LPIC-1

Linux Professional Institue Certification
(Study Guide)

June 13, 2024

Assessment test (xxxvii)

inetd (de l'anglais internet service daemon) est un démon Unix qui permet de gérer les connexions à des services réseau.

Au démarrage, inetd écoute un ensemble de ports configurés. Quand une demande de connexion TCP ou un datagramme UDP est reçue, inetd lance l'application configurée pour ce port.

inetd est utilisé pour minimiser le nombre de processus qui correspondent à des démons peu fréquemment utilisés et économise des ressources par rapport à l'alternative qui consiste à lancer un démon indépendant pour chaque service. Il introduit cependant une latence due au démarrage du processus serveur.

Pour activer un service inetd il suffit d'ajouter une ligne au fichiers /etc/inetd.conf dont la forme générale est:

service type-socket protocole propriétaire chemin-absolu argument

Par exemple pour activer le serveur ftpd (partage de fichiers via FTP) :

ftpd stream tcp nobody /usr/bind/ftpd ftpd -s

Le fichier **/etc/inetd.conf** est le fichier de configuration par défaut du démon inetd . Ce fichier vous aide à spécifier les démons à démarrer par défaut et à fournir les arguments correspondant au style de fonctionnement requis pour chaque démon. Ce fichier fait partie de TCP/IP dans Network Support Facilities.

Si vous modifiez le fichier /etc/inetd.conf , exécutez la commande refresh -s inetd ou kill -1 InetdPID pour informer le démon inetd des modifications apportées à son fichier de configuration. Le fichier inetd.conf indique les démons qui démarrent par défaut et fournit des arguments déterminant le style de fonctionnement de chaque démon.

Le démon inetd contrôle les démons suivants: comsat, ftpd, telnetd, rshd, rlogind, rexecd, fingerd, tftpd, talkd, Uucpd.

La fonction **Dynamic Host Configuration Protocol (DHCP)** est un protocole client/serveur qui fournit automatiquement une adresse Internet Protocol (IP) et d'autres informations de configuration pertinentes à un hôte IP (par exemple, masque de sous-réseau et passerelle par défaut). Les RFC 2131 et 2132 définissent DHCP comme une norme IETF (Internet Engineering Task Force) basée sur bootstrap Protocol (BOOTP), un protocole avec lequel DHCP partage

de nombreux détails d'implémentation. DHCP permet aux hôtes d'obtenir les informations de configuration TCP/IP requises à partir d'un serveur DHCP.

Windows Server 2016 inclut Serveur DHCP, qui est un rôle de serveur réseau facultatif que vous pouvez déployer sur votre réseau pour louer des adresses IP et d'autres informations sur des clients DHCP. Tous les systèmes d'exploitation clients Windows incluent le client DHCP dans le cadre de TCP/IP, et le client DHCP est activé par défaut.

Le protocole **NBNS (pour NetBIOS Name Server)** fait partie de la famille des protocoles NetBios over TCP/IP (NBT). Il est implémenté pour le service WINS (Windows Internet Name Server) dans les systèmes Windows.

Une vulnérabilité y a été décelée. Elle permet à un individu mal intentionné d'empêcher à un serveur WINS de répondre à des requêtes provenant d'une machine possédant un nom donné. Description

Par conception, le protocole NBNS entre en jeu dans la résolution des conflits de noms sous Windows. C'est aussi, par conception, un protocole sans authentification il est donc sujet à l'usurpation d'identité (spoofing).

Un individu mal intentionné peut, par manipulations des mécanismes de Conflit de Nom (Name Conflict) et Liberation de Nom (Name Release), faire interpréter par une autre machine que son nom est en conflit. Selon le cas, celà peut avoir deux conséquences différentes : la machine ne peut plus enregistrer son nom sur le réseau, ou la machine libèrera son nom, alors qu'elle avait déjà été enregistrée sous ce nom sur le réseau. Dans les deux cas, elle ne répondra plus à des renquêtes envoyées à ce nom.

1 Exploring Linux Command-Line Tools

1.1 Understanding Command-Line Basics

Using Environment Variables An **environment variable** is a user-definable value that can affect the way running processes will behave on a computer. Environment variables are part of the environment in which a process runs. For example, a running process can query the value of the TEMP environment variable to discover a suitable location to store temporary files, or the HOME or USERPROFILE variable to find the directory structure owned by the user running the process.

In all Unix and Unix-like systems, as well as on Windows, each process has its own separate set of environment variables. By default, when a process is created, it inherits a duplicate run-time environment of its parent process, except for explicit changes made by the parent when it creates the child. At the API level, these changes must be done between running fork and exec. Alternatively, from command shells such as bash, a user can change environment variables for a particular command invocation by indirectly invoking it via env or using the notation. A running program can access the values of environment variables for configuration purposes.

Shell scripts and batch files use environment variables to communicate data and preferences to child processes. They can also be used to store temporary values for reference later in a shell script. However, in Unix, non-exported

variables are preferred for this as they do not leak outside the process.

Linux Commands

```
#!/bin/bash
echo the $# parameter destroys listings syntax highlighting
```

Command to ...

```
traceroute
```

Command to clear history of commands (shell).

```
history -c
```

Command to print the route packets trace to network host.

```
traceroute
```

Command to run a System V init script.

```
service --status-all
```