重要客户端上的每个操作系统用户和数据库用户 操作系统用户的审计由主机的审计服务进行管理，数据库用户的审计由Oracle数据库系统来管理。

开启内核audit系统

auditctl –e 1

开启审计服务守护进程：

service auditd start

设置开机自动启动，用root用户执行：

chkconfig auditd on

查看audit运行状态

auditctl –s
AUDIT_STATUS：enabled=1 flag=1 pid=1585 rate_limit=0 backlog_limit=256 lost=0 backlog=0

1.3.2 b) 审计内容是否包括重要用户行为、系统资源的异常使用和重要系统命令的使用等系统内重要的安全相关事件

查看已有的audit规则

```
# auditctl –l
To see unsuccessful open calls:
# auditctl -a exit,always -S open -F success=0
To watch a file for changes (2 ways to express):
# auditctl -w /etc/shadow -p wa
# auditctl -a exit,always -F path=/etc/shadow -F perm=wa
To recursively watch a directory for changes (2 ways to express):
# auditctl -w /etc/ -p wa
# auditctl -a exit,always -F dir=/etc/ -F perm=wa
```

添加一条audit规则，记录uid=500用户的所用open系统调用

auditctl -a entry，always -S open -F uid=500

删除这条audit规则

auditctl -d entry，always -S open -F uid=500

如不想看到用户登陆类型的消息，可以如下添加规则： #auditctl -a exclude，always -F msgtype=USER_LOGIN

这里过滤是以“消息类型”为对象的。

监视/etc/passwd文件被读、写、执行、修改文件属性的操作记录 #auditctl -w /etc/passwd -p rwax

查看程序所有的系统调用

#auditctl -a entry，always -S all -F pid=1005

查看指定用户打开的文件

#auditctl -a exit，always -S open -F auid=510

查看不成功的open系统调用

auditctl -a exit，always -S open -F success！=0

设置规则和显示规则的命令样例列出如下：

```
#auditctl -a entry，always -S all -F pid=1005
#auditctl -l
LIST_RULES：entry，always pid=1005（0x3ed）syscall=all
```

1.3.3 c) 审计记录是否包括事件的日期、时间、类型、主体标识、客体标识和结果等　查看audit日志文件：

#cat /var/log/audit/audit.log

type=SYSCALL msg=audit(1175176190.105:157): arch=40000003 syscall=5 success=yes exit=4 a0=bfba161c a1=8000 a2=0 a3=8000 items=1 ppid=4457 pid=4462 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 comm="less" exe="/usr/bin/less" subj=unconstrainedkey="LOG_audit_log"

1.3.4 d) 是否能够根据记录数据进行分析，并生成审计报表 aureport 命令格式：

aureport [选项]

主要选项：

报告关于访问向量缓冲（access vector cache，AVC）的消息

报告关于配置修改的消息

报告关于crypto事件的消息

报告关于事件的消息

报告关于文件的消息

报告关于主机的消息

报告关于登录的消息

报告关于账户修改的消息

报告关于Mandatory Access Control（MAC）事件的消息

报告关于进程的消息

报告关于系统调用的消息

报告关于终端的消息

如果执行aureport时没有使用任何选项，则会显示如汇总报表。

要显示每个日志的启动和停止时间，可以添加-t选项：

aureport -<flag> -i -t

要仅显示失败事件，则使用- -

failure，注意这个选项前面有两条虚线而不是一条： aureport -<flag> -i —— failed

要仅显示成功事件，则使用- -

success，注意这个选项前面有两条虚线而不是一条： aureport -<flag> -i —— success

要产生来自一个日志文件的报表而不是默认报表，则可用-if选项指定它：

aureport -<flag> -i -if /var/log/audit/audit.log.1

1.3.5 e) 是否保护审计进程，避免受到未预期的中断 只有root权限能够控制审计系统的开启和关闭。

1.3.6 f) 是否保护审计记录，避免受到未预期的删除、修改或覆盖等

/var/log/audit/目录和其中的所有审计日志文件只对根用户可读。

1.4 剩余信息保护

1.4.1 a) 是否保证操作系统和数据库系统用户的鉴别信息所在的存储空间，被释放或再分配给其他用户前得到完全清除，无论这些信息是存放在硬盘上还是在内存中

检查/etc/passwd,/etc/group等系统配置文件，保证没有已删用户的数据残留，在删除用户后确认Home目录同时被删除

1.4.2 b) 是否确保系统内的文件、目录和数据库记录等资源所在的存储空间，被释放或重新分配给其他用户前得到完全清除

清理回收站，/tmp,/var/tmp等临时目录下可删除的文件，删除数据库用户的同时删除其拥有的数据库对象和表空间

1.5 资源控制

1.5.1 a) 是否通过设定终端接入方式、网络地址范围等条件限制终端登录

禁止root远程登录

vi /etc/ssh/sshd_config :
（修改下面两条，需重启sshd：service sshd restart）PermitRootLogin no
屏蔽banner信息

vi /etc/ssh/sshd_config : 注释掉banner的相关条目 #Banner /some/path

vi /etc/xinetd.d/telnet

把disable项改为yes,即disable = yes # services xinetd restart

"限制终端接入方式为经过数据加密的ssh方式，关闭telnet，xdmcp，XVnc等连接方式。在/etc/hosts.deny和/etc/hosts.allow中配置可接入的IP地址范围"

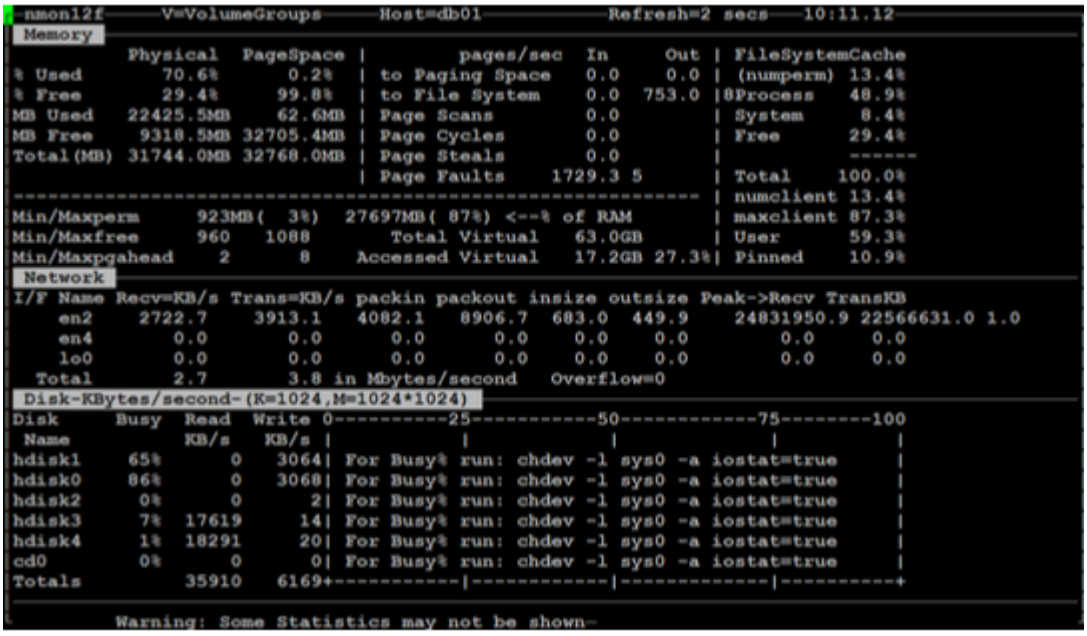
1.5.2 b) 是否根据安全策略设置登录终端的操作超时锁定 开启屏幕保护，长时间无操作后锁定屏幕。Ssh连接长时间无请求自动断开连接。"

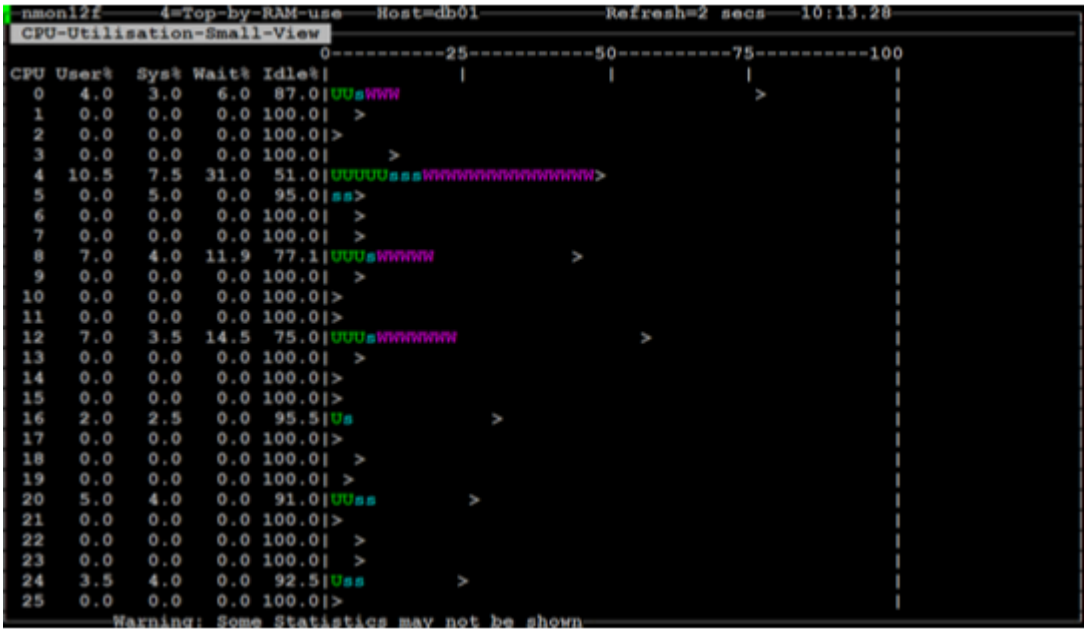
```
ClientAliveInterval 60
ClientAliveCountMax 3
```

1.5.3 c) 是否对重要服务器进行监视，包括监视服务器的CPU、硬盘、内存、网络等资源的使用情况

可使用*stat命令查看服务器资源使用情况　或安装nmon工具，对系统资源使用情况进行监控和统计。

nmon监控命令：# /nmon/nmon



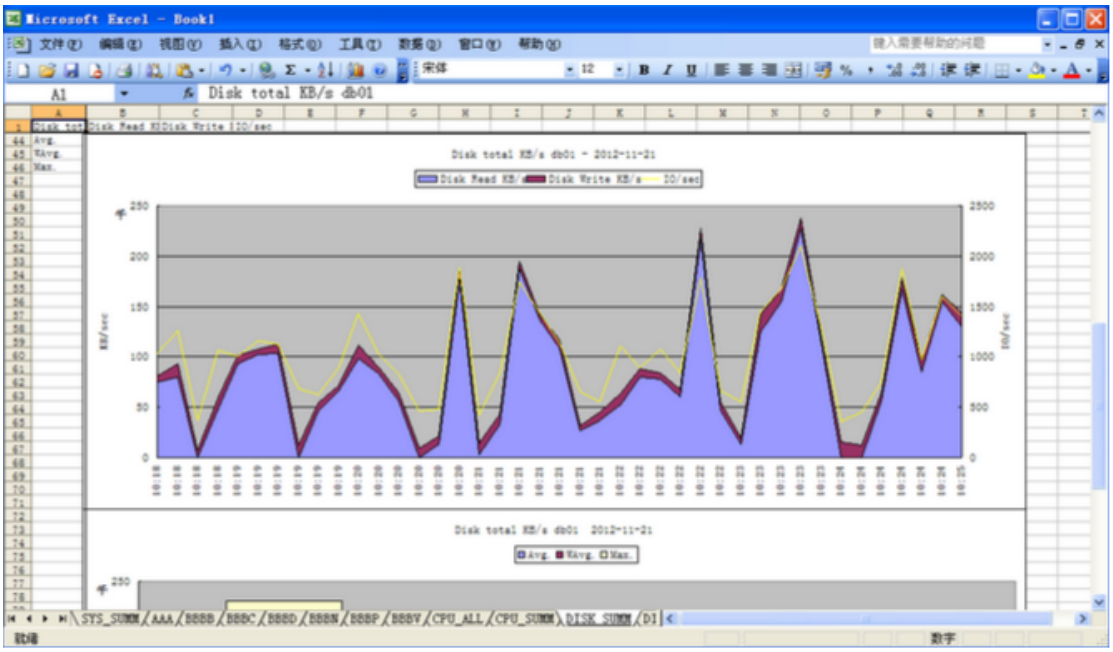


将数据捕获到文件，便于以后分析和绘制图形：

/nmon/nmon -fT -s 10 –m /nmon

（使用# /nmon/nmon -h查看nmon命令的用法）

可将生成的.mon文件ftp到windows主机，用nmon analyzer工具（基于excel宏）对数据进行分析和绘制图形：



1.5.4 d) 是否限制单个用户对系统资源的最大或最小使用限度

在/etc/pam.d/login 文件中加入:
session required /lib/security/pam_limits.so

编辑vi /etc/security/limits.conf文件对用户可使用的系统资源数量进行限制。

1.5.5 e) 是否能够对系统的服务水平降低到预先规定的最小值进行检测和报警

编写shell脚本对cpu，内存，磁盘使用情况进行检测，超过阈值后发送告警邮件