Codenomicon whitepaper:

Smart TV Hacking: Crash Testing Your Home Entertainment

- Rikke Kuipers, Eeva Starck & Hannu Heikkinen -

PREEMPTIVE SECURITY AND
ROBUSTNESS TESTING SOLUTIONS

1 Introduction

Home entertainment has expanded beyond the traditional television. Modern TV sets are similar to desktop computers: they have a processor, memory, a hard disk and some sort of an operating system. They are now constantly connected to the Internet and offer a wide range of online services such as videos, music, online shopping and various web services. All these connections use communication protocols which must be tested and hardened to be secure.

This white paper describes how we found unknown vulnerabilities in smart TVs and the implications of our findings. First we map the attack surfaces of a wide range of TVs, then conduct fuzz testing, and finally discuss the test results. All tested smart TVs crashed with a number of protocols.

2 From Simple to Smart TVs

Modern Internet-enabled TVs are basically very specialized computers. Most of them run a stripped down version of Linux, UNIX or Windows. The OS provides access to the Internet and other services such as image/video/audio decoding. Some of the more expensive models have full-blown browser implementations. They have pre-installed applications (or widgets), and some allow you to download new ones. Multimedia content can be played from USB-connected devices or SD-cards.

3 Attack surfaces

The attack surface can be divided to active and passive. The active attack surface refers to all the attack vectors of a given device where the attacker has to actively transmit data to perform the attack. For TVs, this is mostly determined by the various services running on the TV. Bluetooth is more and more commonly found in TVs, and is definitely an interesting and well-known attack vector in other embedded devices.
The passive attack surface refers to interfaces the attackers can "listen in on" to gain information without altering the data in any way. Our focus in this paper is in the active attack surface. The attack surface depends heavily on the type of TV:

Dumb DVB-enabled TVs

So-called dumb TVs do not have much in terms of connectivity, except for DVB. The standard Digital Video Broadcasting (DVB) protocol is the successor of analog broadcasting used in more than 80 countries worldwide. There are many DVB standards, each developed for its own intended use. The first of the DVB standards to be agreed upon by ETSI and others was the DVB-S standard (1994) for satellite transmission. DVB-T is used for terrestrial transmissions, and was commercialized around 1997. DVB-C is used in cable transmissions.

Media Center TVs

Media Center TVs provide basic network connectivity and external media support from USB connectors and memory card readers. Network connectivity may appear to be limited to the local area network, providing media services and firmware updates.

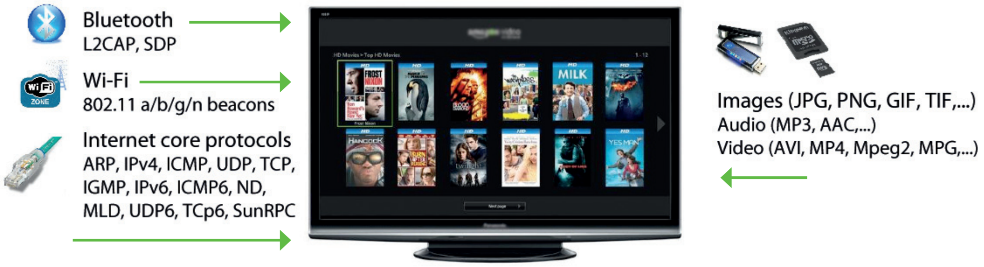
Bluetooth
L2CAP, SDP
Wi-Fi
802.11 a/b/g/n beacons
Internet core protocols
ARP, IPv4, ICMP, UDP, TCP, IGMP, IPv6, ICMP6, ND, MLD, UDP6, TCp6, SunRPC
Images (JPG, PNG, GIF, TIF,...)
Audio (MP3, AAC,...)
Video (AVI, MP4, Mpeg2, MPG,...)

Figure 1: TV attack surfaces

Besides UPnP and DLNA type media center protocols and a basic IP-stack, DHCP and HTTP/FTP connections are used just to fetch LAN based media content or firmware upgrades over WAN. These TVs may host some low level network services themselves.

Internet-Enabled TVs

In addition to the functionality of the Media Center TVs, these even more advanced TVs host applications, applets and widgets which enable a channel to pull dynamic content directly from the Internet, typically from various web services. New applications and application updates can be fetched and installed directly from the Internet.

This level of openness in terms of attack surface is very similar to current smart phones or tablets, both of which have a connection to the Internet. The TV's browser may be used to access a wide variety of external content, exposing it to an even wider variety of attacks.

4 Threat Scenarios

1. Denial Of Service attacks

A DoS attack means crashing the services in a TV so that a manual reboot of the TV is required for it to operate again. In some cases a firmware flash from the manufacturer is needed to fix permanent damage caused by the attack.

2. Exploits

When a vulnerability is analyzed, debugged and turned into a working exploit, TVs are even more defenseless than computers. Malicious code can be run on the TV to gain unauthorized access to the TV. Countermeasures such as memory protection technologies are not available in most TVs, making them easy targets. Detecting a breach can be hard because diagnostic tools for TVs are not available and users have no operating system level access to the TV.

3. Covert Malware

Botnets and other espionage software can be installed and remain undetected in home entertainment systems. Hidden malware can access functionalities such as cameras (e.g. Kinect motion detection) and microphones on the TVs. As TV distribution is often very homogenous, running only a limited range of operating systems, a single instance of malware can infect and populate hundreds of millions of homes, creating powerful botnets for launching Denial of Service attacks.

4. Loss of sensitive data

With the rise of "on-demand" services, such as the ability to rent movies and watch TV shows within minutes, the TV has to support payment options. Credit card numbers are saved on the TV itself, which implies a possibility that they could be extracted. Services like email and social media pull and store sensitive data on the TV, which could be accessed by an attacker.

5. Social Engineering and Static Media

A much underestimated vector is the delivery of content to a TV by static media such as USB sticks and memory cards. The software on a TV decodes movie files and displays pictures, which can be malicious and cause crashes or buffer overflows. Users can be tricked to inserting malicious USB media into the TV.

5 What is Fuzzing

Fuzzing is a proactive technology that can be used to assess the robustness and implementation level security characteristics of any system. Fuzz testing or fuzzing is a software testing technique, often automated or semi-automated, that involves providing invalid, unexpected, or random data to the inputs of a computer program. The program is then monitored for exceptions such as crashes or failing built-in code assertions. Fuzzing is commonly used to test for security problems in software or computer systems.

There are two primary forms of fuzzing: mutation-based and generation-based. File formats and network protocols are the most common targets of testing, but any type of program input can be fuzzed. Interesting inputs include environment variables, keyboard and mouse events, and sequences of API calls. Even items not normally considered "input" can be fuzzed, such as the contents of databases, shared memory, or the precise interleaving of threads.

Since the fuzz tests utilize the actual communications interfaces to the application being tested, they are implementation language neutral. Fuzzing does not require access to the source code and it is relatively simple and straightforward to execute, producing no false positives. A crash is always a crash.

6 Methodology

For this research, we used Codenomicon Defensics, a powerful set of fuzzing tools based on comprehensive protocol models. The test cases were created automatically and sent to the system under test. The TVs' firmware was updated to the latest available. All available services in the TVs were enabled in order to get the maximum test coverage. The TV was then scanned for open ports and services to map the attack surfaces. In addition, any accompanying documentation was consulted. All the discovered protocols were then tested by fuzzing with the appropriate Defensics suite.

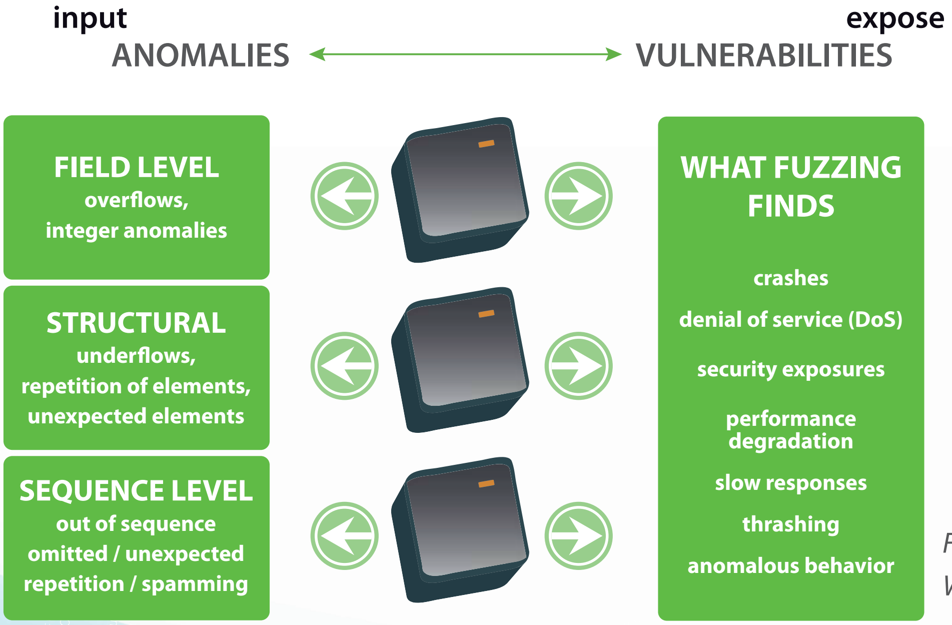

Figure 2: What fuzzing finds

The DVB Protocol

The most interesting protocol to test in TVs is DVB. The DVB protocol is not used just to transmit video/audio streams to TVs. It is also used for local communication (e.g. device-to-device transmission), navigational purposes, access to content on hand-held devices (DVB-H), and even to provide Internet access in remote locations where no cable or mobile communication is possible (IP-over-DVB/MPEG). Aside the commercial implementation, a second generation DVB satellite signaling (DVB-S2), is used by military organizations. As with most protocols, DVB implementations in devices have vulnerabilities when not properly implemented.

On a high level, content providers such as cable companies deliver one DVB stream to the TV. This stream contains several channels, each on its own frequency. The channels are combined, or "muxed" into one transport stream and delivered to the TV, which "demuxes" the signal so it can be "read" from the various channels. In addition to the audio/video streams (the payload), there are also a number of tables included in the transport stream. These tables provide the TV with information about the stream. An example is the Program Association Table, which lists all available programs in the transport stream.

Depending on the guidelines set per country, a TV can be picky about which tables should be present in the transport stream. When fuzzing DVB, to mimic the exact stream a TV would normally accept, we used a capture device to record directly from the DVB-T/C source. This captured transport stream was then run through the Defensics MPEG TS suite in order to create a fuzzed version of the transport stream. In the last step, we muxed the transport streams into a DVB signal and sent it back to the TV using a modulator. Because there was no instrumentation possible by using just the protocol, we used an ICMP heartbeat to verify the health of the TV under test.

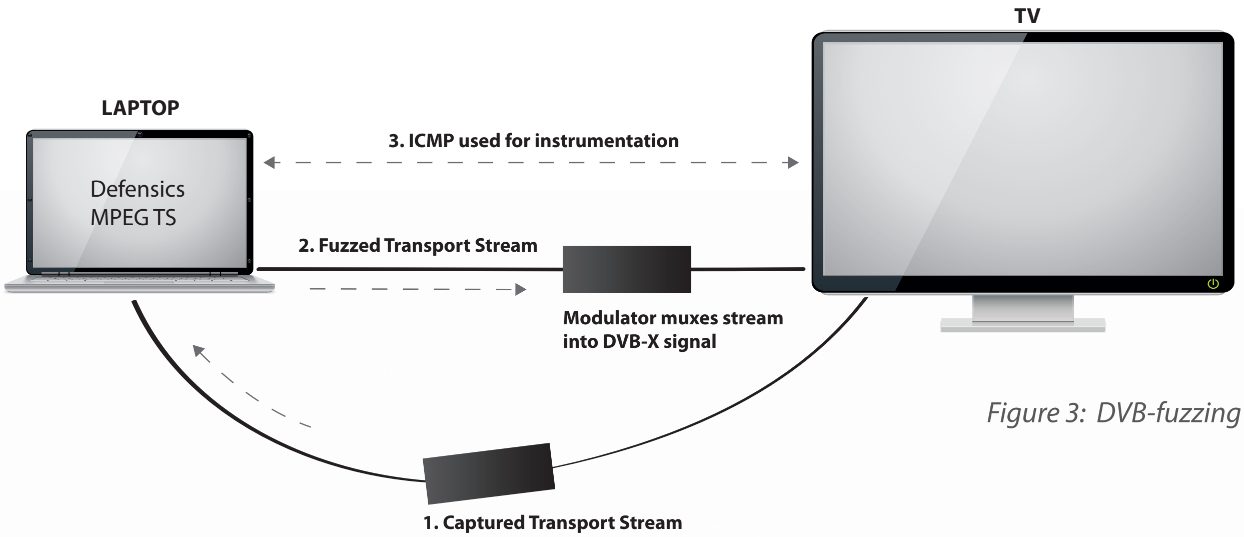
TV
LAPTOP
Defensics
MPEG TS
3. ICMP used for instrumentation
2. Fuzzed Transport Stream
Modulator muxes stream
into DVB-X signal
1. Captured Transport Stream

Figure 3: DVB-fuzzing

7 What We Found

These are the results from our tests conducted with the Defensics test suites.

Protocol	Vendor 1	Vendor 2	Vendor 3	Vendor 4	Vendor 5	Vendor 6
IPv4	pass	fail	fail	pass	pass	fail
DVB	fail	fail	fail	fail	fail	fail
UPnP	n/a	fail	pass	n/a	n/a	fail
Images	pass	fail	fail	n/a	n/a	fail
Audio	pass	pass	n/a	n/a	n/a	pass
Video	fail	fail	pass	fail	fail	fail

Legend:

- pass - survived
- fail - vulnerable
- n/a - not available for testing

A test was deemed failed if the TV behaved abnormally: either it did not respond, responded in a way that was incorrect, or crashed. A pass was given when the device responded to an anomaly as expected (no response or error message), and continued responding correctly to regular, non-anomalous messages.

While testing the IPv4 stack we discovered vulnerabilities in some of the TVs. Unfortunately none of the TVs tested were IPv6 capable. As can be seen from the result table, all of the TVs crashed with several protocols. Surprisingly, protocols which have been around for a very long time, such as IPv4, still cause some models to crash or malfunction. This could be due to the use of an old kernel, a custom stack implementation or a CPU with insufficient power to handle larger or fragmented packets.

Most failures occurred in what should be the TV's primary function: decoding multimedia. Simple fuzzed images crash the TV with ease, and consequently they present an interesting attack vector due to the numerous way of distributing the possible exploits to the victim, such as embedding the hostile images into web pages.

Video decoding generated crashes on most TVs as well. These crashes were found using the same video sample on each of the TVs. This means that not all of the codecs within the TVs were tested, when one simple one caused such crashes.

As discussed earlier, the DVB protocol is interesting for several reasons. The protocol has undergone years of conformance testing, but apparently robustness testing was never required.

8 Conclusion

Traditionally, security people tend to think that smart TVs need strong security mechanisms in order to remain safe. Of course some basic security measures need to be there such as password protection for ensuring the users' privacy. Nonetheless, it is impossible to determine whether your digital television has been compromised. Anti-virus for embedded devices does not work, and home firewalls cannot detect all attacks against tailor-made TV applications. But fact remains that smart TVs have similar vulnerabilities as desktop PCs.

The only solution is to test for security. Black-box testing techniques like fuzzing should be integrated in the software development lifecycle to eradicate security vulnerabilities before rolling the TVs out and exposing consumers to the threats. As shown in this study, the TV-specific protocols are likely to open up new attack vectors which can be used to attack the TV set. Zero-day vulnerabilities in communication protocols open the devices to remote compromise.

This research shows that the video protocols were surprisingly unstable, which is disappointing when considering that it is one of the key features a TV is supposed to handle. The DVB protocol seems to be vulnerable throughout the tested smart TVs. The DVB is an interesting attack vector for several reasons. In the case of "simple" customer devices such as TVs, these kinds of vulnerabilities could be triggered remotely by overpowering the DVB-T signal in the air. Another possible attack would be to buy airtime from the commercial break, provide the broadcaster with your normal looking shampoo commercial, i.e. your prepared transport stream, that would contain exploits for the top 10 Internet-enabled TVs. Military implementations of DVB could be remotely crashed, get infected with malware, be eavesdropped upon, or at worst, get controlled by the attackers.

The TV manufacturers should be able to provide updates in a quick and easy manner for their deployed TVs. Open source software used in TVs poses a threat of its own, as known vulnerabilities in open source are always public. Reacting to found open source bugs has to be immediate, which is not always possible with TVs. Expensive and automated patch release process has to be in place if the security bugs are not found proactively, before release.

STP Ethernet Finger GARP FTP GRE GTPv0 GTPv1 H.248 H.323 HSRP HTTP ICMP ICMPv6 IGM
v2 OSPFv3 PBT/PBB-TE
TP GRE GTPv0 GTPv1 H.248 H.323 HSRP HTTP ICMP ICMPv6 IGMP IKEv2 IMAP4
SigComp SIP SIT TT SMBv2 SMPP SMTP SNMPv1 SNMPv2c SNMPv3 SRTP SSH1 SSH2 STP STUN
AKMP/IKE iSCSI Kerberos L2TPv2 L2TPv3 LACP LDAPv3 LLDP LPD MGCP MIPv6 MIME Modbus MPLS/LDP MSTP NFS NHRP NTP O
NFS NHRP NTP OAM OCSP OSPFv2 OSPFv3 PBT/PBB-TE PIM-DM/SM
8 Finger GARP FTP GRE GTPv0 GTPv1 H.248 H.323 HSRP HTTP

CODENOMICON
defensics®

Codenomicon develops proactive security testing software and situation awareness tools that help discover problems at the earliest possible moment.

Defensics is the world leading fuzzing solution. It provides fully automated security testing suites for over 200 communication interfaces. Defensics uses model-based, systematic fuzz testing to provide the best testing coverage.

Situation awareness solutions create interactive visualizations from real-time network traffic and abuse information. Status of the networks and critical resources can be seen in one glance, allowing better informed and faster decision making.

Codenomicon's solutions are used by top governments and leading software companies, operators, service providers and manufacturers.

http://www.codenomicon.com

CODENOMICON Ltd.
info@codenomicon.com
www.codenomicon.com

Tutkijantie 4E
FIN-90590 OULU
FINLAND
+358 424 7431

12930 Saratoga Avenue, Suite B-1
Saratoga, CA 95070
UNITED STATES
+1 408 252 4000

25/F., Queen's Road Centre
152 Queen's Road Central
HONG KONG
+852 3426 22900

46B Tras Street
SINGAPORE 078985
+65 9188 1502