Investigating HbbTV Privacy Invasiveness Across European Countries

Carlotta Tagliaro
TU Wien
Austria
carlotta@seclab.wien

Florian Hahn
University of Twente
Netherlands
f.w.hahn@utwente.nl

Riccardo Sepe
Guess Europe Sagl
Italy
riccardo@sepe.it

Alessio Aceti
Sababa Security SpA
Italy
alessio.aceti@sababasecurity.com

Martina Lindorfer
TU Wien
Austria
martina@seclab.wien

***Abstract*—Smart TVs enable the integration of the traditional broadcast signal with services offered by the Internet. Specifically, the Hybrid Broadcast Broadband TV (HbbTV) protocol allows broadcasters to offer consumers additional features via the Internet (e.g., quizzes and the ability to restart programs), enriching their viewing experience. For broadcasters its bi-directional nature also enables them to measure viewing preferences and provide targeted advertisements (marketed as "Addressable TV"). HbbTV works using standard web technologies as transparent overlays over a TV channel, thus, porting web security and privacy concerns to the Smart TV. However, despite the increasing adoption of HbbTV worldwide, studies on security and privacy issues in its deployments are scarce.**

In this paper, we discuss how we tested a range of 36 channels across five European countries and which challenges we faced; Specifically, every country adopts different ways of delivering the broadcast signal to the TVs. Thus, we provide a common experiment setup and detailed instructions on how we assess the TV channels' privacy level in each country. We also show how the URLs pointing to the HbbTV applications we extracted can foster further replicability and studies. Finally, to complement our technical experiments we also measured Italian users' awareness (N=174) of the security and privacy risks HbbTV introduces and we discuss our methodology to do so.

I. Introduction

As of 2021, 1.72 billion TV households exist worldwide [48], and each viewer, on average, spends around three hours per day watching TV [45], [46]. Thus, TV content can significantly impact society as a whole, for example, depending on the content and spin of news headlines—in addition to being a valuable target for advertisers. To combine standard TV's broadcast content delivery with the powerful digital content delivery of the new platforms and improve the viewing experience for users, an industrial consortium launched the *Hybrid Broadcast Broadband TV (HbbTV)* [27] initiative in 2009. HbbTV sets a standard for a broadcast/broadband hybrid protocol to deliver content to Smart TVs, set-top boxes, and other connected multiscreen devices in an interconnected environment. In this setting, an HbbTV application is loaded and executed by a Smart TV's built-in browser and displayed as a graphic overlay on top of regular broadcast content. HbbTV transforms the traditional TV viewing experience from merely *receiving* content to also *transmitting* data, enabling new functionality for consumers (e.g., interactive programming and shopping) and broadcasters (e.g., measuring viewing preferences and providing targeted content).

In particular the opportunity of exploiting the bi-directional nature of HbbTV for targeted advertisement ("Addressable TV") based on household demographics and behavior, such as their viewing preferences) keeps being promoted by industry. Most recently, in November 2022, during the HbbTV Symposium and Awards, major industry players took steps to deploy the *HbbTV specification for Targeted Advertising (HbbTV-TA)* in the European market [28]. While representing a new possibility for monetization for broadcasters and other stakeholders, this also introduces security and privacy risk for consumers, such as profiling and targeting—privacy issues that users so far ar mainly familiar (and concerned) with on the web [51], [17], [52] and in mobile apps [44], [22], [35]. In both the web and mobile domain privacy invasions have been extensively studied, resulting in methodologies and pipelines for assessments of websites [20], [12], [32], [36], [29], [23] and Android/iOS apps [42], [38], [43], [41], [31], [33], [30] at scale. Related work has further documented the practice of "cross-device tracking" that links user profiles across mobile and desktop devices [55], [16]. One effective feature do so so is using the devices' IP address, something that could potentially be used to also link the behavior observed on the Smart TV to a users' profile gathered from other devices. Given these privacy threats, we hope that further studies will provide more transparency on HbbTV-based profiling. However, a portable and reliable testing pipeline is essential to enable further research in this direction. We stress that our threat (and thus testing) scenario is different from related work: instead of focusing on tracking by apps installed on Smart TVs and over-the top (OTT) video streaming devices [34], [50], we study the content transmission protocol deployed by broadcasters to deliver the TV channels.

In the remainder of this paper, we discuss our experiment setup for identifying privacy issues raised by HbbTV, what equipment was required, and which testing methodologies we adopted. We found that experiments performed in previous

Learning from Authoritative Security Experiment Results (LASER) 2023
3 March 2023, San Diego, CA, USA
ISBN 1-891562-86-X
https://dx.doi.org/10.14722/laser-ndss.2023.24102
www.ndss-symposium.org

research are not viable anymore due to the greater adoption of encrypted communication channels [26], [25], [24]. We developed a testing methodology to bypass this issue with the added benefit of not requiring a Smart TV, thus also fostering reproducibility and easing geographical barriers. In the first part of this paper (Section IV), we present the challenges we faced, mainly in gaining privileged access to the Smart TV, as well as the common characteristics between our *On-TV* and *Off-TV* experiments. In the second part of this paper (Section VI), we discuss our methodology for the HbbTV risk awareness survey we performed with 174 users of Smart TVs in Italy. In the final part of this paper (Section VII), we discuss the parts of our methodology that we borrowed from previous research and highlight the need for replication studies, e.g., updated versions of past measurement studies.

Full Study. For details on the study for which we developed this testing methodology please see our NDSS'23 paper [49].

Artifacts. The source code of our experiment setup is available at https://github.com/SecPriv/hbbtv-blocker.

II. Background

Hybrid Broadcast Broadband TV (HbbTV). As stated by the HbbTV Association [27], HbbTV is *"... a global initiative aimed at harmonizing the broadcast and broadband delivery of entertainment services to consumers through connected TVs, set-top boxes, and multiscreen devices."* In other words, it represents both a widely adopted standard (the ETSI Technical Specification 102 796 [21]), and a driving force to promote a unified hybrid TV delivery across different platforms [15] offering broadcast and broadband content to viewers. The HbbTV initiative started in 2009 when an industrial consortium led by the German broadcaster RTL decided to keep standard television at pace with new digital media services being developed.

Content Delivery and Format. A Smart TV can support two connections in parallel; On one side, it is connected to a broadcast Digital Video Broadcasting (DVB) network. On the other side, it is connected to the Internet via a broadband interface. The TV receives standard broadcast Audio/Video (A/V) content through the first one. The Internet connection allows for bi-directional communication with the provider and can receive non-linear A/V content.

The Internet-delivered HbbTV applications are embedded as URLs in the DVB stream sent by broadcasters. Such URLs, after being extracted, are sent to the Runtime Environment where the TV's browser, responsible for presenting and executing the application, is located. Any website with standard web techniques (e.g., HTML, JavaScript) can serve content. Such applications are then presented to the consumers through transparent graphical overlays on top of current TV broadcasts.

Main Study Results. This paper presents supplemental material on the experimental aspect of our main research contribution published at the Network and Distributed System Security Symposium (NDSS'23) [49]. While the structure and content are similar, here we highlight our testing equipment and setup and describe our experiments and procedures more in-depth. For completeness, we briefly summarize our main results: Considering that we performed our study in European countries, we contextualized our results in light of current data protection regulations in force in the EU, such as the General Data Protection Regulation (GDPR). Out of the 36 channels we analyzed, 26 (72.2%) start tracking and profiling the consumer before receiving their consent to the data treatment policy; Notably in Austria, all four channels contact track.tvping.com every second even before receiving user consent. Further, seven channels do not present any privacy policy at all, not in any submenus of the application. Both practices violate the "Conditions for Consent" article of GDPR. We further found *tracking pixels* to be a popular technique for user profiling and deployed across 20 (55.5%) of the 36 studied channels. Such a technique consists of uploading a 1×1 pixel image, which is invisible to the naked eye when the user visits a website or opens specific content to track their behavior. Finally, a German shopping channel, HSE, still handles users' personal information and credit card data in plaintext, allowing attackers to steal and misuse this data—even though this issue has been known since 2014.

III. Testing Infrastructure

Our testing infrastructure consists of five devices: Two Smart TVs, a Xiaomi Mi 4A Smart TV (Android 9, released 2018) and a Samsung M5500 Smart TV (Tizen 3.0, released 2017), a laptop running Ubuntu 20.04, an antenna and a HiDes UT-100c (de)modulator. The modulator is needed to modulate the broadcast signal captured by the antenna and "forward" it to the laptop on which it is parsed. We selected the modulator based on its stable support on Linux systems.

Ideally, one Smart TV should suffice; however, we found that some HbbTV applications did not run as intended on the Android-based Smart TV while correctly functioning on the Samsung one. Samsung Smart TVs are based on a different operating system, Tizen; Thus, we assume a different implementation of internal software components might be the reason. We did not further investigate the reason for such inconsistent behavior, and we leave this for future work.

The HiDes UT-100c[1] is a USB-based (de)modulator supporting DVB-T ('T'=Terrestrial, in contrast to 'S'=Satelite in DVB-S, and 'C'=Cable in DVB-C) transmission and reception.[2] The device is directly powered from the USB bus, and, given that it is already equipped with its CPU, no host CPU computation is required. Such a device can be easily found on eBay for around EUR 150 (∼ USD 169), thus making our infrastructure easily and cheaply replicable. For other supported modulators, either for terrestrial or other broadcast signals, please refer to the TSDuck [8] documentation.

[1] http://www.hides.com.tw/product_cg74469_eng.html (last accessed April 2023 and archived at https://archive.is/QqwtE)

[2] We already had this device available in our lab, hence, we focus on terrestrial signals. Nevertheless, our approach could be replicated for satellite or cabled signals with a slightly different setup.

IV. On-TV vs. Off-TV Experiment Setup

We follow two strategies to evaluate the HbbTV protocol and perform both *On-TV* and *Off-TV* experiments (see Figure 1). In the former setting, we record 90 minutes of traffic while in the later we only capture 30 minutes. We determined these timeframes experimentally trying to capture all relevant traffic within the testing time; The *On-TV* timeframe is longer as working with a Smart TV results in slower interaction times.

A. On-TV Experiments

We start by capturing all traffic generated and directed to the Smart TV in our *On-TV* test; We place our laptop as a Monkey-In-The-Middle (MITM) proxy between the Smart TV and the Internet and use Wireshark [10] to record all traffic. We adapt our methodology from previous work by Ghiglieri et al. [25]. However, we encountered traffic encrypted using the Transport Layer Secure (TLS) protocol, thus making it not easily inspectable. To address this issue and intercept encrypted traffic, we needed to make the Smart TV trust the proxy's self-signed Certificate Authority (CA) certificate. Such certificates must be placed in a root-accessible folder in the Android file system (/system/etc/security/cacerts). Thus, root privileges on the Smart TV are required. We only perform rooting attempts on the Xiaomi device and not on the Samsung device as it was loaned to us by family members.

Rooting Attempt #1: Security Vulnerabilities. For trying to gain root access to the TV we first checked whether any privilege escalation was present in Android 9 (security patch level of September 2020); However, at the time of our experiments, this was not the case. The Android Security Bulletin of October 2020 [1] shows that previous versions were vulnerable to privilege escalation attacks (marked with *EoP* for "Elevation of Privilege"). However, we cannot confirm that these issues indeed apply to Android-based Smart TVs or that viable exploits are available. Additionally, we also checked Metasploit [4] but found no exploits applicable to our device.

Rooting Attempt #2: Bootloader Image. Next, we tried to look for the Smart TV's bootloader image in its partition scheme. If we can extract such an image, we can feed it to Magisk [11], a systemless rooting system, patch it and flash it back to the TV to gain the needed permissions. When checking the partition scheme with df we notice that no boot partition is present; This means such partition is part of the root filesystem and boot.img is a combined kernel and ramdisk image made with special tools (u-boot-tools on Debian-based systems). Searching for the file in all the user-accessible folders in the file system did not yield any results either. In addition, no firmware image for our device is to be found, so boot.img cannot be extracted from the stock firmware as well. Furthermore, rooting requires the bootloader to be unlocked; When we try to unlock it, the Smart TV shuts down and shows a black screen (the only way to make it work again is to unplug it from electrical power). To unlock the bootloader, first, the USB debugging should be enabled, and OEM should be unlocked as well in the developer options of the Smart TV settings. We then installed the Android Debugging Bridge (ADB) and Fastboot on the laptop; We launched the command adb reboot bootloader in the terminal to enter bootloader mode. When then launching fastboot flashing unlock, the TV shows a black screen, and nothing happens; A factory reset should be performed after this procedure, but it is not, suggesting that something fails during the process.

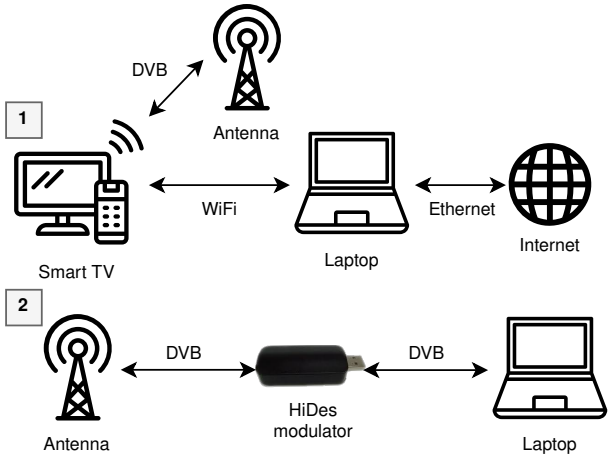

Fig. 1: Overview of our Testing Environment: (1) *On-TV HTTP(S) Traffic Capture*, (2) *Off-TV HTTPS Traffic Inspection*.

Rooting Attempt #3: Custom Recovery. The second way to root the device is to use a custom recovery. Unfortunately, no recovery image was to be found for our Smart TV for any custom recovery apps (e.g., Team Win Recovery Project (TWRP)). In this case, the bootloader also needs to be unlocked to carry out the procedure, making this option unviable.

Rooting Attempt #4: Xiaomi Procedure. Finally, we tried a procedure specific to Xiaomi Smart TVs. We first created a new Mi Cloud account and logged in to our device. According to the official Xiaomi documentation, any account receives one bootloader to unlock every 30 days for a maximum of four times a year [53]. We then downloaded the *Xiaomi Mi Unlock Tool* [54] on our laptop and ran it with admin privileges (note that the program is only available for Windows machines). We turned on our Xiaomi Smart TV in fastboot mode and connected it to the laptop. While the Mi Unlock Tool recognized the TV, tapping the "Unlock" button resulted in an error message stating that the procedure could not be completed successfully, and the TV rebooting.

B. Off-TV Experiments

Considering the challenges in gaining root access, we developed a second approach—the *Off-TV* experiment. Rooting a device is invasive and not always desired since it can potentially result in the device malfunctioning or being "bricked" in the worst case. Furthermore, not requiring a physical Smart TV reduces the barrier of entry for researchers interested in this area and thus increases the portability and replicability of our approach. The resulting dataset of HbbTV URLs we extract using this approach further increase flexibility and reproducibility—allowing the corresponding HbbTV

Listing 1: Parsed DVB Stream of the DMAX channel.

```
|=============================================================================|
|  Service: 0x0032 (50), TS: 0x0204 (516), Original Netw: 0x001D (29)         |
|  Service name: DMAX, provider: Persidera                                    |
|  Service type: 0x19 (Advanced codec HD digital television service)          |
|  TS packets: 33,154, PID's: 6 (clear: 6, scrambled: 0)                      |
|  PMT PID: 0x00FA (250), PCR PID: 0x04B0 (1200)                              |
|-----------------------------------------------------------------------------|
|     PID  Usage                                          Access      Bitrate |
|   Total  Advanced codec HD digital television service . C    3,919,148 b/s  |
|  0x00FA  PMT ........................................... C        4,965 b/s  |
|  0x04B0  AVC video (1920x1080, high profile, level 4.0  C    3,774,222 b/s  |
|  0x04B1  MPEG-1 Audio (ita, Audio layer II, 128 kb/s,   C      132,277 b/s  |
|  0x13ED  MPEG-2 Private sections (AIT)................. C        1,537 b/s  |
|  0x13F0  DSM-CC U-N (HbbTV) ........................... C        3,073 b/s  |
|  0x13F1  DSM-CC Stream Descriptors ..................... C        3,073 b/s  |
|          (C=Clear, S=Scrambled, +=Shared)                                   |
|=============================================================================|
```

applications to be tested by anyone from anywhere, with only potentially requiring a VPN to initiate a connection from a specific target country.

Extraction of HbbTV URLs. To extract HbbTV URLs from the DVB stream directly we use the HiDes modulator and the antenna described in Section IV. In addition, we use the TSDuck toolkit to parse the DVB stream [8].

We first scan the Ultra High Frequency (UHF) signal, looking for broadcast signals corresponding to the ones of the channels under analysis using the tsscan function of TSDuck. If found, we listen and capture the signal for 100 seconds on that UHF with the tsp function in Transport Stream (TS) file format. We then convert the file into text format for ease of reading and further processing. Listing 1 shows an example of a parsed DVB stream table. It shows the different sub-streams of the broadcast signal with their respective PIDs. It is important to note that, together with the standard A/V streams, also the Application Information Table (AIT) is present. Knowing that HbbTV information is contained in the AIT section, we extract the respective Program IDs associated with such section and convert them into XML format to make them human-readable and ready for further processing. We inspect the obtained files and extract the HbbTV application URLs. In Listing 2, we show the XML output for an AIT; The fields *url base* and *simple_application_location_descriptor initial_path* combined result in the HbbTV application URL(s).

With the *Off-TV* approach, we mainly faced two challenges: First, our antenna was not strong enough to capture broadcast signal for all channels, limiting our coverage. Second, some countries, e.g., Germany, only adopt cabled signals, while our testing infrastructure only supports for air-broadcasted signals. This limitation can be addressed by (1) adopting a stronger antenna and (2) by using a different modulator that does not only capture the DVB-T signal but also DVB-S.

Nevertheless, we successfully extracted HbbTV URLs from 24 out of 36 (66.6%) channels as listed in Table I. We also highlight that two thirds of the URLs adopt HTTP (18, 69.2%); this is true only for the start URL, as the HbbTV application will, in most cases, upgrade to HTTPS.

Testing of HbbTV URLs. In theory, we can open the extracted HbbTV URLs in any regular desktop browser, in our case Chrome (version 91.0.4459.2) on the laptop. In practice, we found the following two issues: (1) Some HbbTV applications detect that the User-Agent (UA) is not from a Smart TV.

Listing 2: Parsed AIT of the DMAX channel in XML format.

```
<AIT version="4" current="true" test_application_flag="false"
  application_type="0x0010" >
    <metadata PID="5,101"/>
    <application control_code="0x01">
      <application_identifier organization_id="0x0000001D"
                  application_id="0x0001"/>
      <transport_protocol_descriptor transport_protocol_label="0x00">
        <http>
          <url base="http://discovery.castoola.tv/" />
        </http>
      </transport_protocol_descriptor>
      <transport_protocol_descriptor transport_protocol_label="0x01">
        <object_carousel component_tag="0x01"/>
      </transport_protocol_descriptor>
      <application_descriptor service_bound="true" visibility="3"
                  application_priority="5">
        <profile application_profile="0x0000" version="1.2.1"/>
        <transport_protocol label="0x00"/>
        <transport_protocol label="0x01"/>
      </application_descriptor>
      <application_name_descriptor>
        <language code="eng" application_name="HbbTV - DMAX"/>
      </application_name_descriptor>
      <simple_application_location_descriptor initial_path="dmax" />
    </application>
</AIT>
```

Luckily, changing the UA is straightforward. We chose the UA of the Android TV we were testing as an example; However, several Smart TVs UAs can also be found online [19]. (2) HbbTV applications can also detect the Smart TV environment based on other features. In this case we use a Smart TV browser emulator, specifically the *RedOrbit HbbTV Emulator* [6] (v.0.7), to bind the key events (e.g., pressing the Red button), and simulate embedded broadcast signal which could be checked by an HbbTV application to understand whether it is running inside a Smart TV browser environment.

With this latter approach, we can bypass the encryption issues faced with our *On-TV* experiments when using the laptop to MITM connections and instead access plaintext traffic generated by the directly interacting with the HbbTV applications from our test laptop during our further tests detailed in the next section.

V. Simulating Consent & Collecting Data

Giving and Revoking Consent to Data Collection. We divide our traffic capture experiments into four main phases to spot different behaviors when different conditions are met (e.g., consent to data treatment is given). In fact, we expect different requests and collected data since data about the consumer should not be communicated before consent or after its revocation. The phases are as follows:

(1) **Before Consent**: During this phase, we expect no other request except for the initial one to retrieve the HbbTV application, as specified in the protocol standard. Channels should present the user with privacy policies containing information on what data is collected and how it is processed. Further, no tracking and advertisement domains should be contacted before the consumer explicitly consents to data treatment policies as defined by GDPR [3].

(2) **After Consent**: We then consent to privacy policies (in case channels do actually show any) and start the interaction with the HbbTV application. We manually explore as many functionalities as possible and check what information is exchanged.

TABLE I: HbbTV Application Start URLs for 24 channels that we extracted during our *Off-TV* experiments and whether they deploy HTPPS from the start (🔒). Note that channels might still upgrade to HTTPS later.

Country	Channel Name	HTTPS?	HbbTV Application Start URLs
Italy	Sportitalia		http://www.sportitalia.kbbtv.tech/hbbtv/sportitalia/sportitaliachannel/index.html
	RDS		http://hbbtv.rds.radio
	RealTime		http://discovery.castoola.tv/realtime
	RTL	🔒	https://cloud.rtl.it/hbbtv.rtl.it/rtlchannel/index.html
	Rai 1	🔒	https://www.raiplay.it/hbbtv/launcher/RemoteControl/index.html?delivery=2 https://www.raiplay.it/hbbtv/RaiPlay2020/index.html
	Spike		http://www.kbbtv.tech/viacom/viacomchannel/index.html
	Canale 5	🔒	http://hbbtv.mediaset.net/app/mplayhbbtvgold/backdoor.shtml http://hbbtv.mediaset.net/app/mplayhbbtvgoldzoo/dev/index.html https://mhptivu.mediaset.net/app/mplayhbbtvtivu/index.html https://tivuon-hbbtv-lativu.tivu-alchemy.net/index.html?configuration=prod
	La7	🔒	https://ht.la7.it/index.php
	Radio Kiss Kiss		http://www.kisskiss.kbbtv.tech/hbbtv/kisskisschannel/index.html
	Radio Libertà	🔒	https://hbbtv.persidera.it/hbbtv/jump/index.html?channelId=38893
	BOM Channel		http://95.110.225.170/hbbtv_bootstrap/index.php
	NOVE		http://discovery.castoola.tv/nove
	Caccia e Pesca	🔒	https://app.cacciaepesca.tv/hbbtv-cp/
	QVC		http://qvc-italy-hbbtv-app.qvc-italy.c.nmdn.net/redbutton
	TeleNordEst		http://hbbtv.tdbnet.it/run.php?pid=2049
	TeleChiara		http://iphd.it/hbbtv/telechiara/index.php
	SuperTennis	🔒	https://hbbtv.persidera.it/hbbtv/launcher/index.html?appId=25970
	LineaGem		http://www.grupposciscione.kbbtv.tech/hbbtv/lineagem/index.php
	Warner TV		http://it.container.enhanced.live/warnertv/
	TV 8	🔒	https://data-hip-gcdn-skycdn-it.akamaized.net/iapp/produzione/hbbtv/Addressable/index.html
Austria	ATV		http://hbbtv.prosiebensat1puls4.com/service/redbutton.php?brand=atvat
	ORF		http://orfhbbtv.orf.at/orf/newsportal/index.html
	Servus TV		http://hbbtv.servustv.com/content/themes/hbbtv-theme/1-0/dist/index.html?stvat-hd-t2
	RTL		http://rtl-digitaltext.rtl-hbbtv.de/launchbar/index-rtl-at.html

(3) **Revoke Consent**: We revoke consent to data treatment whenever possible. In this case, cookies must be deleted, and tracking must stop. If this does not happen, we flag it as a privacy violation. Note that some HbbTV applications do not allow for consent revocation, or consent to functional and technical cookies are always turned on.
(4) **Restore Consent**: Finally, we give consent again and turn off/on the Smart TV. We then check what cookies are set, e.g., if the same ones as in *After Consent* are used, meaning that either they were not deleted at all or are computed deterministically.

The testing guidelines and procedures we adopted can be replicated by other researchers for future research on HbbTV and more details can be found in our artifact repository.

Collection and Parsing of Network Traffic. We capture traffic in *.pcap* files and extract the following information using TShark [9]: The contacted domain name, the testing phase during it was contacted (*before-consent*, *after-consent*, *consent-revoked*, *consent-restored*) and the number of requests to that host. In addition, we check if the requests are made adopting plain HTTP or encryption (with TLS), if cookies are set, and what their expiration date is. We further check the returned object type (if any) and found a widely adopted technique to track and profile consumers, the *Tracking Pixel*, a quasi-invisible image which we mark as such.

Automation vs. Manual Effort. We automate as many tasks as possible to make our tests more efficient and scalable, but also to avoid imprecisions. For example, when capturing the traffic, we precisely time our four phases using a bash script to start and stop our running instances of Wireshark. However, we require manual work as well. Given the fact that every HbbTV application supports different interactions and set of buttons on the remote controller, we had to interact manually with them to explore all their functionalities. In theory, automation could be achieved by issuing Key Events over ADB by connecting the Smart TV to a laptop [13]; However, the different configurations and layout of the applications do not make this task trivial. Exploring applications with automatic inputs is an open challenge—one that we frequently encounter in related research on privacy issues in mobile apps [42], [40]. Despite considerable research in this direction [18], a still frequently used approach is testing with random user interface inputs [14], which, however, is not yet capable of

exhaustively triggering complex functionalities. Luckily in the HbbTV case the limited number of channels that need to be tested makes manual interaction feasible to exhaustively explore all functionality within a given testing time window.

The evaluation of the collected data also required manual effort. While it may seem straightforward to match the collected domains against well-known tracking denylists, i.e., PiHole [5] and EasyList [2] (versions from May 2022), we realized these lists are incomplete and do not contain tracking domains specific to the Smart TV ecosystem. Thus, we manually labeled the domains in our dataset to avoid false negatives. To do so, we searched for information about the domains and their owners and marked them as tracking or content providers.

VI. User Awareness Survey

Or survey has some characteristics typical of a *Positivist* and *Interpretivist* approach, as described by Palen [37]. On the one hand, we try to build knowledge given that we do not know a priori the level of awareness and what people consider a risk. Conversely, Ghiglieri published similar results (see Section VII), giving us possible clues on the outcome.

Ethics. After defining the questions in the questionnaire (see our NDSS'23 paper [49]), we considered the ethical aspects of our survey. We fully respected the ethical guidelines defined by the University of Twente (the university where we started this project at) and received approval from the ethical committee before sending the survey to participants. The university provides a self-assessment questionnaire for research that raises possible ethics concerns, which we followed.[3] In addition, on the first page of our survey, we report our contact information and explicitly state that participation is voluntary and that the survey can be stopped at any point.

Implementation. We built the survey using *SoSci Survey* [7], a German platform that allows heavy customization of questions. For example, it allowed us to display eight risky scenarios in a random order, different for each participant, and show questions only if certain pre-conditions were met. Furthermore, SoSciSurvey ensured that data was stored only on German servers, making the whole data processing subject to European data regulations, e.g., GDPR.

Methodology. We adopt a mixed-method design approach for our survey by including quantitative and qualitative methodologies. The quantitative part of this study is reflected in closed questions (multiple- or single-choice). We used these questions (e.g., age, whether participants own a Smart TV or not) to gather general statistics that can be easily summarized with mathematical and statistical functions such as average or standard deviation. On the other hand, we adopted a qualitative approach through open-ended questions; We aimed to simulate an interview by asking the participants to resonate and think about potential risks and their consequences. The semi-structured open-ended questions give the participants a structure for their answers.

[3]Procedures at the University of Twente's Ethical Board for Computer & Information Science: https://www.utwente.nl/en/eemcs/research/ethics/

One lesson learned from our survey is that participants often felt discouraged by its length and stopped before reaching the end (thus invalidating their answers). A progress bar showing how much of the questionnaire is left could be a starting point to increase participation.

Survey Size. Out of 817 people that received our survey, 21.3% (174) answered it, 75.9% (132) of which stated that they either possess a Smart TV or are willing to buy one. To ensure sufficient statistical power, we calculated the effective confidence interval with a confidence level of 95%. We consider as the total population the number of individuals in Italy over 18 years that watch TV (45.235.000 in 2021 [47], we do not expect much variation in 2022). We extended the size of our survey as part of the Major Revision requirements for our NDSS'23 paper, from originally 70 participants (12% confidence interval) by successfully collecting around 100 more responses (for a total of 174), decreasing the confidence interval to around 7%, hence improving our statistical power. This means that the reliability of our results to represent the whole population also significantly increased.

VII. Methods Borrowed From Prior Research

Ghiglieri et al. are the primary authors of HbbTV security and privacy aspects, but their studies date back to 2013–2015 [26], [25], [24]. Nevertheless, their approach is a good baseline, and following, we outline our main similarities:

TV Experiments. Our *On-TV* experiments adopt a similar approach to Ghiglieri et al.'s; We borrow the same division in the four testing phases and similar timings. However, while they only focused on German channels, we expand our analysis to four other countries as well as check the state of maturity of HbbTV adoption across them. Furthermore, since the studies of Ghiglieri et al. date back almost ten years, results are now outdated, considering the increasingly widespread use of HbbTV and the introduction of version 2.0 in 2015. One significant change in the deployment is encryption: Ghiglieri et al. observed mostly unencrypted traffic, while we noticed an increase in the adoption of HTTPS across the HbbTV-enabled channels. Therefore, in our case, the *On-TV* experiments did not suffice to spot potential misbehaviors of the broadcaster prompting us to design our second testing procedure, the *Off-TV* experiments (see Section IV).

Awareness Survey. The set of questions we define are similar to the ones posed by Ghiglieri et al., but with more focus on HbbTV and on the experience consumers have with this protocol. Furthermore, we sent the survey to Italian consumers while previous surveys were conducted in Germany.

Reproduction Aspects. Measurement studies in security and privacy tend to be "one-shot" and reproduction studies are limited. However, as new protocol versions and functionalities are introduced and in light of changes in adoption, study result might outdate quickly; In our case, measurement results from five years ago do not depict the current HbbTV landscape. We argue that there is value in revisiting and replicating prior studies, to (a) confirm the results or provide an updated view,

and (b) add new dimensions to the measurement, such as in our case extending the study to different countries. We hope by documenting and providing our testing methodology to the community, we enable further studies in this area. Based on the discussions at the LASER workshop, this does not only concern HbbTV, but could also be extended to other protocols deployed in other countries around the world.

VIII. Open Challenges and Future Work

Protocol and Country Coverage. As a first step in this direction, we plan to test more HbbTV-enabled countries and channels in the European landscape. However, we are also interested in expanding our study to similar protocols across the world, e.g., to Japan and the US. Thanks to feedback from the LASER Workshop, we were made aware of a protocol in Japan, *Hybridcast*, that combines broadcast and broadband content delivery to Smart TVs. This protocol works similarly to HbbTV, and adapting our tests should be straightforward, but we have yet to plan feasibility and logistics.

A possible solution to bypass budget and geographical limitations would be replicating the setup of Pavur et al. [39]. They use relatively cheap equipment (i.e., around EUR 350 or USD 384) to capture satellite signals (DVB-S) encompassing an area of over 100 million square kilometers. With their setup and by capturing Satellite instead not Terrestrial signals, we could theoretically reach broadcast signals from more countries without the need to travel and perform tests in loco, thus saving time and money.

Testing Infrastructure. While we argued that there are benefits to the *Off-TV* experiment setup we designed to bypass encryption, ideally, we would still like to provide a full testing infrastructure for *On-TV*. As detailed in Subsection IV-A, this hinges on our ability to successfully root a Smart TV to install a proxy CA certificate. However, such a setup would alleviate the limitations encountered when running the HbbTV application in a laptop's browser environment and allow us to more fully explore all the functionalities without any exceptions, as documented in Subsection IV-B.

Responsible Disclosure. We will further pursue our responsible disclosure process to make broadcasters aware of potential privacy and security threats in their HbbTV applications. We have, at the moment, contacted two national CERTs from Austria and the Netherlands and the Italian Garante della Privacy without any actionable feedback. We had a preliminary meeting with the Dutch organization where the issues we found were discussed, but due to their volunteer-based structure and thus limited time, no further steps could be taken. We will continue the the conversation and are also considering contacting CERTs in Germany, as our results also concern TV channels there. As a a last resort, we will directly notifying the affected broadcasters (and partially have done so again without feedback), even though we are still hoping for guidance and organizational support through a CERT or similar entity.

IX. Conclusions

During the experiments as part of our NDSS'23 paper [49] we observed the need for a portable and reliable pipeline for testing the deployment of the HbbTV protocol across different countries. We developed a two-fold approach to bypass geographical and hardware limitations, prompted by the need to bypass increased encryption when trying to replicate tests from previous research. Our approach fosters replicability and scalability and we hope to encourage further measurement studies in this area, both in terms of coverage of transmission protocols and country deployments.

Acknowledgements

We thank Martina Komsic for her help in testing Austrian and German TV channels and our contacts in Finland who helped us with the experiments there.

This research has been funded by the Vienna Science and Technology Fund (WWTF) and the City of Vienna [Grant ID: 10.47379/ICT19056], as well as SBA Research (SBA-K1), a COMET Center within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMAW, and the federal state of Vienna. The COMET Programme is managed by FFG.

References

[1] (2022) Android Security Bulletin—October 2020. Last accessed: 2023-03-16. [Online]. Available: https://source.android.com/docs/security/bulletin/2020-10-01

[2] (2022) EasyList. Last accessed: 2022-05-23. [Online]. Available: https://easylist.to/

[3] (2022) General Data Protection Regulation (GDPR). Last accessed: 2022-05-21. [Online]. Available: https://gdpr-info.eu/

[4] (2022) Metasploit. Last accessed: 2023-04-28. [Online]. Available: https://www.metasploit.com/

[5] (2022) Pi-hole. Last accessed: 2022-05-30. [Online]. Available: https://docs.pi-hole.net/

[6] (2022) RedOrbit HbbTV Emulator. Last accessed: 2022-11-29. [Online]. Available: https://chrome.google.com/webstore/detail/redorbit-hbbtv-emulator/mmgfafehampkahlmoahbjcjcmgmkppab?hl=en

[7] (2022) SoSci Survey. Last accessed: 2022-11-29. [Online]. Available: https://www.soscisurvey.de/

[8] (2022) TSDuck. Last accessed: 2022-11-29. [Online]. Available: https://tsduck.io/

[9] (2022) TShark. Last accessed: 2022-05-30. [Online]. Available: https://www.wireshark.org/docs/man-pages/tshark.html

[10] (2022) Wireshark. Last accessed: 2022-05-30. [Online]. Available: https://www.wireshark.org/

[11] (2023) Magisk. Last accessed: 2023-03-16. [Online]. Available: https://github.com/topjohnwu/Magisk

[12] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, "The Web Never Forgets: Persistent Tracking Mechanisms in the Wild," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014.

[13] Android Developers. (2023) Android KeyEvents. Last accessed: 2023-04-28. [Online]. Available: https://developer.android.com/reference/android/view/KeyEvent

[14] ——. (2023) UI/Application Exerciser Monkey. Last accessed: 2023-04-28. [Online]. Available: https://developer.android.com/studio/test/other-testing-tools/monkey

[15] Better Software Group. (2019) HbbTV: What Is it and How Does it Work? Last accessed: 2021-02-17. [Online]. Available: https://bsgroup.eu/hbbtv-what-is-it-and-how-does-it-work/

[16] J. Brookman, P. Rouge, A. Alva, and C. Yeung, "Cross-Device Tracking: Measurement and Disclosures," in *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2017.

[17] F. Chanchary and S. Chiasson, "User Perceptions of Sharing, Advertising, and Tracking," in *Proceedings of the USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2015.

[18] S. R. Choudhary, A. Gorla, and A. Orso, "Automated Test Input Generation for Android: Are We There Yet?" in *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2015.

[19] DeviceAtlas. (2022) List of smart TV User-Agent strings. Last accessed: 2023-04-28. [Online]. Available: https://deviceatlas.com/blog/list-smart-tv-user-agent-strings

[20] S. Englehardt and A. Narayanan, "Online Tracking: A 1-Million-Site Measurement and Analysis," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016.

[21] European Telecommunications Standards Institute (ETSI), "ETSI TS 102 796 V1.5.1, Hybrid Broadcast Broadband TV," Standard, Sep. 2018. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.05.01_60/ts_102796v010501p.pdf

[22] A. P. Felt, S. Egelman, and D. Wagner, "I've Got 99 Problems, but Vibration Ain't One: A Survey of Smartphone Users' Concerns," in *Proceedings of the ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*, 2012.

[23] I. Fouad, N. Bielova, A. Legout, and N. Sarafijanovic-Djukic, "Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels," in *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2020.

[24] M. Ghiglieri, "I Know What You Watched Last Sunday - A New Survey Of Privacy In HbbTV," in *Proceedings of the IEEE Web 2.0 Security & Privacy Workshop (W2SP)*, 2014.

[25] M. Ghiglieri and E. Tews, "A Privacy Protection System for HbbTV in Smart TVs," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC)*, 2014.

[26] M. Ghiglieri and M. Waidner, "HbbTV Security and Privacy: Issues and Challenges," *IEEE Security & Privacy*, vol. 14, no. 3, 2016.

[27] HbbTV Association. (2022) HbbTV Overview. Last accessed: 2022-05-26. [Online]. Available: https://www.hbbtv.org/overview/

[28] ——. (2023) Press Release: Deployment of HbbTV Targeted Advertising boosted at 2022 Symposium. Last accessed: 2023-03-16 (Archived: https://archive.is/rman3). [Online]. Available: https://www.hbbtv.org/news-events/deployment-of-hbbtv-targeted-advertising-boosted-at-2022-symposium/

[29] U. Iqbal, S. Englehardt, and Z. Shafiq, "Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021.

[30] K. Kollnig, P. Dewitte, M. V. Kleek, G. Wang, D. Omeiza, H. Webb, and N. Shadbolt, "A Fait Accompli? An Empirical Study into the Absence of Consent to Third-Party Tracking in Android Apps," in *Proceedings of the USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2021.

[31] K. Kollnig, A. Shuba, R. Binns, M. V. Kleek, and N. Shadbolt, "Are iPhones Really Better for Privacy? A Comparative Study of iOS and Android Apps," in *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2022.

[32] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner, "Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016," in *Proceedings of the USENIX Security Symposium*, 2016.

[33] W. Meng, R. Ding, S. P. Chung, S. Han, and W. Lee, "The Price of Free: Privacy Leakage in Personalized Mobile In-Apps Ads," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2016.

[34] H. Mohajeri Moghaddam, G. Acar, B. Burgess, A. Mathur, D. Y. Huang, N. Feamster, E. W. Felten, P. Mittal, and A. Narayanan, "Watching You Watch: The Tracking Ecosystem of Over-the-Top TV Streaming Devices," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2019.

[35] P. Nema, P. Anthonysamy, N. Taft, and S. T. Peddinti, "Analyzing User Perspectives on Mobile App Privacy at Scale," in *Proceedings of the International Conference on Software Engineering (ICSE)*, 2022.

[36] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2013.

[37] L. Palen, "Empirical Epistemologies Applied to Human-Centered Computing Research: A One Page Guide," University of Colorado Boulder, Tech. Rep., 2014.

[38] E. Pan, J. Ren, M. Lindorfer, C. Wilson, and D. Choffnes, "Panoptispy: Characterizing Audio and Video Exfiltration from Android Applications," in *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2018.

[39] J. Pavur, D. Moser, V. Lenders, and I. Martinovic, "Secrets in the Sky: On Privacy and Infrastructure Security in DVB-S Satellite Broadband," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019.

[40] A. Pradeep, T. M. Paracha, P. Bhomwick, A. Davanian, A. Razaghpanah, T. Chung, M. Lindorfer, N. Vallina-Rodriguez, D. Levin, and D. Choffnes, "A Comparative Analysis of Certificate Pinning in Android & iOS," in *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2022.

[41] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill, "Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018.

[42] J. Ren, M. Lindorfer, D. Dubois, A. Rao, D. Choffnes, and N. Vallina-Rodriguez, "Bug Fixes, Improvements, ... and Privacy Leaks – A Longitudinal Study of PII Leaks Across Android App Versions," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2018.

[43] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes, "ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic," in *Proceedings of the International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2016.

[44] I. Shklovski, S. D. Mainwaring, H. H. Skúladóttir, and H. Borgthorsson, "Leakiness and Creepiness in App Space: Perceptions of Privacy and Mobile App Use," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2014.

[45] Statista. (2021) Daily TV viewing time in european countries. Last accessed: 2022-05-21. [Online]. Available: https://www.statista.com/statistics/422719/tv-daily-viewing-time-europe/

[46] ——. (2022) Average daily time spent watching TV in the United States from 2019 to 2023. Last accessed: 2022-05-21. [Online]. Available: https://www.statista.com/statistics/186833/average-television-use-per-person-in-the-us-since-2002/

[47] ——. (2022) Number of individuals watching TV in Italy in 2021, by age. Last accessed: 2023-04-28. [Online]. Available: https://www.statista.com/statistics/539635/television-viewers-by-age-italy/

[48] ——. (2022) Number of TV households worldwide from 2010 to 2026. Last accessed: 2022-05-19. [Online]. Available: https://www.statista.com/statistics/268695/number-of-tv-households-worldwide/

[49] C. Tagliaro, F. Hahn, R. Sepe, A. Aceti, and M. Lindorfer, "I Still Know What You Watched Last Sunday: Privacy of the HbbTV Protocol in the European Smart TV Landscape," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2023.

[50] M. Tileria and J. Blasco, "Watch Over Your TV: A Security and Privacy Analysis of the Android TV Ecosystem," in *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, 2022.

[51] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang, "Smart, Useful, Scary, Creepy: Perceptions of Online Behavioral Advertising," in *Proceedings of the USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2012.

[52] B. Weinshel, M. Wei, M. Mondal, E. Choi, S. Shan, C. Dolin, M. L. Mazurek, and B. Ur, "Oh, the Places You've Been! User Reactions to Longitudinal Transparency About Third-Party Web Tracking and Inferencing," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

[53] Xiaomi. (2022) Complete Guide to Unlocking the Bootloader: Announcement, Troubleshoot and Tips. Last accessed: 2023-04-28 (Archived: https://archive.is/XzXo1). [Online]. Available: https://new.c.mi.com/global/post/101245?utm_source=miui&utm_medium=official_web_faq&utm_campaign=official_web_miui

[54] ——. (2023) Mi Unlock. Last accessed: 2023-03-16. [Online]. Available: https://en.miui.com/unlock/download_en.html

[55] S. Zimmeck, J. S. Li, H. Kim, S. M. Bellovin, and T. Jebara, "A Privacy Analysis of Cross-device Tracking," in *Proceedings of the USENIX Security Symposium*, 2017.