

Réveiller un ordinateur par le réseau - Wake On Lan (WOL)

Le réveil est déclenché quand la carte Ethernet de l'ordinateur reçoit un **paquet magique** qui est une trame de données Ethernet contenant les octets **FF FF FF FF FF FF** suivis de **seize répétitions de l'adresse MAC** de la cible, puis d'un mot de passe (si nécessaire) de quatre ou six octets.

Le paquet magique est une trame réseau transmise sur le **port 0** (historiquement le port le plus communément utilisé), 7 ou 9 (devenant les ports les plus utilisés) en mode non-connecté donc en **UDP**.

Le module python à utiliser. Enregistrez ce script dans un fichier que vous appellerez **wol.py**.

Attention la chaîne de caractères **adressebroadcastreseau** est à remplacer par l'adresse de broadcast du réseau.

```
#!/usr/bin/env python
# -*-coding:Latin-1 -*-
# wol.py
#
# This module is from ActiveState Code Recipes:
# http://code.activestate.com/recipes/358449-wake-on-lan/
# and patched for Python 3 with:
# http://code.activestate.com/recipes/577609-wake-on-lan-for-python-3/
#
# Example:
# import wol
# wol.wake_on_lan('70:F3:95:15:00:B5')
#
import socket
import struct

def wake_on_lan(macAdresse):
    """ Mettre en route des ordinateurs en utilisant WOL. """

    # Analyser l'adresse MAC et tenter de la corriger
    if len(macAdresse) == 12:
        pass
    elif len(macAdresse) == 12 + 5:
        # extraire le séparateur (en général : ou - )
        separateur = macAdresse[2]
        macAdresse = macAdresse.replace(separateur, '')
    else:
        raise ValueError('Format de l\'adresse MAC incorrect')

    # Préparer le paquet magique
    data = b'FFFFFFFFFFFF' + (macAdresse * 20).encode()
    send_data = b''

    # Split up the hex values in pack
    for i in range(0, len(data), 2):
        # struct.pack permet de créer une chaîne en byte
        send_data += struct.pack('B', int(data[i: i + 2], 16))

    # Envoi en Broadcast sur le LAN
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
    sock.sendto(send_data, ('adresse_broadcast_reseau', 7))
```

N'hésitez pas à **documenter** votre module en lui donnant une chaîne de documentation (doc string). Pour cela utilisez les **tripes guillemets** qui indiquent une chaîne multi-lignes. Tout ce qu'il y a entre l'ouverture et la fermeture des guillemets fait partie de la chaîne, y compris les retours chariot et les autres guillemets. Cela est souvent utilisé pour définir une doc string mais on peut les utiliser partout.

Pour utiliser ce module, il suffit de créer un programme python qui **importe le module wol**. Le fichier **wol.py** doit être dans le même dossier que le programme qui l'utilise.

```
import wol
wol.wake_on_lan('00:11:32:1B:B3:5F')
print("paquet magique envoyé")
```

From:

/ - **Les cours du BTS SIO**

Permanent link:

[/doku.php/dev/python/wol](#)

Last update: **2014/11/09 22:20**

