Examen Administration des Systèmes et des Réseaux
Durée 2 heures
Documents autorisés
Portables, ordinateurs et téléphones, éteints

J-M Moreno

Jeudi 26 mars 2009

Attention

Sauf indication contraire, les questions sont indépendantes. Vous devez expliciter et argumenter vos réponses. Par ailleurs il n'y a pas forcément de « bonne » réponse, l'argumentation que vous adosserez à vos commentaires n'en sera que plus importante.

1 L'utilitaire *dig*, permet d'interroger le *DNS* (c'est une alternative à *nslookup*). L'exemple ci-dessous montre 2 types de requêtes :

```
messidor# dig +short A moumoune.informatique.univ-paris-diderot.fr
194.254.199.47
messidor# dig +short -x 194.254.199.47
minette.informatique.univ-paris-diderot.fr.
messidor#
```

La première permet d'obtenir le *RR* de type *A* (adresse *IPv4*), la seconde une résolution dans le domaine inverse grâce à l'option *-x*. L'option *+short* n'est là que pour obtenir une réponse sous une forme concise... Les réponses aux 2 requêtes vous paraissent elles être cohérentes entre elles ? Commentez. Si vous décelez un problème d'où peut-il provenir ?

2 La commande *find* permet de parcourir une arborescence du système de fichiers en pratiquant une sélection et en exécutant au besoin un traitement sur cette sélection. Ainsi :

```
find /info -type f -name "*toto*" -size +100000c -exec chown marcel {} \;
```

permet de donner comme propriétaire l'utilisateur *marcel*, à tout fichier contenant *toto* dans son nom et de taille supérieure ou égale à 100 000 caractères. Par ailleurs vous connaissez l'utilitaire *tar* qui permet, entre autres, de réaliser des sauvegardes. Que peut apporter leur utilisation conjointe ? Par ailleurs à travers de quel autre utilitaire serait-il utile de les utiliser ?

3 L'exécution de la commande *ifconfig* ci-dessous et le message consigné dans */var/adm/messages* ont-ils, selon vous un rapport de cause à effet ?

```
aigreurs# ifconfig dmfe0:2 inet 194.254.199.99 netmask 255.255.255.0 up
aigreurs# ifconfig dmfe0:2
dmfe0:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 194.254.199.99 netmask ffffff00 broadcast 194.254.199.255
aigreurs# tail /var/adm/messages
Mar 19 17:16:39 germinal ip: [ID 903730 kern.warning] WARNING: IP: Hardware
address '00:0a:95:93:c1:18' trying to be our address 194.254.199.099!
aigreurs#
```

Expliquez la signification du message et, dans tous les cas, donnez en la raison. Que pourrait laisser supposer l'arrivée inopinée de ce message ?

4 **Le protocole *SMTP* possède un mécanisme pour faire suivre le courrier. Sous *Unix* il se matérialise par un fichier *.forward*, placé sur le catalogue racine du destinataire, contenant le *récipient* final du courrier. Si l'utilisateur *marcel* place dans le fichier *.forward artichaud@legumes.gouv.fr*, son courrier sera expédié à cette dernière adresse. Sous *Unix*, les utilitaires de transport de courrier, bien souvent *sendmail*, refusent de prendre en compte cette redirection si les droits ne sont pas au moins 755 (rwxr-xr-x), i.e. il faut que l'écriture soit, au plus, réservée au propriétaire. Cela vous paraît il justifié ? Si oui, indiquez comment vous pourriez utiliser cette faille éventuelle.**
Dans le cas ou les droits du fichier *.forward* sont 400 (r - - - - - - - -, lisible uniquement par l'utilisateur et ne pouvant être écrit par personne), mais où le catalogue le contenant peut être modifié par d'autres utilisateurs que le propriétaire, la réexpédition est encore refusée. Trouvez vous cela exagéré ?

5 **Le programme *C faux-gros.c* ci-dessous se contente de créer un fichier nommé *gros-lapin*.**

```
#include <sys/types.h>
#include <fcntl.h>
#include <unistd.h>
int f;
main()
{
    if((f=open("gros-lapin", O_WRONLY|O_CREAT|O_LARGEFILE,0600))==-1) {
        perror("y'a un probleme");
        exit(1);
    };
    llseek(f,100000000000LL,SEEK_SET);
    if(write(f,"Lapin",5)==-1) {
        perror("y'a un probleme");
        exit(1);
    }
    close(f);
    exit(0);
}
```

L'option *O_LARGEFILE*, la fonction *llseek* permettent d'utiliser des valeurs sur 64 bits. Le format *LL* est celui des entiers de type *long long*. On trouvera ci-dessous le résultat de l'exécution de ce programme.

```
<ouindose-321-[15:34]>% df -kh .
Système de fichiers  taille utilisé  dispo capacité  Monté sur
/dev/dsk/c1t2d0s3       16G   9,5G   6,1G    61%    /users
<ouindose-322-[15:34]>% ./faux-gros
<ouindose-323-[15:35]>% df -kh .
Système de fichiers  taille utilisé  dispo capacité  Monté sur
/dev/dsk/c1t2d0s3       16G   9,5G   6,1G    61%    /users
<ouindose-324-[15:35]>% ls -l gros-lapin
-rw-------   1 jmm      other    100000000005 mars 20 15:35 gros-lapin
<ouindose-325-[15:35]>% ls -lh gros-lapin
-rw-------   1 jmm      other        93G mars 20 15:35 gros-lapin
<ouindose-326-[15:35]>%
```

Qu'en pensez vous ? Cela peut-il avoir des conséquences sur la bonne marche du système ? Que se passerait il en cas de suppression du *gros-lapin* ?

6 **La commande *dladm* permet, sous *Solaris 10*, de gérer des liens et en particulier des agrégations. Sachant que les périphériques *nge*n sont des interfaces *Ethernet*, que pouvez dire en étudiant ce qui suit :**

```
amertume # dladm show-aggr
key: 1 (0x0001) policy: L4      address: 0:14:4f:ed:f8:81 (auto)
    device       address                 speed           duplex  link    state
    nge1         0:14:4f:ed:f8:81          1000  Mbps    full    up      attached
    nge0         0:14:4f:ed:f8:80          1000  Mbps    full    up      attached
amertume # dladm show-link aggr581001
aggr581001      type: vlan 581  mtu: 1500       aggregation: key 1
amertume # dladm show-link aggr929001
aggr929001      type: vlan 929  mtu: 1500       aggregation: key 1
amertume # ifconfig aggr929001 ; ifconfig aggr581001
aggr929001: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index 3
    inet 192.168.70.83 netmask ffffff00 broadcast 192.168.70.255
    ether 0:14:4f:ed:f8:81
aggr581001: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 1500 index 2
    inet 194.254.199.83 netmask ffffff00 broadcast 194.254.199.255
    ether 0:14:4f:ed:f8:81
amertume #
```

L'affichage qui suit est obtenu sur un *switch*. Il concerne respectivement une agrégation, des *VLAN* et des adresses *MAC*.

```
traveling-riverside-blues# show trunks 15,16
 Load Balancing
  Port | Name                              Type      | Group Type
  ---- + --------------------------------- --------- + ----- -----
  15   | aggregat trk5                     100/1000T | Trk5  LACP
  16   | aggregat trk5                     100/1000T | Trk5  LACP
traveling-riverside-blues# show vlans ports trk5
 Status and Counters - VLAN Information - for ports Trk5
  VLAN ID Name                | Status     Voice Jumbo
  ------- ------------------- + ---------- ----- -----
  581     ufop7               | Port-based No    No
  929     rc07                | Port-based No    No
traveling-riverside-blues# show mac-address trk5
 Status and Counters - Port Address Table - Trk5
  MAC Address
  -------------
  00144f-edf881
traveling-riverside-blues#
```

Commentez cet affichage. Peut on relier celui-ci à l'affichage précédent provenant de *dladm* ?

7 **Pour éviter la propagation du mot de passe de *root*, on peut utiliser sous *UNIX* le mécanisme dit «des *sudoers*», à l'aide de la commande *sudo*. Celui-ci est basé sur une liste d'utilisateurs autorisés à exécuter des commandes dans un mode privilégié. Ainsi si l'utilisateur *marcel* est défini comme *sudoer*, l'utilisation de *sudo ls* lui permettra d'exécuter *ls* en possédant les droits de *root*. Avant toute première utilisation de *sudo* le mot de passe de l'utilisateur, *marcel*, est demandé. Ce mécanisme est utilisé de façon intensive sous *MacOS X*. Vous trouverez ci-dessous un exemple.**

```
coccinelle:~ jmm$ mkdir /toto
mkdir: /toto: Permission denied
coccinelle:~ jmm$ sudo mkdir /toto
Password:
coccinelle:~ jmm$ ls -ld /toto
```

```
drwxr-xr-x  2 root  admin  68 Mar 21 11:41 /toto
coccinelle:~ jmm$
```

Faites une analyse critique[1] de l'utilisation de cette commande et de son bien, ou mal, fondé.

8 Que vous inspire la suite de commandes qui suit :

```
amertume # who am i
root       pts/3        Mar 20 21:46    (localhost)
amertume # rm -r / toto
```

Trouveriez vous raisonnable de valider la commande *rm*[2] ? Peut on en déduire une ligne de conduite à tenir ? La commande *sudo* peut-elle apporter une aide ?

9 Commentez l'affichage de la commande *top* ci-dessous :

```
<ouindose-68-[12:27]>% top -Ujmm -b
load averages:  1.98,  1.21,  0.54;               up 94+22:30:09       12:27:16
69 processes: 65 sleeping, 1 stopped, 3 on cpu
CPU states: 49.8% idle, 50.0% user,  0.2% kernel,  0.0% iowait,  0.0% swap
Kernel: 200 ctxsw, 35 trap, 260 intr, 427 syscall, 10 flt
Memory: 8192M phys mem, 6848M free mem, 8193M total swap, 8193M free swap

   PID USERNAME NLWP PRI NICE  SIZE   RES STATE    TIME    CPU COMMAND
  6387 jmm         1   0    0 7630M  544K cpu/3    4:41 24.85% carotte
  6389 jmm         1   0    0  992K  640K cpu/1    4:32 24.85% radis
  6424 jmm         1  59    0 2528K 1560K cpu/2    0:00  0.00% top
 28196 jmm         1  59    0 3680K 1896K stop     0:00  0.00% vi
 26841 jmm         1  59    0 3456K 3048K sleep    0:00  0.00% tcsh
  6299 jmm         1  59    0 3240K 2848K sleep    0:00  0.00% tcsh
 26839 jmm         1  59    0 7560K 2688K sleep    0:00  0.00% sshd
  6297 jmm         1  59    0 7560K 2680K sleep    0:00  0.00% sshd
  6409 jmm         1  59    0 3816M  656K sleep    0:00  0.00% navet
```