Windows Passwörter

Es gibt zwei Hash-Algorithmen unter Windows: das veraltete LM und das neuere NTLM bzw. NTLMv2. Da LM die Sicherheit eines Systems aushebeln kann, sollte es unbedingt deaktiviert werden. Hierfür muss folgender Registry Schlüssel erzeugt werden:

```
HKLM\System\CurrentControlSet\Control\LSA
NoLMHash  REG_DRWORD  1
```

Die Passwörter werden in der SAM Datenbank gespeichert:

```
C:\Windows\System32\config\SAM
oder
HKLM\SAM
```

Diese Datenbank ist im laufenden System nicht lesbar, man muss also offline auf die Platte zugreifen oder die Werte aus dem RAM auslesen. Letzteres macht beispielsweise PGDump.

Total Commander FTP Passwörter

Folgendes Python Skript entschlüsselt die gespeicherten FTP Passwörter, die z.B. in `%APPDATA%\GHISLER\wcx_ftp.ini` gespeichert sind:

```
import os
import optparse

def process_file(filename):
    try:
        print("-> Trying: " + filename)
        f = open(filename,"r")
        print("-> Found: " + filename)
        print("-> Decrypting: " + filename)
        print("")
        for line in f:
            if ("password" in line.strip()):
                print("password=" + tc_decrypt(line.strip().split("=")[1]))
            else: print(line.strip())
        f.close()
        print("")
    except IOError:
        print("-> Not found: " + filename)
        print("")

def search_ini():
```

```
    """
    Search the wcx_ftp.ini file in common places
    """
    folder = []
    folder.append(os.getenv('APPDATA') + "\\GHISLER\\wcx_ftp.ini")
    folder.append(os.getenv('SYSTEMROOT') + "\\wcx_ftp.ini")
    folder.append("wcx_ftp.ini")
    for ini in folder:
        process_file(ini)

def tc_random(nMax):
    global RANDOM_BASE
    RANDOM_BASE = ((RANDOM_BASE * 0x8088405) & 0xffffffff) + 1
    return (((RANDOM_BASE * nMax) >> 32)& 0xffffffff)

def tc_shift(n1, n2):
    return (((n1 << n2) & 0xffffffff) | ((n1 >> (8 - n2)) & 0xffffffff)) &
0xff

def tc_decrypt(pwd):
    global RANDOM_BASE
    password=[]
    for i in range(int(len(pwd)/2) - 4): #skip last 8 characters (4 * 2
bytes)
        password.append(int(pwd[2*i:2*(i+1)],16))
    pwlen = len(password)

    RANDOM_BASE = 849521

    for i in range(pwlen):
      password[i] = tc_shift(password[i], tc_random(8))

    RANDOM_BASE = 12345
    for i in range(256):
      a=tc_random(pwlen)
      b=tc_random(pwlen)
      password[a],password[b] = password[b],password[a]

    RANDOM_BASE = 42340
    for i in range(pwlen):
      password[i] = (password[i] ^ tc_random(256)) & 0xff

    RANDOM_BASE = 54321
    for i in range(pwlen):
      password[i] =  (password[i] - tc_random(256)) & 0xff

    for i in range(pwlen):
        password[i] = chr(password[i])

    return "".join(password)
```

```
def main():
    usage = "Usage: %prog [options]"
    parser = optparse.OptionParser(usage=usage)

    parser.add_option('-c', '--common', action='store_true', dest='common',
                      default=False, help='Search wcx_ftp.ini in common
places')
    parser.add_option('-f', '--file', action='store', dest="file",
default='',
                      help='File to decrypt')
    parser.add_option('-p', '--password', action='store', dest='password',
                      default='', help='Password to decrypt')
    options, args = parser.parse_args()
    if options.common:
        search_ini()
    if (options.file != ""):
        process_file(options.file)
    if (options.password != ""):
        pw = tc_decrypt(options.password)
        print("Decrypted password: " + pw)
    if (options.file == "" and options.password == "" and not
options.common):
        print("Nothing specified, run \"tcpwrecovery -h\" for options")

if __name__ == '__main__':
    main()
```