

# Challenges of Coordinated Linux & Android Intrusions



IMF 2014  
Eoghan Casey

May 12, 2014



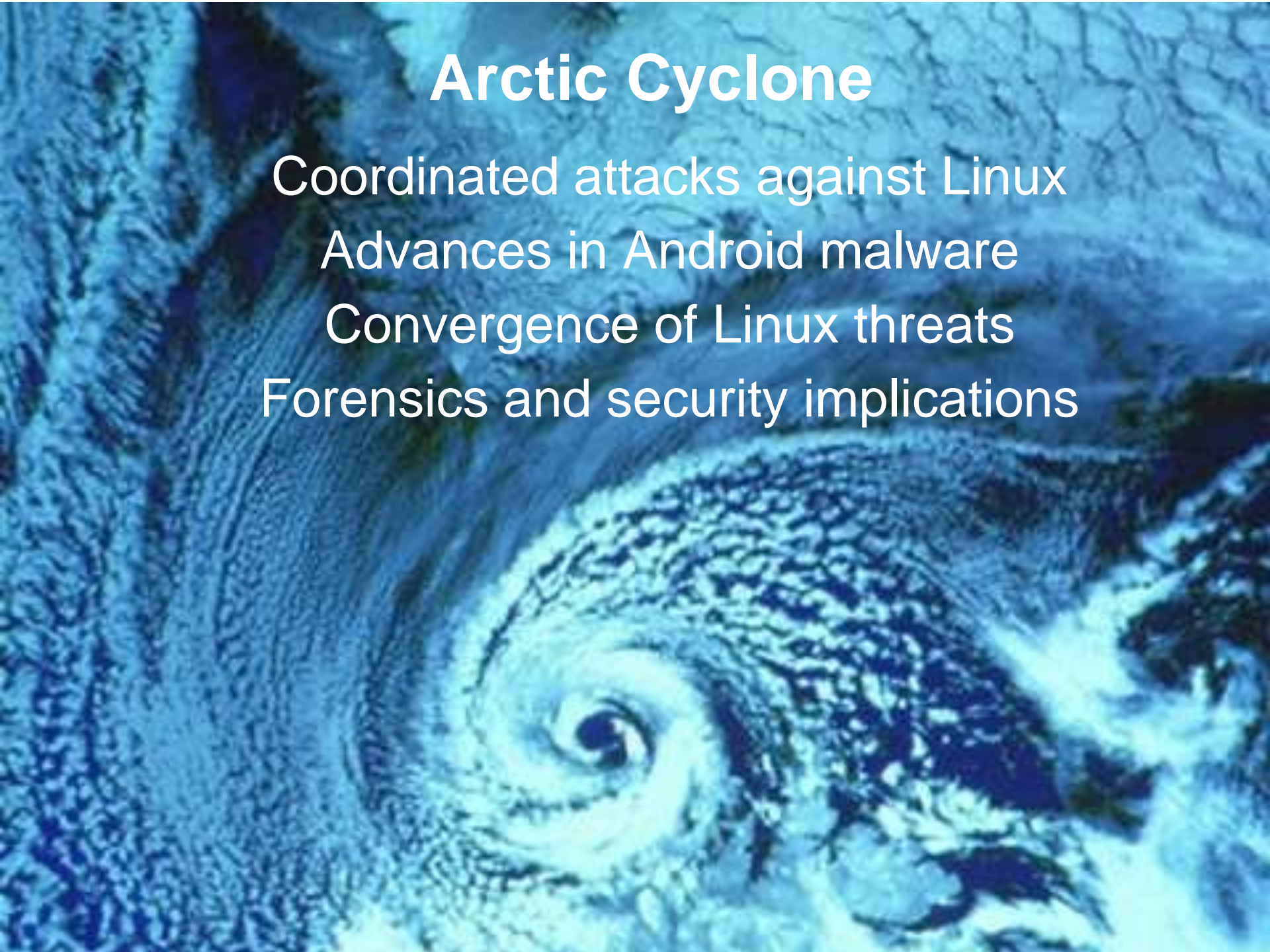
# Arctic Cyclone

Coordinated attacks against Linux

Advances in Android malware

Convergence of Linux threats

Forensics and security implications



# Coordinated Linux Intrusions

2008 - Present

**The Register**<sup>®</sup>

Kernel.org Linux repository rooted in hack attack

**Rootkit not detected for 17 days**

By [Dan Goodin in San Francisco](#) • [Get more from this author](#)

Posted in [Enterprise Security](#), 31st August 2011 22:35 GMT

**Updated** Multiple servers used to maintain and distribute the Linux operating system were infected with malware that gained root access, modified system software, and logged passwords and transactions of the people who used them, the official Linux Kernel Organization has confirmed.



**US-CERT**

UNITED STATES COMPUTER EMERGENCY READINESS TEAM

## 08.26.2008 - Current Activity



### SSH Key-based Attacks

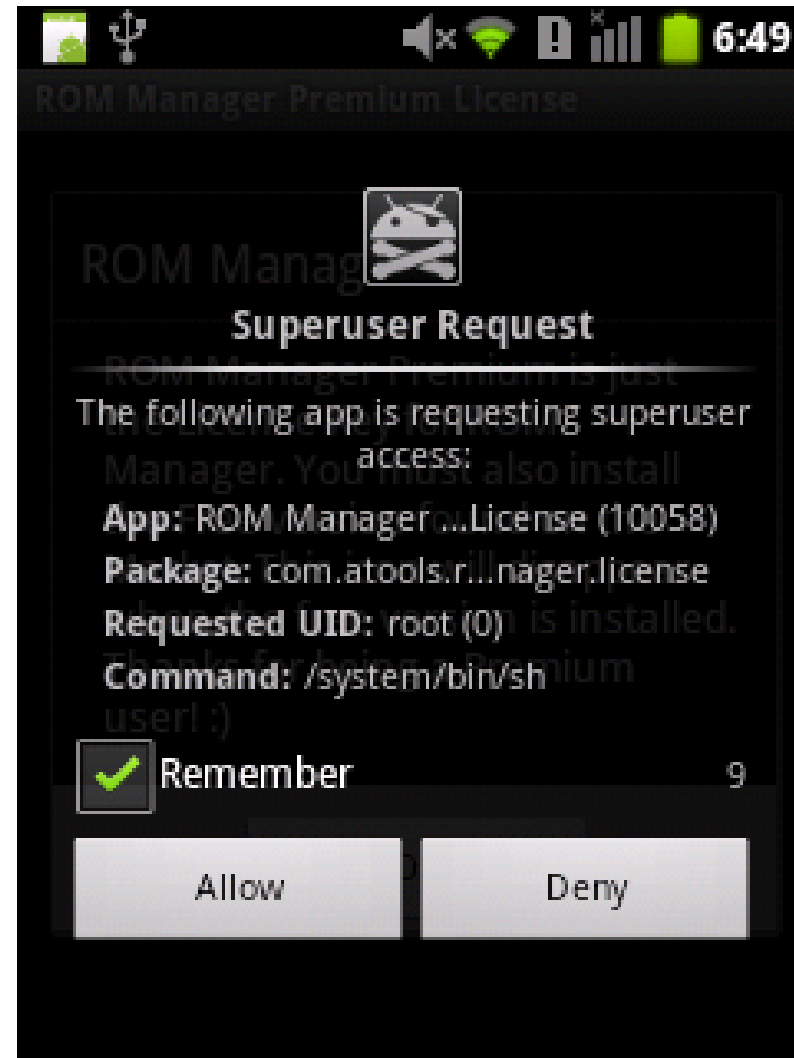
*added August 26, 2008 at 03:41 pm*

US-CERT is aware of active attacks against linux-based computing infrastructures using compromised SSH keys. The attack appears to initially use stolen SSH keys to gain access to a system, and then uses local kernel exploits to gain root access. Once root access has been obtained, a rootkit known as "phalanx2" is installed.

Phalanx2 appears to be a derivative of an older rootkit named "phalanx". Phalanx2 and the support scripts within the rootkit, are configured to systematically steal SSH keys from the compromised system. These SSH keys are sent to the attackers, who then use them to try to compromise other sites and other systems of interest at the attacked site.

# Android Malware

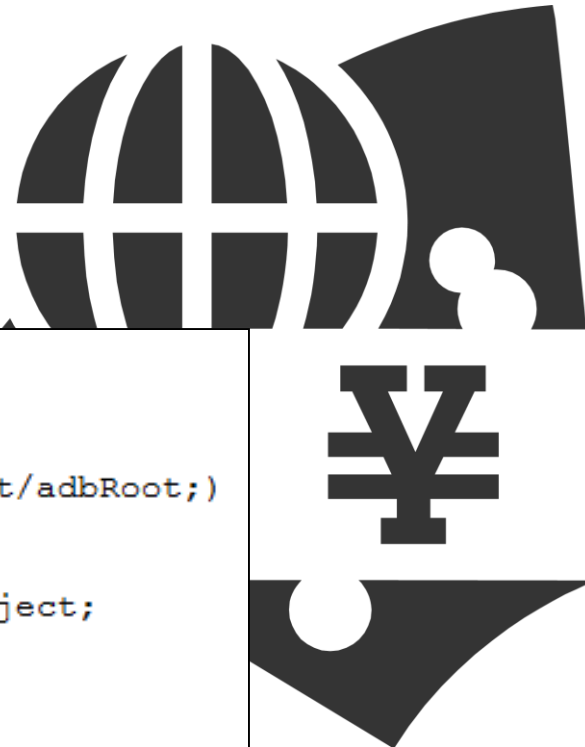
- Undermine the OS
- Steal information
- Download other malware
- DroidDream, DKFBootkit
- Added potential
  - Conversation eavesdropping
  - Geolocation tracking
  - Video surveillance



# Example: DroidDream

- Targeting legitimate application developers
  - Embed malicious code within their applications
- Broad capabilities
  - Root the operating system
  - Exfiltrate IMEI and IMSI
  - Download additional malware

```
d a |write ([BII)V
t u
d v  access$0  5(Lcom/android/root/adbRoot;)
Landroid/content/Context;  access$1  0(Lcom/android/root/adbRoot;)
Landroid/os/Handler; |crypt |([B)V  getIMEI  -
(Landroid/content/Context;)Ljava/lang/String; |phone
€ †getSystemService  &(Ljava/lang/String;)Ljava/lang/Object;
, f
H „ "android/telephony/TelephonyManager †
getDeviceId  ¶()Ljava/lang/String;
^ %
† §  access$  getIMSI  %getSubscriberId
```





# Advanced and Persistent...

Attacker's modus operandi

- Repository of stolen SSH credentials
- Privilege escalation
- LKM rootkits with port knocking backdoor
- Trojanized SSH daemon
- Resilient C2 and exfiltration
- Destroy digital evidence

# Stolen Credentials & Getting Root

- Rely on users re-using keys/passwords
  - Try stolen credentials on other Linux systems
  - Intruders have returned years after initial breach
- Escalate privileges
  - Weak passwords (zero day exploits only if needed)
- Rinse and repeat
  - Grab SSH related information for all users on host
    - known\_hosts, authorized\_keys, .bash\_history
    - usernames, hostnames, IP, passwords, keys
  - Stolen information added to attacker repository
  - Use stolen information to attack other Linux systems

# Advanced Rootkits and Backdoors

## Phalanx2

- Injects or loads into the memory and hides
- Disables audit subsystems
- Uses port knocking backdoor
- Sniffs TTY sessions for passwords
  - Interesting interception technique



# Trojanized SSH and Exfiltration

- Stores captures SSH credentials in RAM
- Automatically sends stolen data to C2 node
- Provides backdoor access
  - Secret handshake to access backdoor
  - Bypasses logging
- Has backup C2/exfiltration method
  - In case default is blocked
  - Falls back to crazy DNS lookup scheme

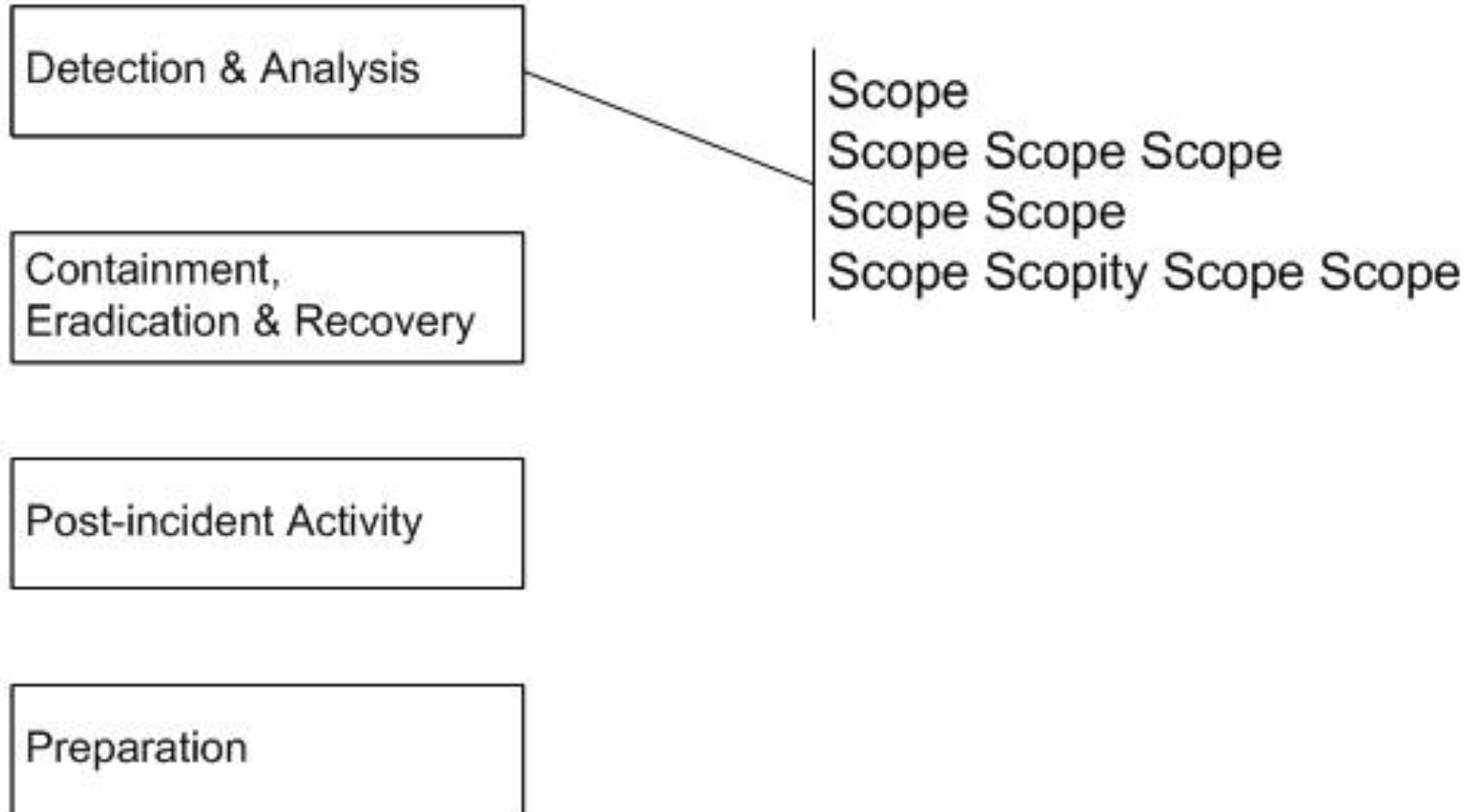
# Quick Containment?

- Current recommendation:

When an incident has been detected and analyzed, it is important to contain it before the spread of the incident overwhelms resources or the damage increases. Most incidents require containment, so it is important to consider it early in the course of handling each incident.

- NIST SP800-61 Rev. 1, page 3-19

# Managing a data breach effectively



# Effective Eradication of Intruders

Detection & Analysis

Containment,  
Eradication & Recovery

Post-incident Activity

Preparation

## Classic containment/eradication:

- Block bad IPs
- Block bad DNS host names
- Reset compromised credentials
- Nuke and pave or clean compromised hosts

## But also...

- Coordinate multiple actions to a single event
- Take evasive action (Ex. Change critical account names and add decoys)
- Restrict policies (Ex. SeDebugPrivilege)
- Establish internal perimeters

# Common Incident Response Mistakes

## 1) Underestimating the adversary

- Too quick to containment

## 2) Lack of evidence

- No centralized logging or backup infrastructure

## 3) Improper evidence handling

- Update antivirus & scan compromised systems

# Linux/Android Incident Response

- Linux & Android incident response process
  - Collect volatile data
  - Forensic examination of Linux memory
  - Forensic examination of EXT file system
  - Malware forensics
- Linux & Android Memory Extraction
  - Johannes Stüttgen (LMAP)
  - Joe Sylve (LiME)

```
# insmod /sdcard/lime.ko path=tcp:6666
```

OR

```
# insmod /sdcard/lime.ko path=/sdcard
```

# Know the Adversary

- Initial intrusions not necessarily sophisticated
  - Spear phishing or vulnerable servers
- Once inside, they spread virulently
- Inside out attacks circumvent egress filtering
- Undermine security monitoring
  - File system tampering
  - Multiple malware versions with custom packing
  - Blend in with normal traffic
  - Encrypt command, control and exfiltration



# Linux Memory Forensics

- Volatility and Rekall
  - Malware detection modules
  - Extracts memory structures

```
% python vol.py -f Phlananx2 linux_check_syscall
```

Table Name	Index	Address	Symbol
64bit	0x0	0xfffffffffa0059000	HOOKED
64bit	0x1	0xfffffffffa0062000	HOOKED
64bit	0x2	0xfffffffffa0035000	HOOKED
64bit	0x3	0xfffffffff81115351	sys_close
64bit	0x4	0xfffffffffa00cb000	HOOKED
64bit	0x5	0xfffffffff8111aa73	sys_newfstat
64bit	0x6	0xfffffffffa00b5000	HOOKED
64bit	0x7	0xfffffffff81126170	sys_poll

<edited for length>

# Linux Memory Forensics

- SecondLook
  - Alerts on unknown kernel modules
  - Extracts memory structures

Analysis | Alerts | Information | Disassembly | Data |

Analysis of the target generated 54 alerts. Click an alert for more information.

Kernel text/rodata mismatch at 0xffffffff816003e0 [sys\_call\_table+0]

Kernel module 'vmci': missing reference module

Kernel module 'vsock': missing reference module

Kernel module 'vmhgs': missing reference module

Return address in non-text memory region in kernel stack trace of pid 3451 (bash): 0xffffffffa005905c [shpchp: \_\_key.28464-

Return address in non-text memory region in kernel stack trace of pid 3444 (sshd): 0xffffffffa005905c [shpchp: \_\_key.28464-

Return address in non-text memory region in kernel stack trace of pid 2848 (bash): 0xffffffffa005905c [shpchp: \_\_key.28464-

Return address in non-text memory region in kernel stack trace of pid 2841 (sshd): 0xffffffffa005905c [shpchp: \_\_key.28464-

Return address in non-text memory region in kernel stack trace of pid 2720 (sshd): 0xffffffffa005905c [shpchp: \_\_key.28464-

Return address in non-text memory region in kernel stack trace of pid 2558 (sshd): 0xffffffffa005905c [shpchp: \_\_key.28464-

Return address in non-text memory region in kernel stack trace of pid 1060 (sedispatch): 0xffffffffa005905c [shpchp: \_\_key.2

Executable mapping in task Xnest (pid 2479) of [stack] is not read-only

Executable mapping in task Xnest (pid 2479) of anonymous memory is not read-only

Executable mapping in task Xnest (pid 2479) of anonymous memory is not read-only

Executable mapping in task Xnest (pid 2479) of anonymous memory is not read-only

Executable mapping in task Xnest (pid 2479) of file /usr/share/ /p-2.5f is not read-only

System call table entry 0 does not match reference kernel entry

System call table entry 1 does not match reference kernel entry

System call table entry 2 does not match reference kernel entry

System call table entry 4 does not match reference kernel entry

# Android Memory Forensics

- Examination of Android physical memory
  - Volatility plugin for Android memory

```
root@newubuntu: ~/volarm

root@newubuntu:~/volarm# python volatility.py --profile=android -f /mnt/data/volimgs/android-full linux_t
Volatile Systems Volatility Framework 1.4_rc1
Name                Pid      Uid
init                1        0
kthreadd            2        0
ksoftirqd/0        3        0
watchdog/0         4        0
events/0           5        0
khelper            6        0
async/mgr          7        0
suspend
sync_supers
bdi-default
kblockd/0
kmmcd
bluetooth
kondemand/0
smd_tty
qmi

root@newubuntu: ~/volarm

root@newubuntu:~/volarm# python volatility.py --profile=android -f /mnt/data/volimgs/android-full linux_t
Volatile Systems Volatility Framework 1.4_rc1
/dev/block/mtdblock4    /system    yaffs2
sysfs                   /sys       sysfs
devpts                  /dev/pts   devpts
/dev/block/dm-1         /mnt/asec/com.rovio.angrybirds-1 vfat
proc                    /proc      proc
none                    /dev/cpuctl cgroup
tmpfs                   /mnt/sdcard/.android_secure tmpfs
```

# Android File System Forensics

Android - Autopsy 3.0.0b2

File Edit View Tools Window Help

Directory Tree File Search

workshop-scenario-userdata-backup.dd

- \$OrphanFiles
- anr
- app
- app-private
- backup
- battd
- dalvik-cache
- data
  - Hurricane.Software
  - android.tts
  - blur.res
  - com.adobe.flashplayer
  - com.amazon.kindle
  - com.amazon.mp3
  - com.android.batteryreport
  - com.android.bluetooth
  - com.android.browser
  - com.android.calculator2
  - com.android.calendar
  - com.android.certinstaller
  - com.android.defcontainer
  - com.android.htmlviewer
  - com.android.inputmethod.latin
  - com.android.magicsmoke

Directory Listing

Table View Thumbnail View

workshop-scenario-userdata-backup.dd\app

Name	Modified Time	Changed Time	Access Time
com.magic.spiral-1.apk	2011-05-27 14:28:28	2011-05-27 14:28:28	2011-05-27 14:28:28
com.noshufou.android.su-1.apk	2011-02-21 14:15:02	2011-02-21 14:15:03	2011-02-21 14:15:02
com.stkiconcepts.hurricaneHound.free-1.apk	2011-08-27 11:05:52	2011-08-27 11:05:52	2011-08-27 11:05:52
com.weather.Weather-1.apk	2011-08-27 11:06:24	2011-08-27 11:06:27	2011-08-27 11:06:24
com.z4mod.z4root-1.apk	2011-02-21 14:13:20	2011-02-21 14:13:20	2011-02-21 14:13:20
example.helloandroid-1.apk	2011-09-01 16:07:50	2011-09-01 16:07:50	2011-09-01 15:59:50
jackpal.androidterm-1.apk	2011-05-24 17:03:29	2011-05-24 17:03:30	2011-05-24 17:03:29
oxygen.agent-1.apk	2011-06-11 15:12:12	2011-06-11 15:12:12	2011-06-11 15:12:12
stericson.busybox-1.apk	2011-05-24 17:05:40	2011-05-24 17:05:40	2011-05-24 17:05:40
vmdl50623.tmp	2011-09-01 16:07:50	2011-09-01 16:07:50	2011-09-01 15:59:50

workshop-scenario-userdata-backup.dd\app\jackpal.androidterm-1.apk

Hex View Picture View String View

Page: 1 of 9 Page

0x000000:	50	4B	03	04	14	00	08	00	08	00	68	7E	B8	3E	AE	08	PK.....
0x000010:	68	5A	0B	02	00	00	49	04	00	00	14	00	00	00	4D	45	hZ....I...
0x000020:	54	41	2D	49	4E	46	2F	4D	41	4E	49	46	45	53	54	2E	TA-INF/MAN
0x000030:	4D	46	AD	D3	4D	6F	DA	30	18	00	E0	3B	12	FF	FD	E3	MF..Mo.0..
0x000040:	26	44	4C	28	49	4B	A5	1D	02	0D	19	1D	24	10	5A	48	&DL(IK....
0x000050:	76	41	7D	ED	A4	A6	76	12	EC	40	62	7E	FD	40	FD	B4	vA....v..@
0x000060:	16	D1	69	21	DE	6C	CB	7E	F4	FA	FD	70	61	46	13	22	..i..l.~..
0x000070:	CB	CF	60	08	49	E3	EC	BF	25	6B	DD	66	63	24	08	2C	T 1



# COTS Android File System Forensics

The screenshot displays a forensic analysis tool interface with a 'Project Tree' on the left and a main 'Extraction Summary' window on the right. The 'Project Tree' shows a hierarchy for 'mtd8', including 'Extraction Summary', 'Device Info', 'Images' (with 'mtd8.dd'), 'Memory Ranges', 'File Systems' (with 'Image' and '\$DeletedNodes'), 'Analyzed Data' (with 'Chats (3)', 'Contacts (9)', 'Emails (65)', 'Locations (28)', 'SMS Messages (17)', 'User Accounts (2)', 'Web Bookmarks (50)', 'Bookmarks (0)', 'Data files' (with 'Images (645)', 'Videos', 'Audio', 'Text (692)', 'Tags', 'Reports'), and 'Device Content'.

The 'Extraction Summary' window has tabs for 'Welcome' and 'Extraction Summary'. It features a 'Device Information' section with a mobile phone icon and the text 'mtd8 Binary Dump'. Below this is an 'Image Hash Information' section with a warning icon and the message: 'No reference hash information is available for this project.' The 'Device Content' section is titled 'Phone Data' and contains a grid of data categories with their respective counts and status (0):

Category	Count	Status
Chats	3	(0)
Contacts	9	(0)
Emails	65	(0)
Locations	28	(0)
SMS Messages	17	(0)
User Accounts	2	(0)
Web Bookmarks	50	(0)

On the right side of the interface, there is a 'swift' window showing a list of contacts and emails. The 'Contacts' list includes 'swiftlogicllc@consultant.com (1 entr...)', 'swiftlogicinc@consultant.com (1 entr...)', and 'Swift Logic (1 entry, 0 addresses, 0 n...'. The 'Emails' list includes '"" <swiftlogic@consultant.com>, ""...', '"Mail Delivery Subsystem" <mailer-d...', and '"Mail Delivery Subsystem" <mailer-d...'. The 'Chats' list includes 'Chat: hersolv@gmail.com' and 'yobtaog@gmail.com: Hey, may make enough b'.

# File System Acquisition of Android

- Smartphone forensics
  - Bootloaders to bypass locked devices
  - JTAG to access hardware
- Rooted devices can be acquired natively

```
mre$ ./adb shell
$ su
```

```
# dd if=/dev/block/userdata bs=1024 |
/system/bin/busybox nc 192.168.2.2 755
7028736+0 records in
7028736+0 records out
7197425664 bytes transferred in 24211.203 secs
```

# Remote Android Acquisition

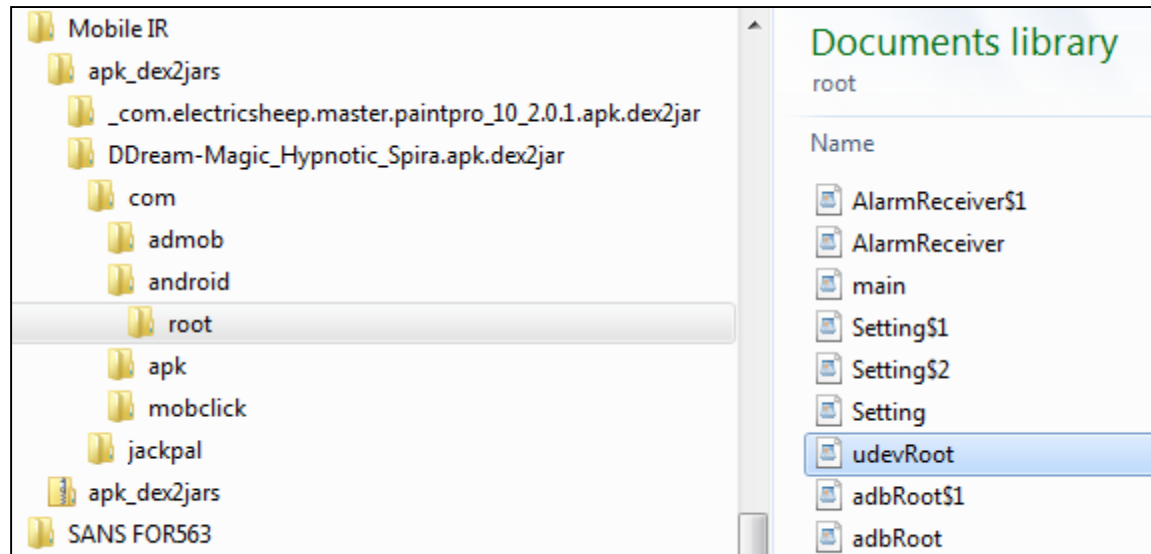
- F-Response
  - ARM agent
  - On SDcard
- GRR...?

```
# ./f-response-ce-e-android -c ./fresponse.ini
F-Response Consultant/Enterprise(Android/ARM Edition) Version 3.09.08
F-Response Disk: /dev/mtd/mtd0 (26624 sectors, 512 sector size)
13 MB write blocked storage on F-Response Disk:mtd0
F-Response Disk: /dev/mtd/mtd1 (5120 sectors, 512 sector size)
2 MB write blocked storage on F-Response Disk:mtd1
F-Response Disk: /dev/mtd/mtd2 (640 sectors, 512 sector size)
0 MB write blocked storage on F-Response Disk:mtd2
F-Response Disk: /dev/mtd/mtd3 (128 sectors, 512 sector size)
0 MB write blocked storage on F-Response Disk:mtd3
F-Response Disk: /dev/mtd/mtd4 (128 sectors, 512 sector size)
0 MB write blocked storage on F-Response Disk:mtd4
F-Response Disk: /dev/mtd/mtd5 (128 sectors, 512 sector size)
0 MB write blocked storage on F-Response Disk:mtd5
F-Response Disk: /dev/mtd/mtd6 (6144 sectors, 512 sector size)
3 MB write blocked storage on F-Response Disk:mtd6
F-Response Disk: /dev/mtd/mtd7 (614400 sectors, 512 sector size)
300 MB write blocked storage on F-Response Disk:mtd7
F-Response Disk: /dev/mtd/mtd8 (12288 sectors, 512 sector size)
6 MB write blocked storage on F-Response Disk:mtd8
F-Response Disk: /dev/mtd/mtd9 (3561472 sectors, 512 sector size)
1739 MB write blocked storage on F-Response Disk:mtd9
█
```



# Android Malware Analysis

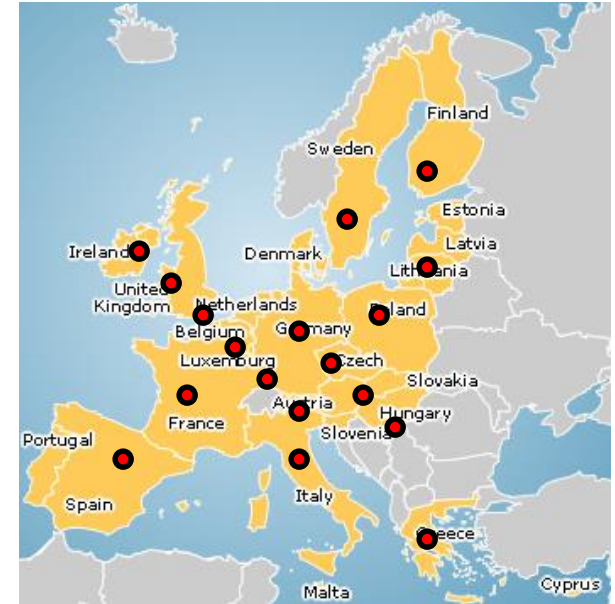
- DroidDream
  - Root exploit
  - Data theft
  - Updates



```
2 | com/android/root/udevRoot | java/lang/Object |
udevRoot.java
BUFFER_SIZE | I |
| FNAME_EXPLOIT | Ljava/lang/String; | exploit
| FNAME_REMOUNT_DATA_RO | remount_data.sh | | FNAME_REMOUNT_SYS_RO | remount_sys_ro.sh
| FNAME_REMOUNT_SYS_RW | remount_sys_rw.sh |
FNAME_SU_BIN
profile | | MOUNT_EXEC_PATH | /system/bin/mount | | ROOT_SHELL_PATH | /data/local/tmp/
rootshell
SU_EXEC_PATH | /system/bin/profile | TAG | UDevRoot | #
bDisableWifi | Z | ctx | Landroid/content/Context;
remountSysRW
wifiManager | Landroid/net/wifi/WifiManager; | -<init> | (Landroid/content/Context;)V |
() V
```

# Cross Border Information Sharing

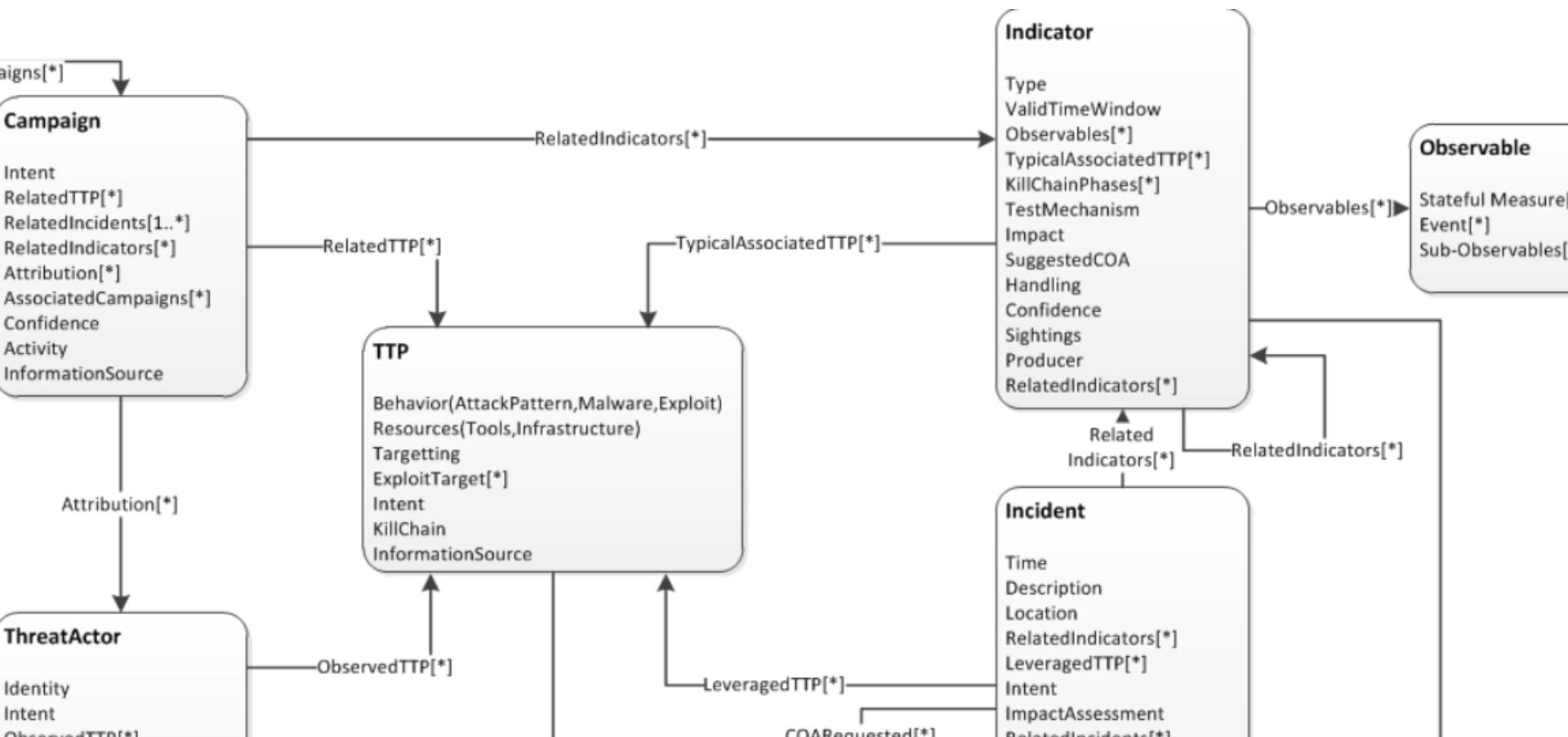
Same attackers targeting  
all EU member states >



- Consolidate adversary knowledge
- Trust between government and industry
- Confidentiality agreements
- More information to examine the better
- Sanitize what is shared to protect victims

# Information Exchange Standards

## STIX – Structured Threat Information eXpression



# Looking Ahead

- Linux and Android forensics R&D
  - Current tools are limited
- Linux and Android malware IOCs
  - Organizations don't know what to look for (detect)
- Linux and Android forensic analysts
  - Current expertise is lacking in this area
- Managing complexity
  - Web applications, databases, distributed storage
- Expand information exchange
  - EU-CERT, Europol, GRID