Security Vulnerability Notice

SE-2011-01

[Security weaknesses in a digital satellite TV platform, Issue 33]

DISCLAIMER

INFORMATION PROVIDED IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW NEITHER SECURITY EXPLORATIONS, ITS LICENSORS OR AFFILIATES, NOR THE COPYRIGHT HOLDERS MAKE ANY REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR THAT THE INFORMATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS, OR OTHER RIGHTS. THERE IS NO WARRANTY BY SECURITY EXPLORATIONS OR BY ANY OTHER PARTY THAT THE INFORMATION CONTAINED IN THE THIS DOCUMENT WILL MEET YOUR REQUIREMENTS OR THAT IT WILL BE ERROR-FREE.

YOU ASSUME ALL RESPONSIBILITY AND RISK FOR THE SELECTION AND USE OF THE INFORMATION TO ACHIEVE YOUR INTENDED RESULTS AND FOR THE INSTALLATION, USE, AND RESULTS OBTAINED FROM IT.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL SECURITY EXPLORATIONS, ITS EMPLOYEES OR LICENSORS OR AFFILIATES BE LIABLE FOR ANY LOST PROFITS, REVENUE, SALES, DATA, OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, PROPERTY DAMAGE, PERSONAL INJURY, INTERRUPTION OF BUSINESS, LOSS OF BUSINESS INFORMATION, OR FOR ANY SPECIAL, DIRECT, INDIRECT, INCIDENTAL, ECONOMIC, COVER, PUNITIVE, SPECIAL, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND WHETHER ARISING UNDER CONTRACT, TORT, NEGLIGENCE, OR OTHER THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF SECURITY EXPLORATIONS OR ITS LICENSORS OR AFFILIATES ARE ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS.

Security Explorations discovered a weakness in the implementation of a Multiroom Standard HD service [1] used by a Polish digital satellite provider NC+ [2]. As a result, the service could be abused by rogue subscribers to gain access to premium TV channels at a discounted rate. This report contains brief information about the vulnerability and the analysis that has lead to its discovery.

INTRODUCTION

Multiroom is a service offered by NC+ digital satellite TV provider that makes it possible to watch subscribed TV channels on additional set-top-box (STB) devices located in a subscriber's home network (i.e. STB devices in other rooms of a subscriber's home).

The primary benefit for a customer is a non-linear cost of having additional set-top-box devices - their cost is a fraction of the cost of a primary subscription associated with the main set-top-box device.

The primary risk for the operator is that client devices assigned to the main (server) set-top-box device might not be necessarily used in a given home network installation. A possibility to use client devices in different homes (by different parties and not necessarily subscribers) decreases operator's income (instead of N main subscriptions corresponding to the number of used devices, an operator bills a fixed fee for each additional STB device, which is a fraction of the main subscription - 5% in the case of the most expensive premium package available in NC+).

There are two variants of the Multiroom offer available in Platform NC+:

1. Multiroom Premium option making it possible for client devices to make use of the signal and recording functionality of the main device, this options seems to be not available to new subscribers any more,
2. Multiroom Standard HD, which requires both server and client devices to be provided with a signal source (SAT TV signal).

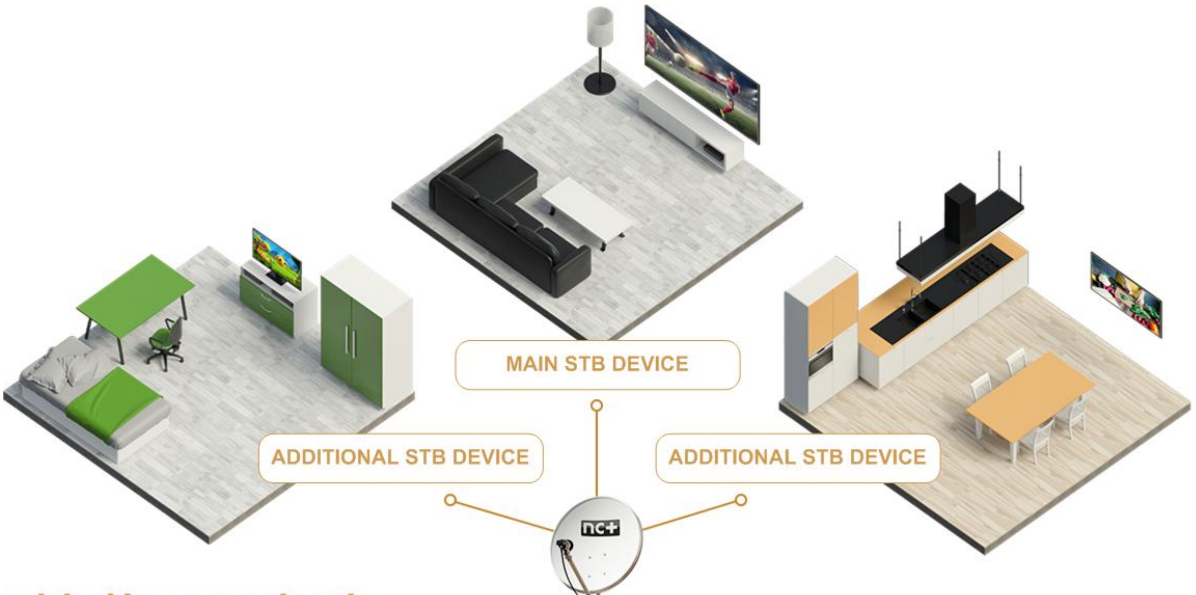
The analysis below is limited to Multiroom Standard HD as it seems to be the primary and only Multiroom offering currently available to NC+ subscribers. We further refer to it as Multiroom throughout this document.

MULTIROOM OPERATION

NC+ Multiroom installation is comprised of a main STB device and a set of a maximum 5 client devices (2-6 set-top-box devices / screens in total per home installation). This is illustrated on Fig. 1.

Multiroom service activation is required for both server and client devices and it proceeds in the following way:

- a main (server) set-top-box device receives a message from the operator including information about authorized client devices (their smart card and chip id numbers). The message is received over a private MPEG transport stream of a STB manufacturer (the so called `AdbEmmCarousel`, dvb locator `dvb://13e.514.3ad4` and MPEG PID 0x641),
- each client set-top-box device receives a configuration message over `AdbEmmCarousel` that assigns a client device to Group ID 102. This configuration setting puts a client device into a Multiroom Standard mode (reboot is required for it to take effect).

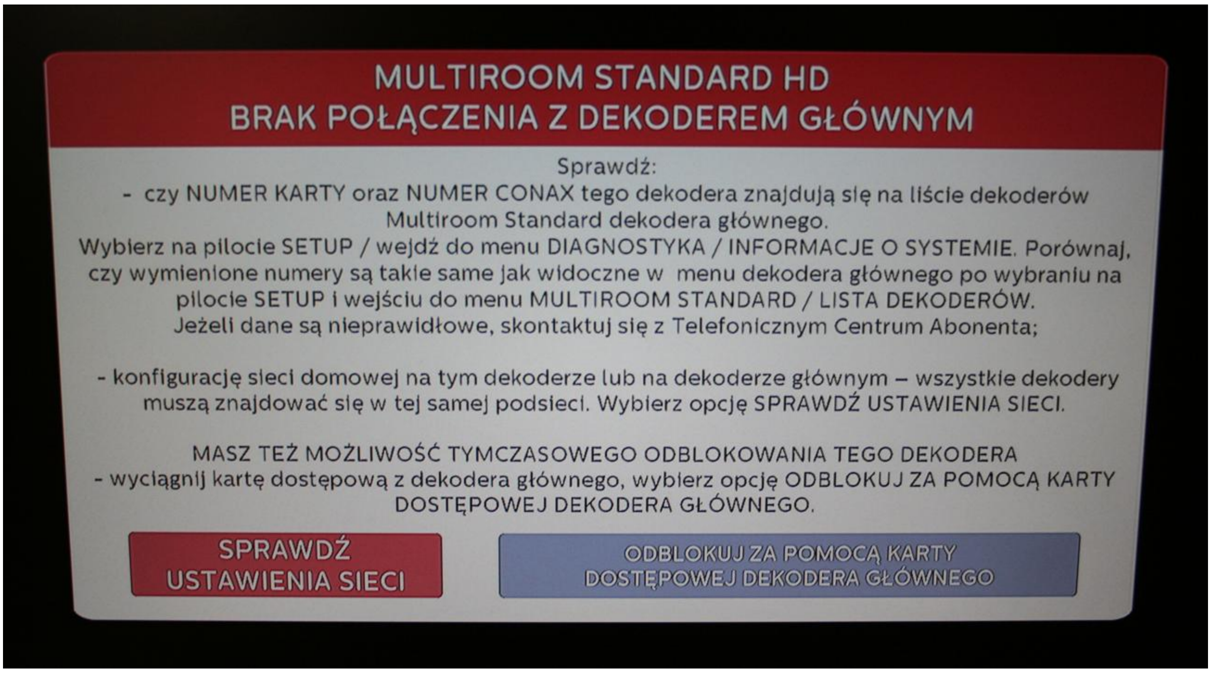


original image: ncplus.pl

Fig. 1 Schema of a sample Multiroom installation.

Upon successful activation of a Multiroom service, client devices that are part of Group ID 102 periodically communicate with a server device of a given home network by the means of HTTP GET requests.

The goal of this communication is to verify security of client devices (whether the authorized devices are connected to the main STB device). In case of an error (missing server device or authorization failure), a client device cannot be used (an error message is presented on a TV screen and no channels can be viewed). This is illustrated on Fig. 2.



MULTIROOM STANDARD HD
BRAK POŁĄCZENIA Z DEKODEREM GŁÓWNYM

Sprawdź:
- czy NUMER KARTY oraz NUMER CONAX tego dekodera znajdują się na liście dekoderów Multiroom Standard dekodera głównego.
Wybierz na pilocie SETUP / wejdź do menu DIAGNOSTYKA / INFORMACJE O SYSTEMIE. Porównaj, czy wymienione numery są takie same jak widoczne w menu dekodera głównego po wybraniu na pilocie SETUP i wejściu do menu MULTIROOM STANDARD / LISTA DEKODERÓW.
Jeżeli dane są nieprawidłowe, skontaktuj się z Telefonicznym Centrum Abonenta;

- konfigurację sieci domowej na tym dekoderze lub na dekoderze głównym – wszystkie dekodery muszą znajdować się w tej samej podsieci. Wybierz opcję SPRAWDŹ USTAWIENIA SIECI.

MASZ TEŻ MOŻLIWOŚĆ TYMCZASOWEGO ODBLOKOWANIA TEGO DEKODERA
- wyciągnij kartę dostępową z dekodera głównego, wybierz opcję ODBLOKUJ ZA POMOCĄ KARTY DOSTĘPOWEJ DEKODERA GŁÓWNEGO.

Fig. 2 Multiroom error message indicating a connection / authorization error.

As part of the verification process, `/sw_upgrades/apollo/multibox` file is fetched from a HTTP server running on port 8080 of a Multiroom server STB. This file contains encrypted authorization data for a client device (or their group) associated with a server device of a given home network.

THE WEAKNESS

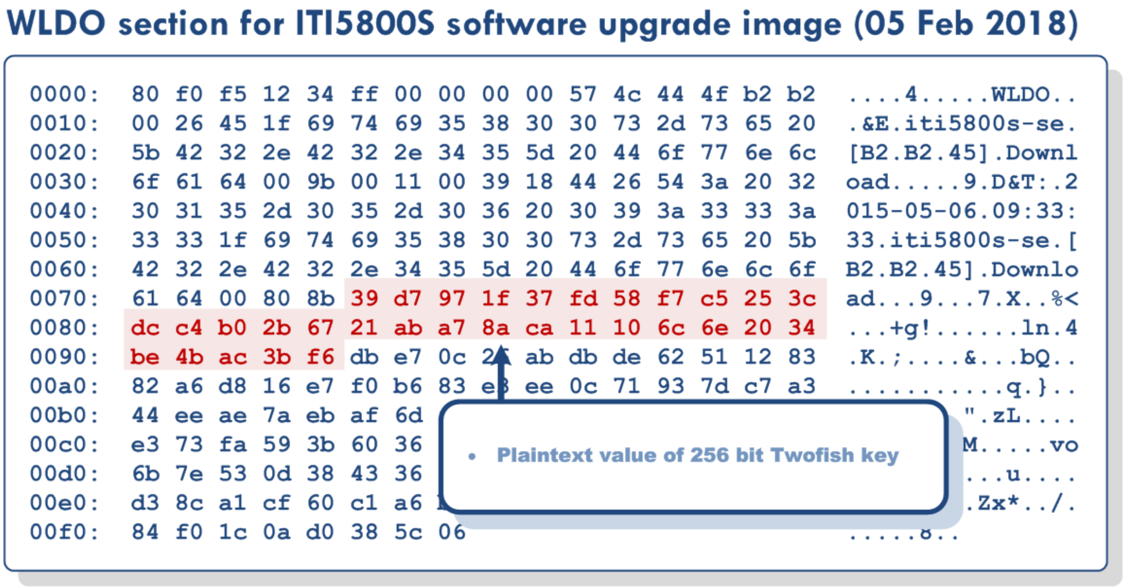
Among client devices supported for a Multiroom service, there is an ITI-5800S [3] set-top-box device from Advanced Digital Broadcast (ADB) company [4] (Fig. 3).

ITI5800S

- HD decoder
- STi7100 processor
- Hermes software
- Serial# family BXZB

Fig. 3 ITI-5800S set-top-box-device.

In 2012, Security Explorations published information about a weakness in the implementation of the system software upgrade mechanism of ITI-5800S and ITI5800SX devices [5][6]. For these devices, the upgrade image sent as part of dedicated MPEG streams (indicated by the SSU data) was encrypted with the use of a tweaked Twofish algorithm and a 128-bit decryption key. The problem with the implementation of the upgrade mechanism implemented for ITI5800S and ITI5800SX devices stemmed from the fact that the decryption key was sent in plaintext among the data for the upgrade image. This key was carried inside a WLDO section and could be successfully used to decrypt the encrypted upgrade image data sent to ITI5800S and ITI5800SX devices (Fig. 4).



WLDO section for ITI5800S software upgrade image (05 Feb 2018)

```
0000:  80 f0 f5 12 34 ff 00 00 00 00 57 4c 44 4f b2 b2  ....4.....WLDO..
0010:  00 26 45 1f 69 74 69 35 38 30 30 73 2d 73 65 20  .&E.iti5800s-se.
0020:  5b 42 32 2e 42 32 2e 34 35 5d 20 44 6f 77 6e 6c  [B2.B2.45].Downl
0030:  6f 61 64 00 9b 00 11 00 39 18 44 26 54 3a 20 32  oad.....9.D&T:.2
0040:  30 31 35 2d 30 35 2d 30 36 20 30 39 3a 33 33 3a  015-05-06.09:33:
0050:  33 33 1f 69 74 69 35 38 30 30 73 2d 73 65 20 5b  33.iti5800s-se.[
0060:  42 32 2e 42 32 2e 34 35 5d 20 44 6f 77 6e 6c 6f  B2.B2.45].Downlo
0070:  61 64 00 80 8b 39 d7 97 1f 37 fd 58 f7 c5 25 3c  ad...9...7.X..%<
0080:  dc c4 b0 2b 67 21 ab a7 8a ca 11 10 6c 6e 20 34  ...+g!......ln.4
0090:  be 4b ac 3b f6 db e7 0c 2  ab db de 62 51 12 83  .K.;....&...bQ..
00a0:  82 a6 d8 16 e7 f0 b6 83 e  ee 0c 71 93 7d c7 a3  ..........q.}..
00b0:  44 ee ae 7a eb af 6d                               ".zL....
00c0:  e3 73 fa 59 3b 60 36                               M.....vo
00d0:  6b 7e 53 0d 38 43 36                               ...u....
00e0:  d3 8c a1 cf 60 c1 a6                               .Zx*../.
00f0:  84 f0 1c 0a d0 38 5c 06                            .....8..
```

- Plaintext value of 256 bit Twofish key

Fig. 4 Location of a plaintext software upgrade key in WLDO section.

We verified that this was still the case nearly 6 years following the public disclosure:

```
# NBOX HDTV client for ITI 5800S, ITI 5800SX, ITI 2850ST, ITI2849ST
# (c) SECURITY EXPLORATIONS    2011 poland

box> go 1
box> root
uid=0(root) gid=0(root)
box> play dvb://13e.514.3ad4
box> ssuinfo
SSU SVID:   0x3aca     PID:   041a
[UPGRADE 00]
- pid                      0x0bbd
- oui                      0x000391(Advanced Digital Broadcast)
- customer_id              0x45
- hardware version         0xb2b0  ITI5800S   (BSKA serial)
- ssu_table_id             0x0080
- ssu_unique_download_id   0x1234
[UPGRADE 01]
- pid                      0x0bbe
- oui                      0x000391(Advanced Digital Broadcast)
- customer_id              0x45
- hardware version         0xb2b1  ITI5800SX  (BSLA serial)
- ssu_table_id             0x0080
- ssu_unique_download_id   0x1234
[UPGRADE 02]
- pid                      0x0bbf
- oui                      0x000391(Advanced Digital Broadcast)
- customer_id              0x45
- hardware version         0xb2b2  ITI5800S   (BXZB serial)
- ssu_table_id             0x0080
- ssu_unique_download_id   0x1234
...
box> upgdnl 0xb2b2
- image information
  name:             iti5800s-se [B2.B2.45] Download
  date:             D&T: 2015-05-06 09:33:33
getting UPGRADE_FILE                             (    12710 blocks) [###########
- processing image
  total size:       0x0132b240
- decrypting image
  algorithm:        tweaked Twofish ECB 256bit
  key:
 size: 00000020
       0000:  39 d7 97 1f 37 fd 58 f7 c5 25 3c dc c4 b0 2b 67  9...7.X..%<...+g
       0010:  21 ab a7 8a ca 11 10 6c 6e 20 34 be 4b ac 3b f6  !......ln.4.K.;.
- saving image
  output:           upgrade.dat
```

It's worth to mention that an MPEG stream containing SSU image for NC+ devices is not broadcasted in an encrypted form (there is no need to decrypt MPEG sections with the use of Control Words[1]). As such, these images are available to any 3rd party (non-NC+ subscribers in particular).

As a result of the above, we were able to a obtain plaintext Compressed ROMFS image[2] containing root filesystem for ITI-5800S device. This image could be successfully unpacked / extracted under Linux OS:

```
root@ncplus:~# ls -la /mnt/USB/u.cramfs
-rwxr-xr-x 1 root root 19182144 2018-02-05 22:21 /mnt/USB/u.cramfs
```

[1] we verified this by 1) removing a Conax card from STB and 2) enforcing a wrong chipset pairing key, to make it impossible to decrypt any scrambled MPEG sections - in both cases, the upgrade image could be successfully downloaded.

[2] CRAMFS header starts at offset 0xe0000 in the upgrade file.

```
root@ncplus:~# cramfsck -x iti-5800s /mnt/USB/u.cramfs
PATH iti-5800s/bin/busybox
PATH iti-5800s/etc/.profile
PATH iti-5800s/etc/fstab
PATH iti-5800s/etc/group
PATH iti-5800s/etc/host.conf
PATH iti-5800s/etc/init.d/modules_init
PATH iti-5800s/etc/init.d/modules_insmod
PATH iti-5800s/etc/init.d/rcS
PATH iti-5800s/etc/inittab
PATH iti-5800s/etc/iptables.sav
PATH iti-5800s/etc/nsswitch.conf
PATH iti-5800s/etc/passwd
PATH iti-5800s/etc/protocols
PATH iti-5800s/etc/udhcpc.script
PATH iti-5800s/lib/firmware/adb_comp_audio_MME311.bin
PATH iti-5800s/lib/firmware/adb_comp_video_MME311.bin
PATH iti-5800s/lib/firmware/as102data1.hex
PATH iti-5800s/lib/firmware/as102data2.hex
PATH iti-5800s/lib/firmware/st_comp_video_MME311.bin
PATH iti-5800s/lib/ld-2.3.3.so
PATH iti-5800s/lib/libBrokenLocale-2.3.3.so
PATH iti-5800s/lib/libSegFault.so
...

root@ncplus:~# ls -la iti-5800s/root
total 21892
drwxr-x---  2 root root     4096 2018-02-05 22:55 .
drwxr-xr-x 11 root root     4096 2018-02-05 22:55 ..
-r-xr--r--  1 root root      175 1969-12-31 19:00 dhcpc.sh
-r-xr--r--  1 root root      136 1969-12-31 19:00 eu
-r-xr--r--  1 root root 22162800 1969-12-31 19:00 main.elf
-r-xr--r--  1 root root      125 1969-12-31 19:00 netd.sh
-r-x------  1 root root     8156 1969-12-31 19:00 root.elf
-rwxr-xr-x  1 root root   191136 1969-12-31 19:00 splash
```

The main MHP application (`main.elf`) was not different from what we encountered in 2012 [6]:

- one big statically linked image,
- a custom Java File System (statically linked ROMFS filesystem),
- obfuscated Java classes for operator specific MHP set-top-box application.

We made use of our reverse engineering tools developed back in 2012 (simple Hitachi SH4 CPU emulator in particular) for extraction and unpacking of the contents of a ROMFS filesystem:

```
ncplus> dromfs
rom0
 - addr 144c9b8
 - size 1c0
   rom0/boot size 373 (packed 269)
   rom0/com/adb/init/PluginInitTable.class size 373 (packed 269)
...

rom18
 - addr 1898488
 - size 26928
   rom18/ait size 1970 (packed 771)
   rom18/app.jar size 181194 (packed 1259)
   rom18/appstorage.zip size 1268 (packed 632)
   rom18/dvb.certificates.1 size 3303 (packed 2599)
   rom18/dvb.hashfile size 90 (packed 65535)
   rom18/dvb.signaturefile.1 size 257 (packed 65535)
   rom18/dvb.storage.0000002d.5600 size 299 (packed 176)
```

It's worth to note that among the files included in a ROMFS dump, there was a Java Xlet application (`com.adbglobal.application.watermark.xlet.WatermarkXlet`) from ADB company that implemented watermarking solution for content protection[3]. As the goal of this application is to make tracking of illegally distributed content possible, its leak naturally makes this security mechanism rather mute. This is due to the possibility to decompile and analyze the watermarking mechanism and remove any hidden tags introduced by it (such as smart card and STB numbers) to protected A/V content.

The other interesting file was an obfuscated Java `adb_h.Dq` class that contained string references to `sw_upgrades/apollo/multibox` URL path and that implemented AES crypto operations (references to `Crypto`, `IvParameterSpec` and `SecretKeySpec` classes in code).

Upon deeper inspection of the implementation of `Dq` class, we figured out that it's goal was to handle multibox server responses and their ciphering. We implemented a quick Java code to verify our findings:

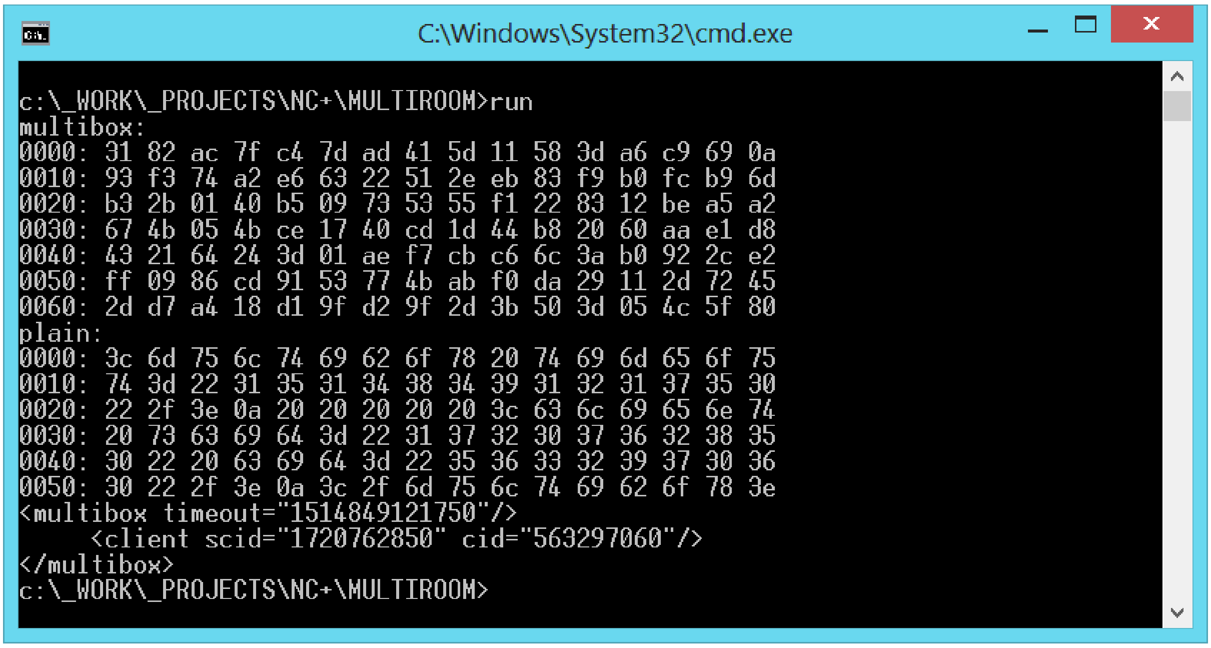
```
 try {
   byte key[] = {
       38, 36, -51, -4, -46, 107, -72, -4, -43, -98, 100, 60, -59, 112, 18, -7
   };

   byte iv[] = {
       0, 1, 2, 5, 3, 4, 6, 11, 8, 9, 7, 10, 12, 14, 13, 15
   };

  byte multibox_data[]=load_file("multibox.dat");
  dump_buf("multibox",multibox_data);

  Cipher cipher=Cipher.getInstance("AES/CBC/PKCS5Padding");
  cipher.init(Cipher.DECRYPT_MODE,new SecretKeySpec(key,"AES"),
                                  new IvParameterSpec(iv));

  byte plain[]=cipher.doFinal(multibox_data);
  dump_buf("plain",plain);

  System.out.println(new String(plain));
 } catch (Throwable t) {
  t.printStackTrace();
 }
}
```

The result of running it on a multibox file fetched from a Multiroom server is illustrated on Fig. 5.

It turned out that a shared key used to secure the Multiroom service offering of NC+ platform was hard coded in an obfuscated Java class file embedded within a binary of the main MHP application. More specifically, a predefined initialization vector and a key for AES cipher working in CBC mode was defined within it.

Decrypted multibox data comprised of an XML like message denoting a time when next verification should be performed (*timeout* field), smart card number (*scid* field) along with a chip id number (*cid*) of a client device authorized by a given Multiroom server. Thus, security of a Multiroom Standard service relies on a verification of the card and chip id numbers received from the server with the numbers corresponding to a client device.

[3] VoD content in particular.



```
c:\_WORK\_PROJECTS\NC+\MULTIROOM>run
multibox:
0000: 31 82 ac 7f c4 7d ad 41 5d 11 58 3d a6 c9 69 0a
0010: 93 f3 74 a2 e6 63 22 51 2e eb 83 f9 b0 fc b9 6d
0020: b3 2b 01 40 b5 09 73 53 55 f1 22 83 12 be a5 a2
0030: 67 4b 05 4b ce 17 40 cd 1d 44 b8 20 60 aa e1 d8
0040: 43 21 64 24 3d 01 ae f7 cb c6 6c 3a b0 92 2c e2
0050: ff 09 86 cd 91 53 77 4b ab f0 da 29 11 2d 72 45
0060: 2d d7 a4 18 d1 9f d2 9f 2d 3b 50 3d 05 4c 5f 80
plain:
0000: 3c 6d 75 6c 74 69 62 6f 78 20 74 69 6d 65 6f 75
0010: 74 3d 22 31 35 31 34 38 34 39 31 32 31 37 35 30
0020: 22 2f 3e 0a 20 20 20 20 20 3c 63 6c 69 65 6e 74
0030: 20 73 63 69 64 3d 22 31 37 32 30 37 36 32 38 35
0040: 30 22 20 63 69 64 3d 22 35 36 33 32 39 37 30 36
0050: 30 22 2f 3e 0a 3c 2f 6d 75 6c 74 69 62 6f 78 3e
<multibox timeout="1514849121750"/>
     <client scid="1720762850" cid="563297060"/>
</multibox>
c:\_WORK\_PROJECTS\NC+\MULTIROOM>
```

Fig. 5 The result of multibox file decryption.

We verified that in order to bypass a security of the mechanism described above, it is sufficient to:

- setup a simple HTTP server listening on port 8080 and handling multibox requests from client devices,
- return a modified version of an original multibox response[4] with a *timeout* field set to a large value (time of a next verification of client devices set in a far away future):

```
<multibox timeout="11517921609447"/>
   <client scid="1720762850" cid="563297060"/>
</multibox>
```

The described bypass was successfully verified for ITI-2851S set-top-box device configured to operate as a Multiroom client with software upgrade from Jan 2018 (SW ver. 0X1A, MHP ver. 1.2.0 V15.3-NCP4740SF-RC-25-G049EF0C, APP ver. 4.4.6:6094982, SW id 5.2.4).

The same Multiroom key is also used by ITI-3740SX (main device from which we initially acquired / decrypted multibox response) and ITI-5800S (the device of which SSU leaked the key). Thus, it is safe to assume that the described bypass should work on all Multiroom devices NC+ has in its Multiroom service offer.

SUMMARY

Our analysis indicates that public claims made by NC+ operator[5] hardly reflect the reality. Regardless of the obligations imposed by content providers, high security of the offered

[4] encrypted with the use of a shared AES key and containing original smart card and chip id numbers.

[5] On Jan 11, 2018 Platform NC+ issued an official message to subscribers about its policy of content security [7]. Among other things, the following statements were included in it: "Platform nc+ as a technology lider in the market and an operator with a rich program offer conducts many activities aimed at providing a high security of the offered content". "In order to fulfill the requirements of content providers, platform nc+ is obliged to completely secure the Multiroom service".

content / complete security of a Multiroom service is not necessarily fulfilled in the environment of NC+. This manifests in the following:

1. Nearly 6 years following a public disclosure of hardware issues in STMicroelectronics' DVB chipsets [8], set-top-boxes proved to be vulnerable to Control Words sharing and PayTV piracy are still deployed in the field in 2018[6],
2. implementation of a Multiroom service with an inadequate level of security (hard coded cryptographic key in compressed and obfuscated binaries broadcasted[7] via HotBird 13° East satellites), ignorance of hardware security features (such as secret chipset SCK key) for a Multiroom service implementation (Multiroom key handling should correspond to pairing key handling),
3. a code of a watermarking solution available as part of the same broadcast as in 2.

We suspect NC+ might not be aware of 2 and 3 as security of set-top-box devices, their functionality and associated communication protocols' implementation such as `AdbEmmCarousel`, SSU and Multiroom is a primary responsibility of a STB vendor.

The nature of the issues disclosed in 2012 and now emphasize the need for PayTV operators to always question and verify the claims of set-top-box, chipset and CAS vendors and never take them for granted in particular.

REFERENCES

[1] Multiroom na platformę nc+
`http://ncplus.pl/multiroom`

[2] NC+ Platform
`http://ncplus.pl/`

[3] Instrukcja konfiguracji dekodera dodatkowego BOX+ HD ITI-5800S
`http://ncplus.pl/~/media/ncplus/dokumenty/instrukcje%20obslugi/multiroom-2017/1804b/dekoder-dodatkowy/konfiguracja-dla-dekodera-box-5800s.pdf`

[4] Advanced Digital Broadcast
`https://www.adbglobal.com/`

[5] SE-2011-01 Issues #5-16,#25-32 (Advanced Digital Broadcast),
`http://www.security-explorations.com/materials/se-2011-01-adb.pdf`

[6] "Security threats in the world of digital satellite television", HITB Talk #1,
`http://www.security-explorations.com/materials/se-2011-01-hitb1.pdf`

[7] Polityka Zabezpieczenia Treści
`http://ncplus.pl/zabezpieczenie-tresci`

[8] "Security vulnerabilities of Digital Video Broadcast chipsets", HITB Talk#2
`http://www.security-explorations.com/materials/se-2011-01-hitb2.pdf`

[6] NC+ replaces old devices with new models for a monthly fee [9][10], although it might be sole responsibility of the operator to do it taking into account its obligation to secure PayTV content (the replacement costs of vulnerable devices should not be subsidized by end users).

[7] service denoted by `dvb://13e.514.3ad4` URL locator and MPEG PID 0x0bbf.

[9] WYMIEŃ SWÓJ DEKODER NA NOWY!
`http://ncplus.pl/wymiana-dekodera`

[10] Screenshot of NC+ operator web page advertising STB replacement to a new model
`http://www.security-explorations.com/materials/ncplus_screenshot.png`

About Security Explorations

Security Explorations (`http://www.security-explorations.com`) is a security start-up company from Poland, providing various services in the area of security and vulnerability research. The company came to life in a result of a true passion of its founder for breaking security of things and analyzing software for security defects. Adam Gowdiak is the company's founder and its CEO. Adam is an experienced Java Virtual Machine hacker, with over 100 security issues uncovered in the Java technology over the recent years. He is also the Argus Hacking Contest co-winner and the man who has put Microsoft Windows to its knees (the original discoverer of MS03-026 / MS Blaster worm bug). He was also the first expert to present a successful and widespread attack against mobile Java platform in 2004.