

Gestion des équipements réseau avec GNU/Linux

Philippe Latu

philippe.latu(at)linux-france.org

<http://www.linux-france.org/prj/inetdoc/>

Historique des versions		
\$Revision: 1379 \$	\$Date: 2009-04-16 23:19:32 +0200 (jeu 16 avr 2009) \$	\$Author: latu \$
Nouvelle édition de la section sur le service TFTP avec tftpd-hpa		
Résumé		
Cet article présente les principaux modes de gestion d'équipements réseau (routeurs, commutateurs, points d'accès, etc.) à partir d'un système GNU/Linux.		

Table des matières

1. Copyright et Licence	3
1.1. Méta-information	3
1.2. Équipements testés	3
1.3. Logiciels utilisés	3
2. Sécurité et architecture	4
2.1. Outils GNU/Linux	5
3. Accès à la console	5
3.1. Ports série	6
3.2. Administration des accès	6
3.3. Installation et configuration par défaut de minicom	7
3.3.1. Paramètres du port série	7
3.3.2. Fichier de configuration par défaut	7
3.4. Configuration utilisateur	8
3.5. Exemples d'utilisation de minicom	9
3.5.1. Séquences Break	9
3.5.2. Transfert d'une image système via xmodem	10
4. Echanges avec le protocole TFTP	12
4.1. Installation et configuration du service tftpd	12
4.2. Exemples d'utilisation du service tftpd	15
4.2.1. Mise à jour système d'un routeur Cisco 2851	15
4.2.2. Mise à jour système d'un commutateur Cisco 2950	16
4.2.3. Sauvegarde de la configuration d'un équipement	16
4.2.4. Mise à jour de la configuration d'un équipement	16
4.3. Un soupçon de sécurité	17
5. Collecte des journaux	18
5.1. Installation et configuration du service syslog	18
5.2. Exemple d'utilisation du service syslog	18
5.3. Utilisation de syslog-ng	19
5.4. Traitement des journaux	20
5.5. Un soupçon de sécurité	20
6. Synchronisation des horloges avec le protocole NTP	22
6.1. Installation et configuration du service ntp	22
6.2. Validation de la configuration ntp	22
6.3. Encore un soupçon de sécurité	22
7. Introduction à la métrologie avec mrtg	24
7.1. Installation et configuration du service mrtg	24
7.2. Exemple d'utilisation sur un commutateur 2950	24

7.3. Encore un soupçon de sécurité	25
8. Pour aller plus loin !	27
A. Configuration type du filtrage réseau	28

1. Copyright et Licence

Copyright (c) 2000,2009 Philippe Latu.
 Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2009 Philippe Latu.
 Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.2 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

1.1. Méta-information

Cet article est écrit avec *DocBook*¹ XML sur un système *Debian GNU/Linux*². Il est disponible en version imprimable aux formats PDF et Postscript : [devmgmt.pdf](#)³ | [devmgmt.ps.gz](#)⁴.

1.2. Équipements testés

Tous les tests présentés dans cet article sont effectués sur des équipements Cisco™. Cela ne signifie pas que ces équipements sont les seuls à être gérés de cette façon. Bien au contraire ! Les outils de gestion décrits sont utilisables avec n'importe quel équipement offrant la fonctionnalité étudiée.

1.3. Logiciels utilisés

Toutes les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. Comme la distribution *Debian GNU/Linux* est utilisée pour l'ensemble des supports du projet *inetdoc.LINUX*, voici une liste des paquets contenant les commandes nécessaires :

- **minicom** - friendly menu driven serial communication program.
- **lrzsz** - Tools for zmodem/xmodem/ymodem file transfer.
- **tftpd-hpa** - HPA's tftp server.
- **netkit-inetd** - The Internet Superserver.
- **tcpd** - Wietse Venema's TCP wrapper utilities.
- **sysklogd** - System Logging Daemon.
- **syslog-ng** - Next generation logging daemon.
- **logcheck** - Mails anomalies in the system logfiles to the administrator.
- **ntp** - Network Time Protocol: daemon for simple systems.
- **mrtg** - multi router traffic grapher.

¹ <http://www.docbook.org>

² <http://www.debian.org>

³ <http://www.linux-france.org/prj/inetdoc/telechargement/devmgmt.pdf>

⁴ <http://www.linux-france.org/prj/inetdoc/telechargement/devmgmt.ps.gz>

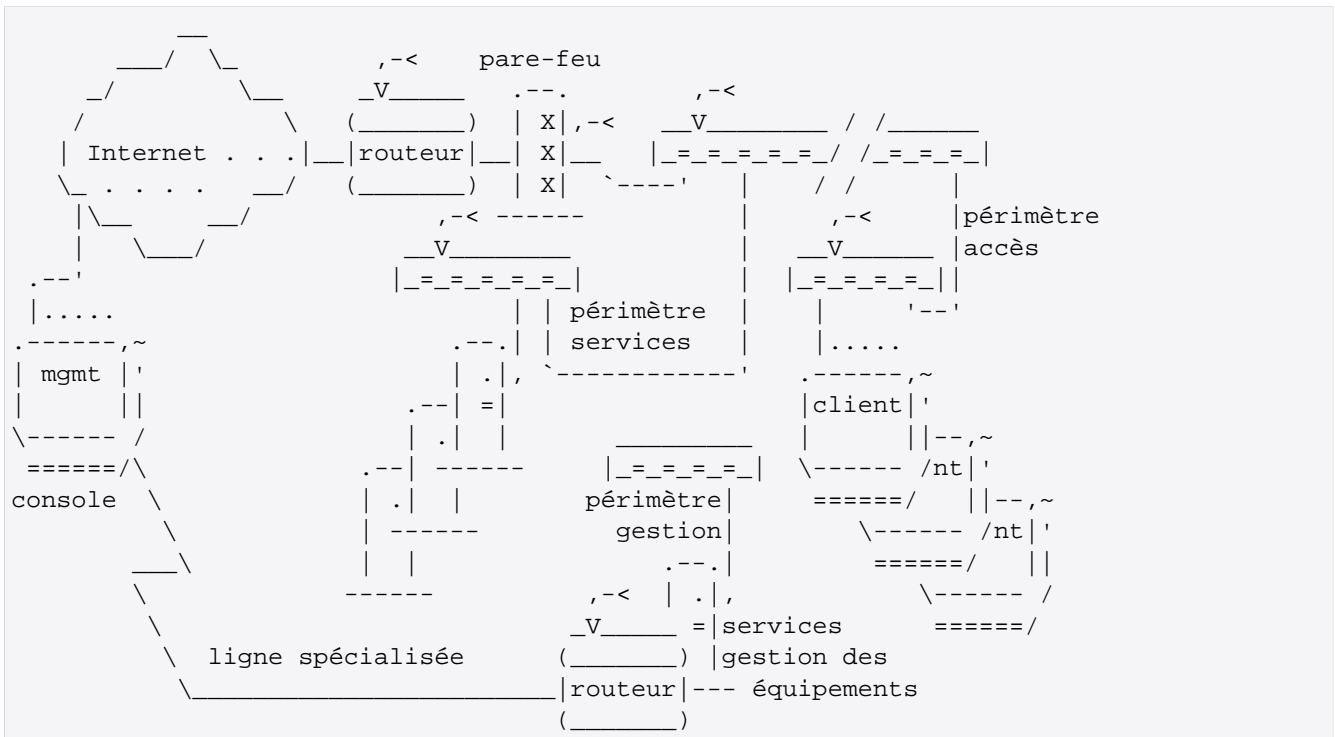
2. Sécurité et architecture

Par définition, la sécurité d'un système d'information est une chaîne de fonctions dont la résistance est égale à celle du maillon le plus faible. La thématique de cet article peut être considérée comme une fonction de sécurité. En effet, il est essentiel d'avoir un accès sûr et continu aux équipements d'interconnexion et/ou d'être capable de redémarrer avec des jeux de configurations fiables en cas de défaut.

L'objet de cet article n'est pas de développer en détails les aspects sécurité de la gestion des équipements réseau. Pour autant, on peut essayer d'énoncer «la règle d'or» de la sécurisation des équipements à partir l'expression consacrée en anglais : *toutes les opérations de gestion des équipements doivent s'effectuer hors de la bande passante utilisateur*. En anglais, on retrouve très fréquemment l'acronyme OOB pour *Out Of Band management*.

Dans la conception d'une architecture d'interconnexion réseau moderne, on doit systématiquement prévoir un périmètre dédié à la gestion de l'infrastructure. Les contours d'un tel périmètre varient en fonction du contexte. À minima, ce périmètre peut être constitué d'un VLAN pour un commutateur unique. Dès que le nombre des équipements s'étoffe un peu on doit disposer d'un réseau IP spécifique et de services complets : SSH, TFTP, SysLog, NTP, SNMP, DNS, WWW, etc.

Voici un exemple d'architecture type dans laquelle les signes ',-<' matérialisent les accès de gestion sur les équipements :



On retrouve ici les deux grandes catégories d'accès pour la gestion des équipements hors de la bande passante utilisateur.

Ligne spécialisée

L'utilisation d'une ligne spécialisée est généralement préconisée pour obtenir un accès physique indépendant à l'infrastructure à moindre coût. Une ligne spécialisée présente une sécurité intrinsèque puisque le canal de communication est dédié et la limitation du débit ne pose pas de problème puisque les opérations de gestion des équipements ne consomment que très peu de bande passante. Cette solution est généralement proposée dans les offres d'hébergements externalisés.

Réseau privé virtuel

Dans le cas où l'on ne dispose pas d'une deuxième voie d'accès physique à l'infrastructure, l'utilisation d'un réseau privé virtuel réservé à la gestion permet d'accéder aux équipements. Le principal défaut de cette solution est évident : le routeur «de tête» devient un point d'entrée encore plus sensible. Si on perd l'accès à ce routeur, on est coupé du reste de l'infrastructure.

2.1. Outils GNU/Linux

Relativement à l'architecture type présentée ci-dessus, les systèmes GNU/Linux conviennent parfaitement à la mise en oeuvre d'un ensemble de services spécifiques dédiés à la gestion des équipements réseau.

Tout d'abord, le serveur qui héberge les services du périmètre doit fonctionner avec GNU/Linux. C'est à cette condition que l'on pourra déployer les fonctions décrites dans la suite du document :

- Routage directement intégré au serveur avec *Quagga*⁵. Voir *Initiation au routage*⁶. Dans le cas d'un système d'information utilisant le protocole de routage OSPF, *Quagga*⁷ peut très bien servir de «démon de référence» dans les calculs de topologie du réseau effectués par les autres équipements.
- Centralisation des accès de configuration. Voir [Section 3, « Accès à la console »](#).
- Centralisation des copies de sauvegarde des configurations. Voir [Section 4, « Echanges avec le protocole TFTP »](#).
- [FIXME: compléter avec les autres fonctions de gestion].

Ensuite, pour accéder à distance aux services du périmètre de gestion des équipements, les outils GNU/Linux fournissent tout un panel de solutions de transmissions chiffrées. :

- On peut utiliser *OpenSSH*⁸ pour ouvrir une console à distance et effectuer les opérations de configuration. Voir [Section 3, « Accès à la console »](#).
- On peut utiliser *Stunnel*⁹ pour chiffrer et/ou traduire les communications sur un service (ou numéro de port) unique. Il est tout à fait possible d'échanger des fichiers d'image système ou de configuration via TFTP à travers un réseau public. *Stunnel*¹⁰ sert alors à chiffrer les communications sur le numéro de port 69 après avoir authentifié les deux extrémités à l'aide de leurs certificats respectifs. Voir [Section 4, « Echanges avec le protocole TFTP »](#).
- Enfin, on peut utiliser *OpenVPN*¹¹ pour un chiffrement global des communications entre la console et le serveur à travers un réseau public. Le coût de mise en oeuvre de cet outil est tellement dérisoire relativement aux solutions propriétaires, qu'il ne faut pas rater une occasion de l'utiliser !

3. Accès à la console

Même s'il est souvent possible de configurer un équipement à partir d'une interface Web, la console a toujours été l'interface de configuration la plus «sérieuse». Voici quelques arguments pour étayer cette idée :

- Les «conditions de sécurité» des accès Web de configuration d'équipement ont toujours été médiocres jusqu'à présent. Soit le serveur Web inclu dans l'équipement contient de nombreux trous de sécurité (il suffit d'interroger les bases de rapport d'incidents du *CERT: Vulnerabilities, Incidents & Fixes*¹² pour s'en convaincre) ; soit le coût financier et humain de modification de l'infrastructure d'accès aux équipements est trop important relativement aux bénéfices attendus.
- Les interfaces Web ne permettent pas d'appréhender globalement les paramètres de configuration. En effet, dès que le nombre des paramètres et des options devient important, les formulaires Web deviennent très lourds à gérer. Naviguer d'un paramètre à l'autre suppose plusieurs manipulations de formulaires alors que la même opération est immédiate à la console.

Sur les systèmes GNU/Linux, minicom est le «couteau suisse» idéal pour toutes les manipulations sur les liaisons séries. Avant d'aborder sa configuration et son utilisation, on vérifie que le système offre bien les fonctions nécessaires : [Section 3.1, « Ports série »](#) et [Section 3.2, « Administration des accès »](#).

⁵ <http://www.quagga.net/>

⁶ <http://www.linux-france.org/prj/inetdoc/guides/index.html#routage>

⁷ <http://www.quagga.net/>

⁸ <http://openssh.org/>

⁹ <http://www.stunnel.org/>

¹⁰ <http://www.stunnel.org/>

¹¹ <http://openvpn.net/>

¹² http://www.cert.org/nav/index_red.html

3.1. Ports série

Avant de s'attaquer à la configuration du logiciel, il faut s'assurer que le port série que l'on veut utiliser est disponible. Comme il s'agit d'un périphérique matériel, c'est dans le noyau Linux que l'on trouve les informations utiles. Normalement, tous les noyaux livrés avec les distributions intègrent directement les pilotes des ports série.

Il est facile de vérifier que les ports sont disponibles à l'aide de la commande **dmesg** :

```
$ dmesg | less
<snip/>
Serial: 8250/16550 driver .Revision: 1.4 . 48 ports, IRQ sharing enabled
ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A
ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A
```

Les informations ci-dessus indiquent que l'on dispose de deux ports : `ttyS0` et `ttyS1` pilotés par le composant 16550A.

Dans le cas où l'on utilise un noyau «compilé maison», il faut retenir les options indiquées ci-dessous à partir des menus de configuration du noyau :

```
Linux Kernel Configuration
  Device Drivers
    Character devices
      Serial drivers

    <*> 8250/16550 and compatible serial support
    [*] Console on 8250/16550 and compatible serial port
    < > 8250/16550 PCMCIA device support
    [ ] 8250/16550 device discovery via ACPI namespace
    (4) Maximum number of non-legacy 8250/16550 serial ports
    [*] Extended 8250/16550 serial driver options
    [*] Support more than 4 legacy serial ports
    [*] Support for sharing serial interrupts
    [ ] Autodetect IRQ on standard ports (unsafe)
    [*] Support special multiport boards
    [*] Support RSA serial ports
    --- Non-8250 serial port support
```

3.2. Administration des accès

Une fois que le noyau Linux assure le pilotage des ports série, il faut administrer les droits d'accès à ces ports. Par défaut, les périphériques du système de fichiers correspondant aux ports série appartiennent au groupe `dialout` :

```
$ ls -l /dev/ttyS?
crw-rw---- 1 root dialout 4, 64 2003-02-13 16:41 /dev/ttyS0
crw-rw---- 1 root dialout 4, 65 2002-03-14 22:51 /dev/ttyS1
crw-rw---- 1 root dialout 4, 66 2002-03-14 22:51 /dev/ttyS2
crw-rw---- 1 root dialout 4, 67 2002-03-14 22:51 /dev/ttyS3
```

Pour qu'un utilisateur normal ait accès aux ports série, il doit appartenir au groupe `dialout` :

```
# adduser etu dialout
Ajout de l'utilisateur etu au groupe dialout...
Fait.
```

L'utilisateur `etu` doit se reconnecter pour bénéficier des ces nouveaux droits d'accès.

3.3. Installation et configuration par défaut de minicom

Comme indiqué dans la section [Section 1.3, « Logiciels utilisés »](#), seuls les paquets de la distribution *Debian GNU/Linux* sont présentés ici. L'installation de minicom se résume donc à l'instruction suivante :

```
# apt-get install minicom lrzsz
```

Après cette étape, il faut créer le fichier de configuration par défaut. Ce fichier sert de dénominateur commun à l'échelle du système. Les paramètres de ce fichier : `/etc/minicom/minirc.dfl` sont activé à chaque lancement du programme minicom. L'utilisateur est ensuite libre de tout modifier et de se créer ses propres paramètres de configuration par défaut.

Voici un exemple de première exécution du programme :

```
# minicom -s
<snip/>

-----[configuration]-----
| Noms de fichiers et chemins          |
| Protocoles de transfert              |
| Configuration du port série ❶       |
| Modem et appel                      |
| Ecran et clavier                    |
| Enregistrer config. sous dfl ❷     |
| Enregistrer la configuration sous... |
| Sortir                               |
| Sortir de Minicom                   |
-----
```

Pour aller au plus simple, on se contente de fixer les paramètres du port série et de sauvegarder.

- ❶ Première étape, le port série : voir [Section 3.3.1, « Paramètres du port série »](#)
- ❷ Enregistrement du fichier `/etc/minicom/minirc.dfl`.

3.3.1. Paramètres du port série

Les valeurs présentés ci-dessous correspondent à la grande majorité des paramètres par défaut des équipements d'interconnexion.

```
-----
| A -          Port série : /dev/ttyS0 ❶ |
| B - Emplacement du fichier de verrouillage : /var/lock |
| C -          Programme d'appel intérieur : |
| D -          Programme d'appel extérieur : |
| E -          Débit/Parité/Bits : 9600 8N1 ❷ |
| F -          Contrôle de flux matériel : Oui ❸ |
| G -          Contrôle de flux logiciel : Non |
|          Changer quel réglage ? |
-----
```

- ❶ En fonction de la liste des ports série disponibles (voir [Section 3.1, « Ports série »](#)), on saisit le nom du premier port série.
- ❷ En fonction des paramètres de l'équipement à gérer, on fixe les paramètres de la liaison à l'aide du menu suivant :
- ❸ Toujours en fonction de l'équipement à gérer, on choisit le contrôle de flux matériel. C'est le mode le plus fiable.

3.3.2. Fichier de configuration par défaut

Voici un exemple de fichier `/etc/minicom/minirc.dfl` obtenu à partir des réglages ci-dessus.

```
# Fichier généré automatiquement - utilisez « minicom -s »
```

```
# pour changer les paramètres
pu port          /dev/ttyS0
pu baudrate      9600
pu bits          8
pu parity        N
pu stopbits      1
pu scriptprog    /usr/bin/runscript
pu minit         ~^M~ATZ^M~
pu mreset        ~^M~ATZ^M~
pu escape-key    Escape (Meta)
```

Il est aussi possible d'effacer toutes les commandes *Hayes* de dialogue avec les modems. Comme *minicom* possède une option de démarrage éliminant tous les dialogues modems, il n'est pas nécessaire de traiter ce problème au niveau système. Il se peut qu'un utilisateur ait effectivement besoin d'un modem.

3.4. Configuration utilisateur

Le jeu des fonctionnalités de *minicom* est très riche. Il suffit de consulter les pages de manuels pour s'en convaincre : **man minicom**. L'objet de ce document n'étant de dresser un catalogue exhaustif de ces fonctions, voici un exemple type de configuration utilisateur :

```
$ MINICOM="-o❶ -8❷ -l❸ -m❹ -con❺ -t linux❻" ; export MINICOM❼
```

- ❶ Option `-o` : *minicom* n'exécute pas les codes d'initialisation. C'est cette option qui permet d'éviter les commandes *Hayes* au démarrage. Dans le contexte de connexion à un équipement réseau, les dialogues modems sont totalement inutiles.
- ❷ Option `-8` : les caractères codés sur 8 bits sont transmis sans aucune modification. Cette option permet d'afficher les caractères accentués en français.
- ❸ Option `-l` : traduction littérale des caractères avec le bit de poids fort à 1. Avec cette option *minicom* n'essaie pas de traduire les caractères IBM en ASCII.
- ❹ Option `-m` : redéfinit la touche de commande avec la touche **Meta** ou **Alt**. Avec un système GNU/Linux, on rencontre deux cas de figure classiques :

Sans interface graphique

```
$ env | grep -i term
TERM=linux
```

Il vaut mieux éviter l'option `-m` et utiliser la séquence de touches **Ctrl+a** pour accéder aux fonctions. Par exemple, on accède au menu principal avec la séquence **Ctrl+a** puis **z**.

Avec interface graphique

```
$ env | grep -i term
TERM=xterm
```

L'option `-m` est utile pour simplifier l'appel aux fonctions. Il suffit d'utiliser la touche **Alt**. Par exemple, on accède au menu principal avec la séquence **Alt+z**.

- ❺ Option `-c` : utilisation de la couleur. On ajoute `on` ou `off` pour activer ou désactiver l'affichage des couleurs.
- ❻ Option `-t` : définition du type de terminal : `linux` dans ce cas. Cette option est très pratique pour conserver les dimensions de la console avec les environnements graphiques utilisateur. Il est fréquent que la même fenêtre de consoles serve à gérer plusieurs équipements et de la documentation. Sans cette option, *minicom* redimensionne la console avec une largeur de 80 caractères.
- ❼ Positionnement de la variable d'environnement `MINICOM`. Cette variable est consultée à chaque exécution de *minicom*.

Une fois que l'on est satisfait de son jeu d'options *minicom*, il est possible de le conserver dans la configuration du *shell* :

```
$ echo export MINICOM="\-o -8 -l -m -con -t linux\" >> ~/.bash_profile
```

De cette façon, toute ouverture d'un nouveau *shell* comprendra la variable d'environnement `MINICOM` et ses options.

3.5. Exemples d'utilisation de minicom

3.5.1. Séquences Break

Les exemples d'utilisation des séquences `Break` sont nombreux. Le plus souvent, il s'agit d'interrompre brutalement le chargement du système d'exploitation sur un équipement. À la suite de cette opération, on peut charger une nouvelle image système ou reprendre la main sur un équipement dont on a perdu les mots de passe.

Suivant la configuration de minicom, les séquences de touches diffèrent.

- Avec un codage clavier sur 7 bits, la séquence est : **Ctrl+a** puis **f**.
- Avec un codage clavier sur 8 bits, la séquence est : **Alt+f**.

Initialisation d'un commutateur Cisco™ 2950

Pour interrompre le chargement du système d'un commutateur de ce modèle il faut que la variable d'environnement `ENABLE_BREAK` soit préalablement positionnée sur `on`.

```
sw1#sh boot
BOOT path-list:
Config file:      flash:/config.text
Private Config file: flash:/private-config.text
Enable Break:    ON
Manual Boot:     no
HELPER path-list:
NVRAM/Config file
  buffer size:   32768
```

Lors du redémarrage de l'équipement il est possible d'utiliser la séquence **Ctrl+a** puis **f** pour envoyer un `Break` et interrompre le chargement du système.

❶ Résultat de la séquence `Break`.

```
flashfs[0]: Bytes available: 3169792
flashfs[0]: flashfs fsck took 7 seconds.
...done initializing flash.
Boot Sector Filesystem (bs:) installed, fsid: 3
Parameter Block Filesystem (pb:) installed, fsid: 4
Loading "flash:/c2950-i6q412-mz.121-22.EA2.bin"...#####bad
Error loading "flash:/c2950-i6q412-mz.121-22.EA2.bin"

Interrupt within 5 seconds to abort boot process.
Boot process failed...

The system is unable to boot automatically.  The BOOT
environment variable needs to be set to a bootable
image.
```

❶switch:

Initialisation d'un routeur Cisco™ 2500

Pour interrompre le chargement du système IOS d'un routeur de cette marque, il faut envoyer un `Break` pendant les 60 premières secondes après la mise sous tension de l'équipement. Comme cet exemple est traité via un portable avec une interface graphique, la séquence de touches est : **Alt+f**.

❶ Résultat de la séquence `Break`.

```
System Bootstrap, Version 11.0(10c)XB2, PLATFORM SPECIFIC RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1998 by cisco Systems
2500 processor with 14336 Kbytes of main memory

❶Abort at 0x11198B6 (PC)
>
>o/r 0x2142
```

```
>
>b
```

3.5.2. Transfert d'une image système via xmodem

En reprenant le cas du commutateur Cisco™ 2950, on peut transférer une image du système d'exploitation via la connexion série avec le protocole xmodem. Ce n'est certainement pas la méthode la plus rapide mais lorsque toutes les informations de la mémoire *flash* ont été détruite, il ne reste pas beaucoup d'autre solution.

Voici un exemple de «catastrophe provoquée» :

1. On commence par détruire le contenu de la mémoire *flash*.

```
switch: format flash:
Are you sure you want to format "flash:" (all data will be lost) (y/n)?y
flashfs[0]: 0 files, 1 directories
flashfs[0]: 0 orphaned files, 0 orphaned directories
flashfs[0]: Total bytes: 7741440
flashfs[0]: Bytes used: 1024
flashfs[0]: Bytes available: 7740416
flashfs[0]: flashfs fsck took 5 seconds.
Filesystem "flash:" formatted
switch: dir flash:
Directory of flash:/

2   -rwx  285      <date>          env_vars

7739904 bytes available (1536 bytes used)
switch:
```

2. On prépare le transfert d'image système via xmodem sur le commutateur.

```
switch: flash_init
Initializing Flash...
...The flash is already initialized.
switch: load_helper
copy xmodem: flash:c2950-i6q412-mz.121-22.EA2.bin
Begin the Xmodem or Xmodem-1K transfer now...
C
```

3. On lance le transfert à partir de la séquence de touches **Ctrl+A** puis **s**.

- ❶ Sélection du protocole de transfert.
- ❷ Sélection du fichier image système à transférer.
- ❸ Transfert du fichier image système.

```
+-[Envoyer]--+
| zmodem      |
| ymodem      |
| xmodem❶    |
| kermit      |
| ascii      |
+-----+
<snip/>

+-----[Sélectionner un fichier pour envoyer]-----+
|Répertoire : /home/etu |
| [...]        |
| [ios]        |
| .bash_history |
| .bash_profile |
| .bashrc     |
| .viminfo    |
| c2950-i6q412-mz.121-22.EA2.bin❷
```

```
| c2950-i6q412-tar.121-22.EA2.tar  
| minicom.log  
|  
| (Échap pour sortir, Espace pour marquer)  
+-----+  
| [Aller] [Préc.] [Voir] [Marq.] [Dém .] [OK]  
<snip/>  
+-----[xmodem envoyer - Appuyez sur CTRL-C pour quitter]-----+  
| Sending c2950-i6q412-mz.121-22.EA2.bin, 24111 blocks: Give yo|  
| ur local XMODEM receive command now. |  
| Xmodem sectors/kbytes sent: 138/17kⓈ |  
| |  
| |  
+-----+
```

4. Echanges avec le protocole TFTP

Le protocole TFTP est défini dans le document [RFC1350 *The TFTP Protocol \(Revision 2\)*](http://www.faqs.org/rfcs/rfc1350.html)¹³. C'est un protocole très simple utilisé pour transférer des fichiers. Il s'appuie sur le protocole UDP au niveau transport. Ce protocole est limité à l'écriture et à la lecture de fichiers entre deux hôtes réseau. Il ne possède pas de fonction d'authentification et il est impossible de lister les fichiers à distance.

L'absence de complexité de ce protocole fait qu'il est implantable dans un espace mémoire réduit. C'est la raison pour laquelle on le retrouve dans la majorité des équipements réseau comme moyen d'échange d'images *firmware* des systèmes d'exploitation (et/ou) de fichiers de configuration.

Dans le contexte de ce document, le poste avec le système GNU/Linux doit échanger les fichiers avec l'équipement réseau. Il s'agit donc de configurer un service `tftpd` avec un minimum de précautions au niveau sécurité.

4.1. Installation et configuration du service tftpd

Comme indiqué dans la [Section 1.3, « Logiciels utilisés »](#), seuls les paquets de la distribution *Debian GNU/Linux* sont présentés ici. À l'heure actuelle, la distribution comprend trois paquets pour ce service : `tftpd`, `atftpd` et `tftpd-hpa`. Les principales différences entre ces trois paquets se situent aux niveaux des tailles maximales des fichiers transférés et de la gestion des droits sur l'arborescence du service TFTP. Le paquet `tftpd-hpa` offre le plus de possibilités dans ces domaines. Son installation et sa configuration sont présentées ici.

L'installation de `tftpd-hpa` et le contrôle de la version installée se résument aux intructions suivantes :

```
# apt-get install tftpd-hpa
<snipped/>
# dpkg -l *tftpd* | grep ^ii
ii tftpd-hpa 0.48-2.2 HPA's tftp server
```

Après ces étapes triviales, il faut configurer le service en effectuant les opérations suivantes.

Compte utilisateur système et arborescence

Cette partie comprend plusieurs étapes. On commence par la création du groupe système baptisé `tftp`.

```
# addgroup --system tftp
Ajout du groupe « tftp » (identifiant 130)...
Terminé.
```

On passe ensuite à la création du compte utilisateur système proprement dit ; baptisé lui aussi `tftp`. Ce compte fait partie du groupe créé ci-dessus. Il dispose d'une arborescence dont la racine est placée au niveau du répertoire `/var/lib/tftp/` et, comme ce compte est de type système, il ne doit être utilisable que par le service TFTP.

```
# adduser --system --ingroup tftp \
  --home /var/lib/tftp --disabled-password --disabled-login tftp
Ajout de l'utilisateur système « tftp » (identifiant : 122)...
Ajout du nouvel utilisateur « tftp » (identifiant : 122) avec le
groupe « tftp »...
Création du répertoire personnel « /var/lib/tftp »...
```

On complète le masque des permissions sur la racine de l'arborescence en donnant les droits d'écriture aux membres du groupe `tftp`.

```
# chmod 2775 /var/lib/tftp
```

On peut contrôler le résultat des opérations précédentes en visualisant les droits sur l'arborescence.

```
# ls -lAh /var/lib/ |grep tftp
drwxrwsr-x  2 tftp      tftp      4,0K sep 27 11:31 tftp
```

¹³ <http://www.faqs.org/rfcs/rfc1350.html>

Enfin, pour que cette nouvelle arborescence soit utilisable par le service TFTP, il faut éditer la configuration par défaut du paquet `tftpd-hpa` donnée dans le fichier `/etc/default/tftpd-hpa`.

```
#Defaults for tftpd-hpa
RUN_DAEMON="no"
OPTIONS="-l -s /var/lib/tftp"
```

Contrôle d'accès local au service

À la suite des opérations décrites ci-dessus, seuls les membres du groupe `tftp`, ont le droit d'écrire dans l'arborescence du service TFTP. Tous les utilisateurs du système ont le droit de lire le contenu de cette même arborescence. Pour autoriser un utilisateur à écrire dans le répertoire `/var/lib/tftp/`, il suffit de l'ajouter au groupe système dédié. Voici un exemple :

```
# adduser etu tftp
Ajout de l'utilisateur « etu » au groupe « tftp »...
Ajout de l'utilisateur etu au groupe tftp
Terminé.
```

Les droits sur le système de fichiers du serveur TFTP attribués de cette façon permettent à un utilisateur normal (pas nécessairement le super-utilisateur) de déposer des images de systèmes d'exploitation (et/ou) des de fichiers de configuration. Attention, ce type de modification des droits d'un compte utilisateur n'est active qu'après une déconnexion/reconnexion.

Contrôle d'accès réseau au service

Historiquement, TFTP fait partie de la catégorie des *small-services* ou des services qui ne possèdent aucun mécanisme de contrôle d'accès et d'authentification. Pour compenser ces défauts, le contrôle d'accès réseau est assuré par un autre service appelé `tcpwrapper` via le démon `inetd`. Pour mettre en place un contrôle d'accès réseau au service TFTP, on doit configurer ces deux applications.

On commence par le démon `inetd` dont le fichier de configuration est `/etc/inetd.conf`. Après l'installarion du paquet `tftpd-hpa`, une ligne a été automatiquement insérée dans ce fichier. On doit l'éditer pour être conforme aux éléments du système de fichiers configurés ci-dessus.

```
#:BOOT: TFTP service is provided primarily for booting.  Most sites
#       run this only on machines acting as "boot servers."
tftp      dgram    udp      wait    root    /usr/sbin/in.tftpd \
          /usr/sbin/in.tftpd -c❶ -u tftp❷ -s /var/lib/tftp❸
```



Note

Pour faciliter l'affichage, la ligne de configuration a été coupée avec le caractère '\'. Pour que l'exemple soit utilisable, il faut supprimer ce caractère et remplacer les options sur une ligne unique.

Les options retenues dans la ligne ci-dessus sont extraites des pages de manuels du service : `# man in.tftpd`.

- ❶ L'option `-c` autorise la création de fichier à partir de l'équipement réseau. Sans cette option, le comportement par défaut du service n'autorise pas l'*upload*. Cette option est très utile pour la sauvegarde directe des fichiers de configuration des équipements.
- ❷ L'option `-u` permet de désigner les identifiants (IDs) de l'utilisateur et du groupe propriétaires du processus du service. Comme nous avons créé un utilisateur système `tftp` dédié au service, c'est cet utilisateur qui doit être affecté ici.
- ❸ L'option `-s` désigne la racine du service dans l'arborescence du système de fichiers du serveur. Comme dans le cas précédent, on affecte le répertoire dédié.

Ensuite, on redémarre le service pour valider les paramètres de configuration mis en place et on visualise le résultat.

```
# /etc/init.d/openbsd-inetd restart
Restarting internet superserver: inetd.

# lsof -i | grep tftp
inetd      22794      root      7u  IPv4 133787      UDP *:tftp
```

Le résultat de la commande **lsotf** montre bien que le démon `inetd` est en écoute sur le port 69/udp sans restriction sur les adresses IP avec le caractère '*'.

Le rôle du `tcpwrapper` est justement d'assurer le contrôle d'accès réseau pour les services qui ne possèdent pas cette fonction.

La configuration du `tcpwrapper` se fait à l'aide des fichiers `/etc/hosts.allow` et `/etc/hosts.deny`. Ces deux fichiers permettent d'appliquer la politique de sécurité classique : «tout ce qui n'est pas explicitement autorisé est interdit».

Le fichier `/etc/hosts.deny` contient donc une instruction unique d'interdiction globale des accès depuis les réseaux externes au système.

```
# cat /etc/hosts.deny | grep -v ^#  
  
ALL: PARANOID EXCEPT 127.0.0.1
```

Dans le cas illustré dans ce document, il s'agit d'autoriser l'accès aux hôtes du réseau d'infrastructure des équipements réseau (192.168.2.0/24 par exemple). On édite le fichier `/etc/hosts.allow` en ajoutant l'accès au service `tftpd` depuis ce réseau :

```
# cat /etc/hosts.allow | grep -v ^#  
  
in.tftpd: 192.168.2.0/24
```

Enfin, on vérifie que les accès sont bien ouverts :

```
# tcpdchk -v  
Using network configuration file: /etc/inetd.conf  
  
>>> Rule /etc/hosts.allow line 14:  
daemons:  in.tftpd  
clients:  192.168.2.0/24  
access:   granted  
  
>>> Rule /etc/hosts.deny line 20:  
daemons:  ALL  
clients:  PARANOID EXCEPT 127.0.0.1  
access:   denied
```

4.2. Exemples d'utilisation du service tftpd

4.2.1. Mise à jour système d'un routeur Cisco 2851

Dans ce cas de figure, le transfert se fait du poste GNU/Linux vers l'équipement. Il n'y a donc aucun problème lié aux accès sur le système de fichiers côté GNU/Linux. Il suffit de placer une copie du fichier à transférer dans le répertoire `/var/lib/tftp/`.

Cet exemple est particulier puisque la taille du fichier image du système d'exploitation est importante. Le service `tftpd` est assuré à partir du paquet `tftp-hpa` qui autorise le transfert de blocs de données importants.

```
$ cd /var/lib/tftp
$ ll c2800nm-advipservicesk9-mz.124-11.T1.bin
-rw-r--r-- 1 etu tftp 38M 2007-03-20 21:10 c2800nm-advipservicesk9-mz.124-11.T1.bin
```

Côté routeur, il faut s'assurer que la mémoire flash dispose d'un espace libre suffisant pour accueillir la nouvelle image du système d'exploitation IOS. Au besoin, il faut effacer le fichier image en cours d'exécution.

```
router#sh flash:
-#- --length-- -----date/time----- path
1      1826 Jul 5 2006 23:53:52 +02:00 sdmconfig-28xx.cfg
2     4734464 Jul 5 2006 23:54:12 +02:00 sdm.tar
3     833024 Jul 5 2006 23:54:26 +02:00 es.tar
4     1052160 Jul 5 2006 23:54:40 +02:00 common.tar
5       1038 Jul 5 2006 23:54:54 +02:00 home.shtml
6     102400 Jul 5 2006 23:55:06 +02:00 home.tar
7     491213 Jul 5 2006 23:55:18 +02:00 128MB.sdf
8     1684577 Jul 5 2006 23:55:38 +02:00 securedesktop-ios-3.1.1.27-k9.pkg
9     398305 Jul 5 2006 23:55:56 +02:00 sslclient-win-1.1.0.154.pkg

54702080 bytes available (9314304 bytes used)
```

On lance le transfert en indiquant le fichier image sur le serveur TFTP comme source et la mémoire flash comme destination.

```
router#copy tftp://192.168.2.1/c2800nm-advipservicesk9-mz.124-11.T1.bin flash:
Destination filename [c2800nm-advipservicesk9-mz.124-11.T1.bin]?
Accessing tftp://192.168.2.1/c2800nm-advipservicesk9-mz.124-11.T1.bin...
Loading c2800nm-advipservicesk9-mz.124-11.T1.bin from 192.168.2.1 \
(via GigabitEthernet0/1): !!!!!!!!!!!!!!!!!!!!!!!
<snip/>

[OK - 39798360 bytes]

39798360 bytes copied in 580.728 secs (68532 bytes/sec)

router#sh flash:
-#- --length-- -----date/time----- path
1      1826 Jul 5 2006 23:53:52 +02:00 sdmconfig-28xx.cfg
2     4734464 Jul 5 2006 23:54:12 +02:00 sdm.tar
3     833024 Jul 5 2006 23:54:26 +02:00 es.tar
4     1052160 Jul 5 2006 23:54:40 +02:00 common.tar
5       1038 Jul 5 2006 23:54:54 +02:00 home.shtml
6     102400 Jul 5 2006 23:55:06 +02:00 home.tar
7     491213 Jul 5 2006 23:55:18 +02:00 128MB.sdf
8     1684577 Jul 5 2006 23:55:38 +02:00 securedesktop-ios-3.1.1.27-k9.pkg
9     398305 Jul 5 2006 23:55:56 +02:00 sslclient-win-1.1.0.154.pkg
10    39798360 Mar 24 2007 17:04:52 +01:00 c2800nm-advipservicesk9-mz.124-11.T1.bin

14901248 bytes available (49115136 bytes used)
```

4.2.2. Mise à jour système d'un commutateur Cisco 2950

Comme dans l'exemple précédent, le transfert se fait du poste GNU/Linux vers l'équipement et le fichier à transférer est dans le répertoire `/var/lib/tftp/`.

Côté équipement, il faut un minimum de configuration réseau IP pour communiquer avec le poste GNU/Linux.

```
sw1#sh run
Building configuration...
<snip/>
!
interface Vlan1
  no ip address
  no ip route-cache
  shutdown
!
interface Vlan2
  ip address 192.168.2.2 255.255.255.0
  no ip route-cache
!
```

On peut ensuite lancer le transfert à partir de l'interface de commande de l'équipement. Voici un exemple de mise à jour via tftpd :

```
sw1#archive tar /x tftp://192.168.2.1/c2950-i6q4l2-tar.121-22.EA2.tar flash:
```

L'accès à l'interface de commande de l'équipement réseau peut être obtenu à l'aide de minicom ou de telnet.

4.2.3. Sauvegarde de la configuration d'un équipement

Ici, le transfert se fait depuis l'équipement réseau vers le poste GNU/Linux sur lequel le service tftpd est exécuté.

Avec la configuration mise en place dans la partie *Contrôle d'accès réseau au service*, on a autorisé la création de fichiers dans l'arborescence sur service TFTP avec l'option `-c` du démon `in.tftpd` fourni avec le paquet `tftpd-hpa`.

En reprenant l'exemple du commutateur 2950, voici l'opération se résume au transfert de la configuration active.

```
sw1#copy run tftp://192.168.2.1
Address or name of remote host [192.168.2.1]?
Destination filename [sw1-config]?
!!
2748 bytes copied in 1.116 secs (2462 bytes/sec)
```

Dans les faits, il est beaucoup plus fréquent que l'on édite la configuration sur le poste GNU/Linux avant de la transférer vers l'équipement. Cette méthode permet d'assurer un suivi des évolutions de configuration via un système de contrôle de version tel que CVS ou SVN. On dispose ainsi d'un dépôt centralisé des configurations.

4.2.4. Mise à jour de la configuration d'un équipement

On retrouve dans ce cas un transfert depuis le poste GNU/Linux vers l'équipement réseau. C'est le cas le plus simple du point de vue de la gestion des droits.

Une fois la nouvelle configuration éditée, on place le fichier correspondant dans le répertoire `/var/lib/tftp/` et on lance le transfert à partir de l'interface en ligne de commande de l'équipement.

```
sw1#copy tftp://192.168.2.1/new-config run
Destination filename [running-config]?
Accessing tftp://192.168.2.1/new-config...
Loading new-config from 192.168.2.1 (via Vlan2): !
[OK - 2756 bytes]

2756 bytes copied in 19.984 secs (138 bytes/sec)
sw1#
```

```

2d00h: %SYS-5-CONFIG_I: Configured from tftp://192.168.2.1/new-config by console
sw1#
<snip/>

sw1#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]

```

Il est préférable de copier la configuration en RAM dans un premier temps. Si la nouvelle configuration provoque un dysfonctionnement réseau, on a toujours la possibilité de réinitialiser complètement l'équipement. Il reprendra la configuration stockée en NVRAM (la version correcte précédente). Autrement, après avoir effectué les tests d'usage, on sauvegarde la nouvelle configuration en RAM dans la NVRAM.

4.3. Un soupçon de sécurité

Comme le service `tftpd` n'est pas un modèle en matière de sécurité, il est souhaitable de bien *encadrer* son utilisation. Généralement, on complète le contrôle d'accès (voir [Contrôle d'accès réseau au service](#)) par une règle de filtrage réseau pour chaque équipement.

Côté équipement on fixe l'adresse IP ou l'interface utilisée pour les transactions TFTP avec des instructions du type :

- Pour un commutateur :

```

interface Vlan2
  ip address 192.168.2.2 255.255.255.0
  no ip proxy-arp
  no ip route-cache
!
ip tftp source-interface Vlan2

```

- Pour un routeur :

```

interface Loopback0
  ip address 192.168.2.2 255.255.255.255
  no ip redirects
  no ip unreachable
  no ip directed-broadcast
  no ip proxy-arp
!
ip tftp source-interface Loopback0

```

Côté service `tftpd`, on n'autorise les accès TFTP (port 69/udp) qu'à partir des adresses IP fixées sur les équipements. Voici un extrait du fichier `/var/lib/iptables/active` utilisé par le script d'initialisation du paquet `iptables`.

```

*filter
:INPUT DROP [0:0]
<snip/>
-A INPUT -s 192.168.2.2 -p udp --dport 69 -m state --state NEW -j ACCEPT
<snip/>

```

Pour un exemple complet, voir [Annexe A, Configuration type du filtrage réseau](#).

5. Collecte des journaux

Les journaux, ou *logs* dans le jargon, servent à enregistrer tous les évènements qui surviennent sur un système. Historiquement, le service `syslog` a été développé pour la branche Unix des systèmes BSD. Depuis, ce service a été très largement adopté. On le retrouve sur tous les systèmes Unix, GNU/Linux et surtout sur les équipements réseau de nombreux constructeurs. Le protocole `syslog` est décrit dans le document [RFC3164 The BSD Syslog Protocol](#)¹⁴.

Dans le contexte de ce document, les messages `syslog` émis par les équipements réseau doivent être collectés par une machine avec un système GNU/Linux. Ce type de machine constitue un dépôt de référence avec horodatage des évènements survenus sur le système d'information. Du point de vue sécurité c'est un maillon essentiel du contrôle d'intégrité. Dans le cas où des équipements ont été compromis, il subsistera toujours des messages permettant de remonter à l'instant d'origine de l'attaque.

Une fois les messages collectés, il faut les traiter. La problématique du traitement des journaux est un sujet d'étude à part entière qui sort du cadre de ce document. La [Section 5.4, « Traitement des journaux »](#) donne juste quelques pistes.

5.1. Installation et configuration du service syslog

Comme indiqué dans la [Section 1.3, « Logiciels utilisés »](#), seuls les paquets de la distribution *Debian GNU/Linux* sont présentés ici. L'installation de `sysklogd` se résume donc à l'instruction suivante :

```
# apt-get install sysklogd
```

Par défaut, la configuration du service ne prévoit pas de recevoir des messages via le réseau. Il faut donc éditer le fichier `/etc/init.d/sysklogd` pour que le démon `syslogd` accepte les messages sur le port 514/udp. Voici une copie des 15 premières lignes du fichier `/etc/init.d/sysklogd` :

```
#!/bin/sh
# /etc/init.d/sysklogd: start the system log daemon.

PATH=/bin:/usr/bin:/sbin:/usr/sbin

pidfile=/var/run/syslogd.pid
binpath=/sbin/syslogd

test -x $binpath || exit 0

# Options for start/restart the daemons
# For remote UDP logging use SYSLOGD="-r"
#
SYSLOGD="-r"
```

On se retrouve ici dans une situation voisine de celle du service `tftpd` à une différence importante près. Il n'est pas possible d'utiliser le *tcpwrapper* du super-démon `inetd` pour mettre en place un contrôle d'accès. Il ne reste que le filtrage pour limiter les accès réseau au démon `syslogd`. Voir [Section 5.5, « Un soupçon de sécurité »](#).

Pour finir de configurer le service, il faut paramétrer la destination des messages reçus via le réseau. Cette opération se fait en ajoutant une ou plusieurs lignes au fichier `/etc/syslog.conf`. En reprenant l'exemple du commutateur 2950, voici un exemple :

```
local6.* /var/log/sw1.log
```

Ici, tous les messages du niveau de gravité 6 (*Informational*) seront placés dans le fichier `/var/log/sw1.log`. Le codage des niveaux de gravité dépend de l'équipement qui émet les messages de journalisation.

5.2. Exemple d'utilisation du service syslog

Une fois encore, c'est la syntaxe de l'IOS Cisco™ qui sert d'exemple de configuration. On ne traite qu'un seul exemple sachant que les commandes de configuration sont identiques entre routeurs et commutateurs. Voici un résumé des commandes permettant d'émettre des messages `syslog` sur le réseau.

¹⁴ <http://www.faqs.org/rfcs/rfc3164.html>

```
logging on
logging buffered 16384
service sequence-numbers
service timestamp log datetime msec localtime show-timezone
!
logging 192.168.2.1
!
logging facility local6
logging trap Informational
!
logging source-interface Vlan2
```

Une fois cette configuration implantée, le fichier de journalisation reçoit les messages émis par l'équipement. Voici un extrait consécutif à un chargement de fichier de configuration via le service TFTP :

```
Jan 25 13:18:45 192.168.2.2 146: 000143: 2w0d: %LINEPROTO-5-UPDOWN: \
  Line protocol on Interface FastEthernet0/24, changed state to up
Jan 25 14:06:31 192.168.2.2 156: 000153: Jan 25 14:06:30.534 CET: %SYS-5-CONFIG_I: \
  Configured from tftp://192.168.2.1/sw1-config by console
```

5.3. Utilisation de syslog-ng

Le service `syslog`, tel qu'il a été introduit, présente au moins deux défauts : la sécurité et la non discrimination des sources.

Le volet sécurité sort du cadre de ce document. On se contente de donc de contrôler simplement les sources de messages : voir [Section 5.5, « Un soupçon de sécurité »](#). Ensuite, on donne quelques pistes sur le contrôle d'intégrité des messages au point [Journalisation système](#) de la [Section 8, « Pour aller plus loin ! »](#).

La non discrimination des sources d'émission de messages de journalisation est beaucoup plus gênante du point de vue exploitation. En effet, la configuration présentée ci-dessus (voir [Section 5.1, « Installation et configuration du service syslog »](#)) a ouvert le port 514/udp en écoute et tous les messages reçus par ce canal sont rangés dans le fichier `/var/log/sw1.log`. Ce cas de figure convient très bien pour un équipement unique. Dans le cas où le nombre d'équipement augmente, ce point de stockage unique devient très vite difficile à gérer.

L'utilisation de `syslog-ng` permet de traiter cette difficulté simplement. Le démon `syslog-ng` se substitue complètement au démon `syslog` précédent. L'installation du paquet *Debian GNU/Linux* n'est pas différente des précédentes et le remplacement de `syslog` est automatique :

```
# apt-get install syslog-ng
```

La grande différence se situe au niveau du fichier de configuration du service : `/etc/syslog-ng/syslog-ng.conf`. Pour chaque catégorie de journalisation, on doit composer avec une définition de *source*, de *filtre* et de *destination*. Voici un exemple reprenant le cas du commutateur :

Définition d'une source

```
source net {
  # journalisation via eth2 -> commutateur sw1
  udp(ip(192.168.2.1));
};
```

Définition d'un filtre

```
filter f_sw1 {
  host(192.168.2.2) and level(info,notice,warn,crit,err);
};
```

Définition d'une destination

```
destination d_net_devices {
  file("/var/log/$HOST.log" owner("root") group("adm") perm(0640));
};
```

Utilisation des trois définitions

```
log {
    source(net);
    filter(f_sw1);
    destination(d_net_devices);
};
```

L'application de cette configuration entraîne la création d'un fichier `/var/log/192.168.2.2.log` qui reçoit tous les messages du commutateur `sw1` qui a l'adresse IP `192.168.2.2`. Le fichier de destination est créé avec un nom correspondant à l'adresse IP de l'équipement parce qu'aucun service DNS n'a été configuré. En exploitation réelle, on installe généralement un service DNS dédié au périmètre de gestion de l'infrastructure. Dans ce cas, les fichiers de journalisation portent le nom d'hôte de l'équipement.

L'ajout de tout nouvel équipement avec un autre nom et une autre adresse IP entraînera le création d'un nouveau fichier de journalisation. On pourra alors traiter les alertes séparément pour chaque équipement.

5.4. Traitement des journaux

La problématique du traitement des journaux bute sur «la motivation très limitée» des responsables d'exploitation. On entend trop souvent que la lecture des journaux est fastidieuse et inutile. Pourtant, on ne compte plus les exemples d'intrusions qui auraient pu être évitées facilement si les *logs* avaient été consultés régulièrement.

L'offre des outils de traitement de *logs* est très diverse. Voici deux propositions d'outils choisis avec un parti pris évident : imposer la lecture des journaux via le courrier électronique.

*Logwatch*¹⁵

logwatch émet un rapport toutes les 24h synthétisant les évènements par service. Dans le contexte de ce document, on active un service correspondant à tous les journaux émis par les équipements réseau. La grande force de *logwatch*, c'est la sommation des entrées répétitives qui optimise la taille du rapport.

*Logcheck*¹⁶

logcheck est un outil conçu à partir des paquets *Debian GNU/Linux*. Il émet un rapport toutes les heures en fonction de trois niveaux d'utilisation : poste de travail, serveur et «paranoïaque». Plus on avance vers la «paranoïa», plus le nombre de messages retenus est important. Pour chaque niveau d'utilisation, la synthèse des évènements comprend trois niveaux de priorité de traitement : alerte, sécurité et système. Dans le contexte de ce document, on ajoute les fichiers de journalisation des équipements réseau à la liste des fichiers traités par *logcheck*. Après la lecture de quelques rapports, on est capable d'éditer ses propres règles de sélection en s'inspirant des règles existantes pour les autres services du système. La grande majorité des opérations de sélection effectuées par *logcheck* sont pré-définies par des utilisateurs très expérimentés : les responsables des paquets. C'est un avantage considérable pour les débutants. On gagne ainsi un temps très important dans l'apprentissage du travail d'analyse.

Voici un extrait de rapport relatif à l'utilisation d'une règle de filtrage sur un routeur :

```
Security Events
=====
Jan 24 23:12:06 Router 20656: Jan 25 00:12:05.856 GMT: %SEC-6-IPACCESSLOGP: \
    list inbound denied udp aaa.aaa.aaa.aaa(53) -> ddd.ddd.ddd.ddd(33434), 3 packets
Jan 24 23:15:06 Router 20660: Jan 25 00:15:05.884 GMT: %SEC-6-IPACCESSLOGP: \
    list inbound denied udp bbb.bbb.bbb.bbb(53) -> ddd.ddd.ddd.ddd(33434), 2 packets
Jan 24 23:20:06 Router 20666: Jan 25 00:20:05.932 GMT: %SEC-6-IPACCESSLOGP: \
    list inbound denied udp bbb.bbb.bbb.bbb(53) -> ddd.ddd.ddd.ddd(33434), 1 packet
```

5.5. Un soupçon de sécurité

Comme `tftpd`, le service `syslogd` n'est pas un modèle en matière de sécurité. Il est donc souhaitable de bien *encadrer* son utilisation. On configure le contrôle d'accès avec une règle de filtrage réseau par équipement.

¹⁵ <http://www.logwatch.org/>

¹⁶ <http://logcheck.org/>

Voici un extrait du fichier `/var/lib/iptables/active` utilisé par le script d'initialisation du paquet `iptables`.

```
*filter
:INPUT DROP [0:0]
<snip/>
-A INPUT -s 192.168.2.2 -p udp --dport 514 -m state --state NEW -j ACCEPT
<snip/>
```

Pour un exemple complet, voir [Annexe A, Configuration type du filtrage réseau](#).

6. Synchronisation des horloges avec le protocole NTP

La mise en oeuvre du protocole NTP suppose que l'on ait besoin d'un horodatage précis des évènements qui surviennent sur les équipements réseau. Classiquement, le serveur du périmètre de gestion exécute un démon `ntpd` qui sert de référence à tous les équipements. Le protocole `ntp` est décrit dans le document [RFC1119 *Network Time Protocol \(version 2\) specification and implementation*](http://www.faqs.org/rfcs/rfc1119.html)¹⁷.

6.1. Installation et configuration du service ntp

Comme indiqué dans la [Section 1.3, « Logiciels utilisés »](#), seuls les paquets de la distribution *Debian GNU/Linux* sont présentés ici. L'installation du paquet `ntp` se résume donc à l'instruction suivante :

```
# apt-get install ntp
```

L'avantage de ce paquet, c'est qu'il ne nécessite aucune configuration. On peut donc passer côté équipement. Voici un exemple de configuration type commun aux routeurs et aux commutateurs :

```
!
ntp server 192.168.2.1 source Vlan2
clock timezone CET +1
clock summer-time EDT recurring
!
```

6.2. Validation de la configuration ntp

La syntaxe IOS de contrôle de l'état du service sur l'équipement est la suivante :

```
sw1#sh ntp status
Clock is synchronized, stratum 4, reference is 192.168.2.1
nominal freq is 250.0000 Hz, actual freq is 249.9962 Hz, precision is 2**18
reference time is C5A12468.AFE91280 (21:02:48.687 CET Tue Jan 25 2005)
clock offset is 0.0817 msec, root delay is 78.25 msec
root dispersion is 122.21 msec, peer dispersion is 0.03 msec

sw1#sh ntp associations

      address          ref clock      st when poll reach delay offset  disp
*~192.168.2.1         213.161.8.44   3   33  64 377   2.0  0.08  0.0
* master (syncd), # master (unsyncd), + selected, - candidate, ~ configured
```

L'utilitaire `ntpq` fournit les mêmes information pour le système GNU/Linux :

```
# ntpq
ntpq> associations

ind assID status  conf reach auth condition  last_event cnt
=====
  1 26588  9614   yes  yes  none  sys.peer  reachable  1
  2 26589  9014   yes  yes  none   reject  reachable  1
ntpq> peer
      remote          refid          st t when poll reach  delay  offset  jitter
=====
*cpe1-8-44.cable 164.8.100.2      2 u  865 1024 377   79.463 -11.913 33.217
LOCAL(0)         LOCAL(0)         13 l   39   64 377   0.000  0.000  0.004
```

6.3. Encore un soupçon de sécurité

Comme `tftpd` et `syslogd`, le service `ntpd` n'est pas un modèle en matière de sécurité. Il est donc souhaitable de bien *encadrer* son utilisation. On configure le contrôle d'accès avec une règle de filtrage réseau par équipement.

¹⁷ <http://www.faqs.org/rfcs/rfc1119.html>

Voici encore un extrait du fichier `/var/lib/iptables/active` utilisé par le script d'initialisation du paquet `iptables`.

```
*filter
:INPUT DROP [0:0]
<snip/>
-A INPUT -s 192.168.2.2 -p udp --dport 123 -m state --state NEW -j ACCEPT
<snip/>
```

Pour un exemple complet, voir [Annexe A, Configuration type du filtrage réseau](#).

7. Introduction à la métrologie avec mrtg

Il est impossible de prétendre faire de la gestion d'équipement réseau sans parler de métrologie. Qui dit métrologie, dit système de gestion intégré ou *Network Management System* (NMS). Ces systèmes intégrés sont trop souvent des « usines à gaz » au coût d'installation et d'administration prohibitif (on n'ose même plus parler du coût de licence !). L'optique de ce document étant de présenter des services simples à mettre en oeuvre, mrtg est l'outil GNU/Linux le plus répandu qui convient bien à cette présentation.

mrtg s'appuie sur le protocole SNMP (*Simple Network Management Protocol*) dont la première version est décrite dans le document [RFC1157 Simple Network Management Protocol \(SNMP\)](http://www.faqs.org/rfcs/rfc1157.html)¹⁸. Ce protocole permet à mrtg d'interroger la base d'informations (MIB ou *Management Information Base*) d'un équipement capable de traiter les messages SNMP. À part pour les équipements de très bas de gamme, tous les constructeurs d'équipements réseau fournissent des bases d'informations MIB pour leurs lignes de produits.

Le principe de fonctionnement est relativement simple : le programme mrtg est exécuté périodiquement pour interroger les bases d'informations MIB des équipements en fonction d'un fichier de configuration. Le résultat de ces interrogations sert à construire des images qui sont rangées dans un répertoire de pageWeb.

7.1. Installation et configuration du service mrtg

Au niveau de chaque équipement, le service `snmp` doit être activé et la chaîne de caractères de définition de communauté paramétrée. Voici un exemple de configuration IOS ; toujours pour un commutateur 2950 :

```
snmp-server trap-source Vlan2❶
snmp-server enable traps syslog
no snmp-server community public ro
!
no access-list 51
access-list 51 permit 192.168.2.1❷
snmp-server community m3tr010g13 ro 51❸
```

- ❶ Les messages SNMP émis par l'équipements ont une adresse IP source unique : celle de l'interface Vlan2.
- ❷ La liste de contrôle d'accès numéro 51 n'autorise que l'adresse IP 192.168.2.1 à interroger la base d'informations MIB.
- ❸ La définition de communauté m3tr010g13 n'a qu'un accès en lecture seule.

Côté système GNU/Linux, on a déjà indiqué dans la [Section 1.3, « Logiciels utilisés »](#), que seuls les paquets de la distribution *Debian GNU/Linux* sont présentés ici. L'installation de mrtg et du serveur web de visualisation apache se résume donc à l'instruction suivante :

```
# apt-get install mrtg apache
```

L'installation des paquets *Debian GNU/Linux* couvre toutes les opérations de configuration de base :

- Création du répertoire `/var/www/mrtg/` et copie des icônes de base des pages HTML produites par mrtg.
- Planification de l'exécution périodique de la commande `mrtg` via le service `cron` avec le fichier `/etc/cron.d/mrtg`.
- L'URL de consultation des résultats est : `http://192.168.2.1/mrtg/`.

7.2. Exemple d'utilisation sur un commutateur 2950

Pour chaque équipement supervisé, il faut créer ou compléter un fichier de configuration puis construire l'index de page web correspondant.

La configuration est obtenue à l'aide de la commande `cfgmaker`. Voici l'exemple de la création d'un nouveau fichier de configuration pour l'équipement paramétré ci-avant (Voir [Section 7.1, « Installation et configuration du service mrtg »](#)) :

```
# cfgmaker m3tr010g13@192.168.2.2 >mrtg.cfg
```

¹⁸ <http://www.faqs.org/rfcs/rfc1157.html>

```
# mv /etc/mrtg.cfg /etc/mrtg.cfg.dpkg-dist
# cp mrtg.cfg /etc
```

Une fois le fichier de configuration créé, il est possible de l'éditer pour l'adapter à ses propres besoins. La modification la plus classique consiste à renommer et/ou renuméroter les interfaces.

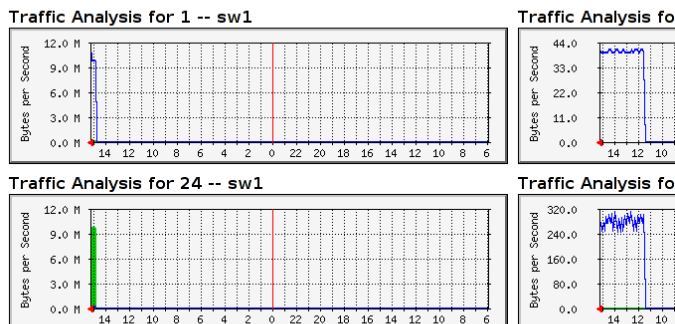
La construction de l'index se fait à l'aide de la commande **indexmaker**. Voici la suite du traitement de l'exemple du commutateur 2950 :

```
# indexmaker --output /var/www/mrtg/index.html /etc/mrtg.cfg
```

Il est aussi possible d'éditer ce fichier d'index pour l'adapter à ses besoins. Il est cependant préférable de faire appel aux options de la commande **indexmaker**. Ces options sont fournies dans les pages de manuels : **man indexmaker**.

Le résultat obtenu est de la forme suivante pour la page d'index principale :

MRTG Index Page



MRTG MULTI ROUTER TRAFFIC GRAPHER
 version 2.10.13
 Tobias Oetiker <oetiker@ee.ethz.ch>
 and Dave Rand <dlr@bungl.com>

Index mrtg - vue complète¹⁹

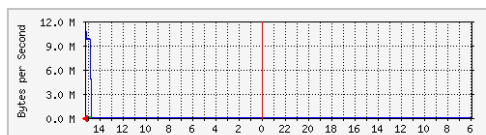
Pour une interface, le résultat se présente sous la forme suivante :

Traffic Analysis for 1 -- sw1

System: sw1 in
 Maintainer:
 Description: FastEthernet0/1 Management trunk
 ifType: ethernetCsmacd (6)
 ifName: Fa0/1
 Max Speed: 12.5 MBytes/s

The statistics were last updated **Friday, 28 January 2005 at 15:20**,
 at which time 'sw1' had been up for **17 days, 23:39:08**.

'Daily' Graph (5 Minute Average)



Max In:298.5 kB/s (2.4%) Average In:43.5 kB/s (0.3%) Current In:290.2 kB/s (2.3%)
 Max Out:11.6 MB/s (93.1%) Average Out:1545.7 kB/s (12.4%) Current Out:11.6 MB/s (93.1%)

Interface mrtg - vue complète²⁰

7.3. Encore un soupçon de sécurité

Comme tous les autres services présentés ci-avant, le protocole SNMP ne fait pas exception. Ce n'est pas un modèle en matière de sécurité. Il existe plusieurs versions de ce protocole : SNMPv1 de 1990, SNMPv2c de 1996 à 2002 et

¹⁹ <http://www.linux-france.org/prj/inetdoc/articles/devmgmt/images/index.mrtg.png>

²⁰ <http://www.linux-france.org/prj/inetdoc/articles/devmgmt/images/interface.mrtg.png>

SNMPv3 à partir de 2000. Plus ces versions ont évolué, plus les fonctions de sécurité se sont étoffées. Les équipements réseau actuels intègrent le plus souvent la version SNMPv2c qui est très limitée.

On se retrouve donc dans une situation identique à celle des autres services présentés. Il est souhaitable de bien *encadrer* l'utilisation du protocole. On configure le contrôle d'accès avec une règle de filtrage réseau par équipement.

Voici un extrait du fichier `/var/lib/iptables/active` utilisé par le script d'initialisation du paquet `iptables`.

```
*filter
:INPUT DROP [0:0]
<snip/>
-A INPUT -s 192.168.2.2 -p udp --dport 162 -m state --state NEW -j ACCEPT
<snip/>
```

Pour un exemple complet, voir [Annexe A, Configuration type du filtrage réseau](#).

8. Pour aller plus loin !

Les outils GNU/Linux permettent de structurer la centralisation de la gestion des équipements réseau. La force du logiciel libre c'est la capacité à adapter son infrastructure au plus près de ses besoins. Cette «capacité d'adaptation» a un coût non négligeable.

En phase de conception ou d'évolution d'un système d'information, il faut étudier attentivement le rapport contraintes d'exploitation sur coût d'acquisition.

Il est fréquent de constater qu'au delà du coût d'acquisition d'une solution propriétaire, les fonctionnalités offertes sont très|trop nombreuses et ne répondent pas nécessairement aux besoins. À l'inverse, la mise en oeuvre des fonctionnalités décrites dans ce document permet d'atteindre rapidement et simplement des résultats faciles à exploiter.

Pour étudier plus en détails les modalités d'exploitation des équipements réseau voici quelques références documentaires :

Guides de la NSA

De plus en plus souvent, les conseils sur les «bonnes pratiques» de gestion des équipements sont incluses dans les document relatifs à la sécurité. Les guides de la NSA sont les exemples emblématiques de cette tendance. Même si ces guides ne traitent que des équipements Cisco™, il ne devrait pas être trop difficile de les adapter aux interfaces|langages de configuration des autres marques.

Cisco IOS Switch Security Configuration Guide

Le guide *Swicth guide version 1.01*²¹ est dédié aux commutateurs. Il présente bien sûr beaucoup plus de fonctions que ce document.

Router Security Configuration Guide

Le guide *Router guide version 1.1b*²² est une excellente référence sur la sécurisation d'une architecture d'interconnexion réseau. Il va bien au delà de la configuration des équipements.

Executive Summary

La feuille *Executive Summary*²³ est une *checklist* définitivement indispensable pour configurer les équipements.

Journalisation système

On reproche beaucoup au service `syslog` son manque de sécurité intrinsèque. C'est oublier un peu vite que pour que ce type de service soit largement adopté, il doit être «léger» et facile à implanter dans un espace mémoire réduit. La sécurisation des échanges de messages `syslog` est relativement facile à obtenir en «canalisant» ces messages hors de la bande passante utilisateur dans des VLANs ou des réseaux privés virtuels spécifiques.

Voici quelques références permettant d'argumenter sur cette question :

Groupe de travail Syslog de l'IETF

Page Web du groupe de travail : *IETF Syslog Working Group Home Page*²⁴.

Présentation HSC

Si cette présentation sur *Normes utiles en sécurité réseau*²⁵ ne traite pas directement de la journaliation, elle met en évidence son importance en tant qu'outil d'audit et de veille.

Voilà ! on dispose maintenant d'une batterie de fonctions et d'outils de gestion des équipements réseau. Il ne reste plus qu'à composer son système de gestion personnel. Si vous avez déjà eu à administrer un système lourd et coûteux vous réaliserez plus facilement et plus vite combien cette facilité d'adaptation est avantageuse.

²¹ http://www.nsa.gov/snac/os/switch-guide-version1_01.pdf

²² http://www.nsa.gov/snac/routers/cisco_scg-1.1b.pdf

²³ http://www.nsa.gov/snac/routers/cisco_exec_sum.pdf

²⁴ <http://www.employees.org/~lonvick/>

²⁵ <http://www.hsc.fr/ressources/presentations/dnac03/>

A. Configuration type du filtrage réseau

Après avoir donné plusieurs exemples de règles de filtrage pour les services de gestion d'équipements réseau présentés, voici un fichier `/var/lib/iptables/active/` complet. Même si le filtrage réseau sort du cadre de ce document, cette configuration essaie de respecter au mieux la règle d'or : *décrire le plus précisément possible le premier paquet d'une nouvelle communication et faire confiance au suivi d'état ensuite.*



Attention !

Le fichier ci-dessous est fourni à titre d'exemple sans aucune garantie de quelque nature que ce soit.

Pour activer l'ensemble des règles il faut saisir une commande du type : `iptables-restore </var/lib/iptables/active.`

```
# Configuration type du filtrage réseau
# pour un serveur de gestion d'équipement réseau
#
#~~~~~
# Définition des interfaces
#~~~~~
#
# . eth0 : interface «réseau public»
# . eth2 : interface «périmètre gestion»
#
#~~~~~
# Tables de traduction d'adresses
#~~~~~
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
#~~~~~
# Tables de filtrage
#~~~~~
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
#
# -> Chaîne INPUT
# . toutes les communications internes sont autorisées
-A INPUT -i lo -j ACCEPT
# . suivi de communication
-A INPUT -p udp -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m state --state ESTABLISHED -m tcp ! --syn -j ACCEPT
-A INPUT -p tcp -m state --state RELATED -m tcp --syn -j ACCEPT
-A INPUT -p icmp -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable \
    -m state --state RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -m state --state RELATED -j ACCEPT
# . administration du serveur avec SSH
-A INPUT -i eth0 -p tcp --syn --dport 22 -m state --state NEW -j ACCEPT
# . services de gestion d'équipements réseau
# . exemple d'un commutateur 2950 avec l'@ IP 192.168.2.2
-A INPUT -i eth2 -s 192.168.2.2 -p udp --dport 69 -m state --state NEW -j ACCEPT
-A INPUT -i eth2 -s 192.168.2.2 -p udp --dport 514 -m state --state NEW -j ACCEPT
-A INPUT -i eth2 -s 192.168.2.2 -p udp --dport 123 -m state --state NEW -j ACCEPT
-A INPUT -i eth2 -s 192.168.2.2 -p udp --dport 162 -m state --state NEW -j ACCEPT
```

```

# . poubelle propre
-A INPUT -m state --state INVALID -j DROP
-A INPUT -p tcp -j REJECT --reject-with tcp-reset
-A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable
#
# -> Chaîne FORWARD
# . suivi de communication
-A FORWARD -p udp -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p tcp -m state --state ESTABLISHED -m tcp ! --syn -j ACCEPT
-A FORWARD -p tcp -m state --state RELATED -m tcp --syn -j ACCEPT
-A FORWARD -p icmp -m state --state ESTABLISHED -j ACCEPT
-A FORWARD -p icmp --icmp-type destination-unreachable \
    -m state --state RELATED -j ACCEPT
-A FORWARD -p icmp --icmp-type time-exceeded -m state --state RELATED -j ACCEPT
# . poubelle propre
-A FORWARD -m state --state INVALID -j DROP
-A FORWARD -p tcp -j REJECT --reject-with tcp-reset
-A FORWARD -p udp -j REJECT --reject-with icmp-port-unreachable
COMMIT

```

Il est possible de simplifier la syntaxe de filtrage des communications par équipement en groupant les services ouverts. En prenant l'exemple ci-dessous on perd en lisibilité puisque le compteur d'utilisation de la règle ne permet pas de distinguer le service utilisé dans la liste.

```

-A INPUT -i eth2 -s 192.168.2.2 -p udp -m multiport --dports 69,123,162,514 \
    -m state --state NEW -j ACCEPT

```

Dans le but de mieux respecter la règle d'or du filtrage, il est possible de tester les adresses MAC source des équipements gérés. De cette façon, on limite les possibilités d'*arp spoofing*.