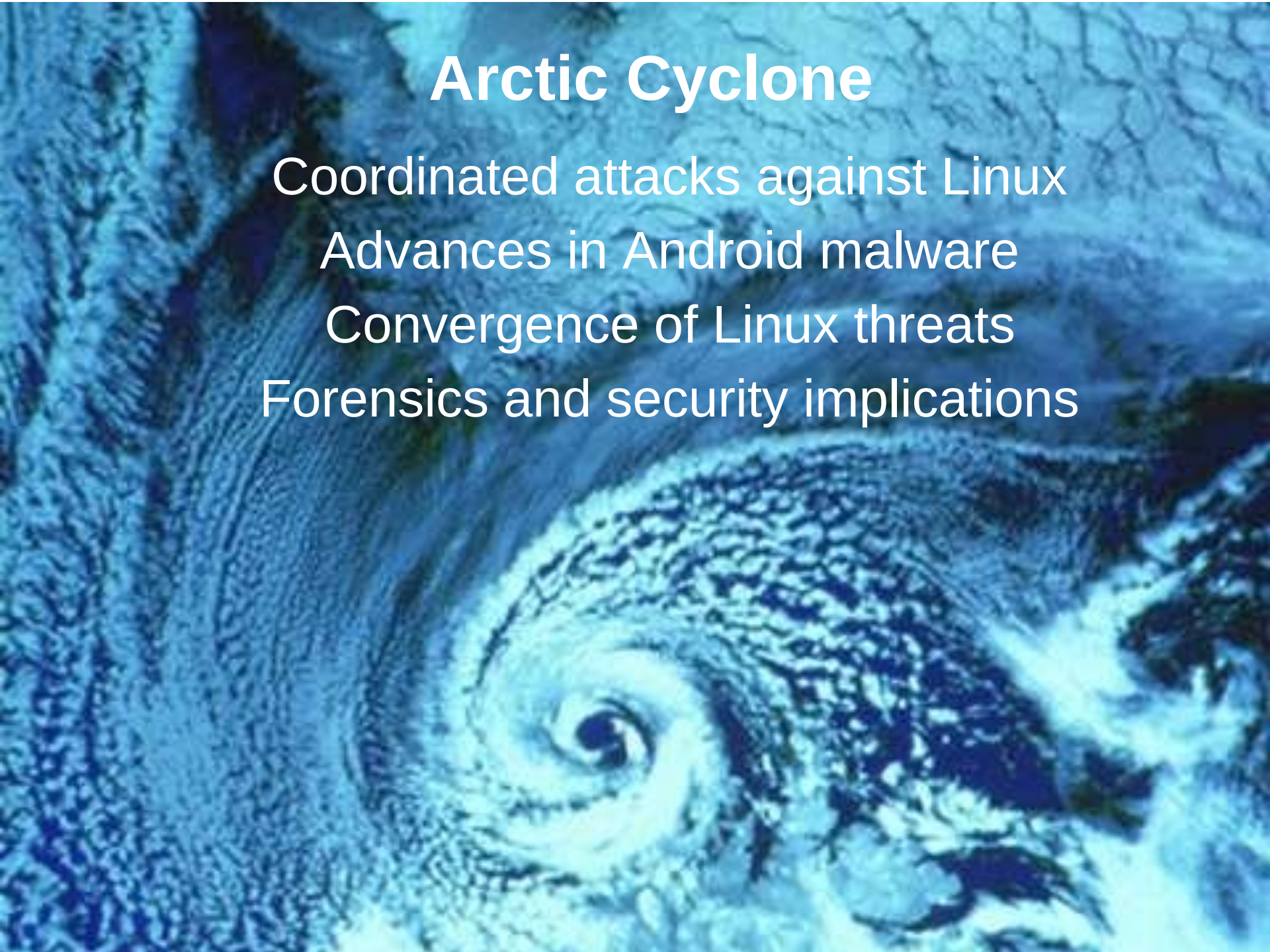
Challenges of Coordinated Linux & Android Intrusions

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®

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

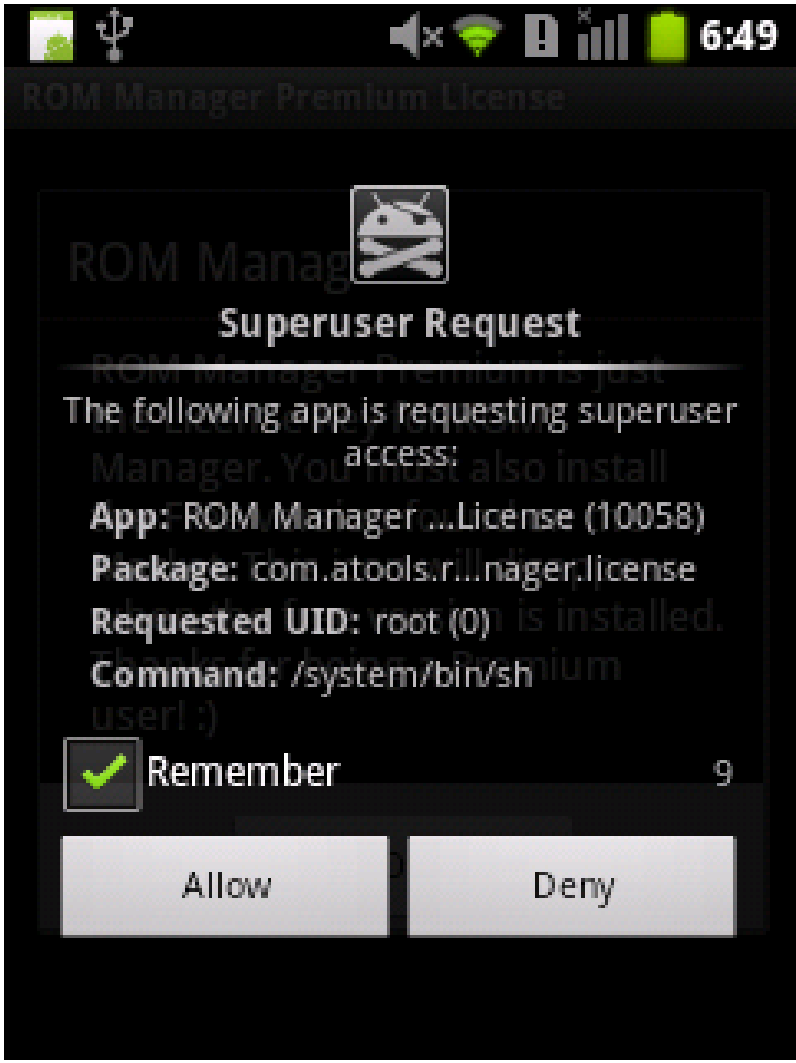
US-CERT is aware of active attacks against linux-based computing infrastructures using compromised SSH keys. The attack appears to initially use stolen SSH keys to gain access to a system, and then uses local kernel exploits to gain root access. Once root access has been obtained, a rootkit known as "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;
 , ƒ
 H „  "android/telephony/TelephonyManager  †
getDeviceId  ¶()Ljava/lang/String;
 ^ ‰
 ‡ Š  ◘ Œ  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

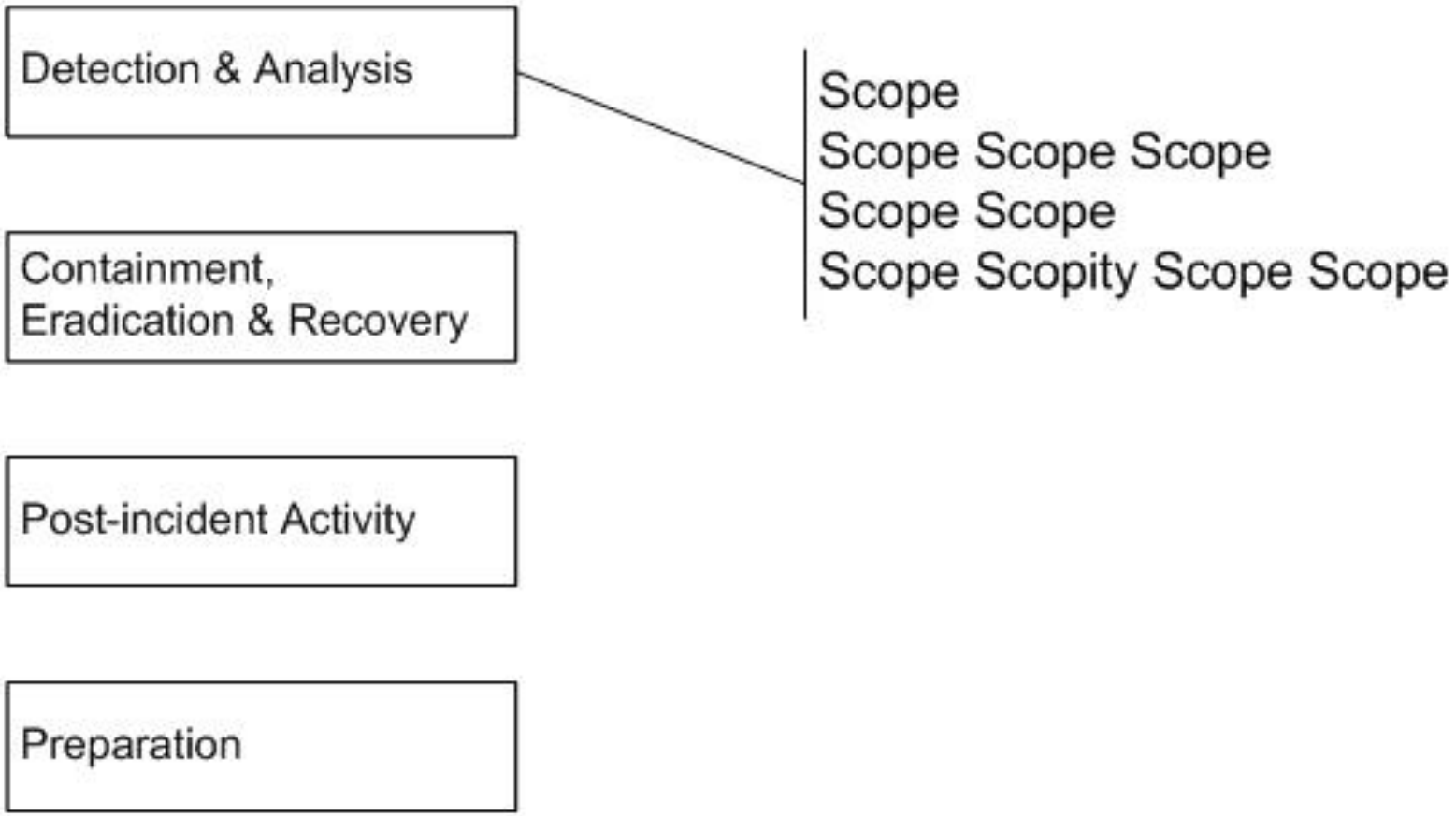
Quick Containment?

- Current recommendation:

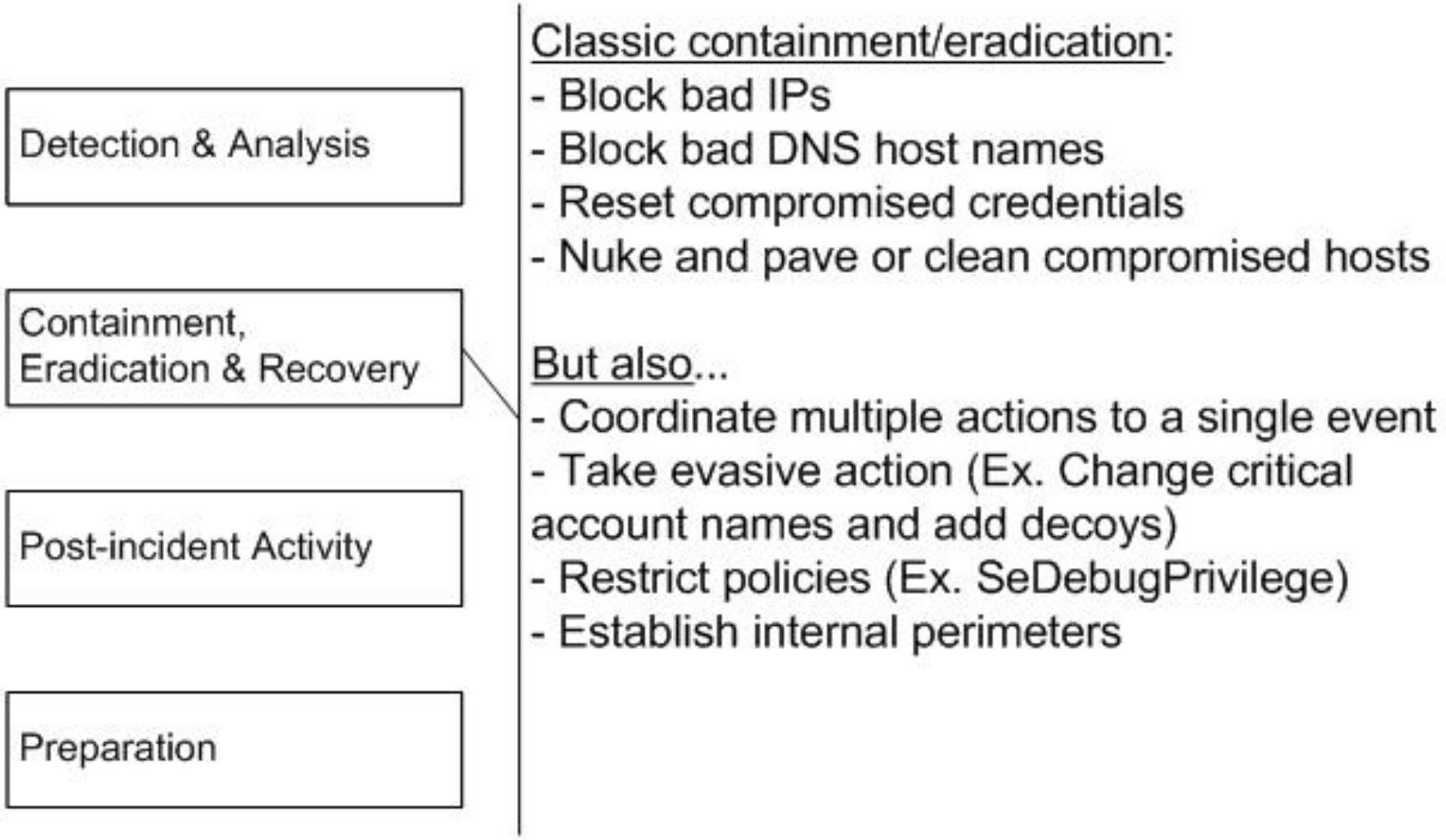
  When an incident has been detected and analyzed, it is important to contain it before the spread of the incident overwhelms resources or the damage increases. Most incidents require containment, so it is important to consider it early in the course of handling each incident.

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

Managing a data breach effectively



Effective Eradication of Intruders



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

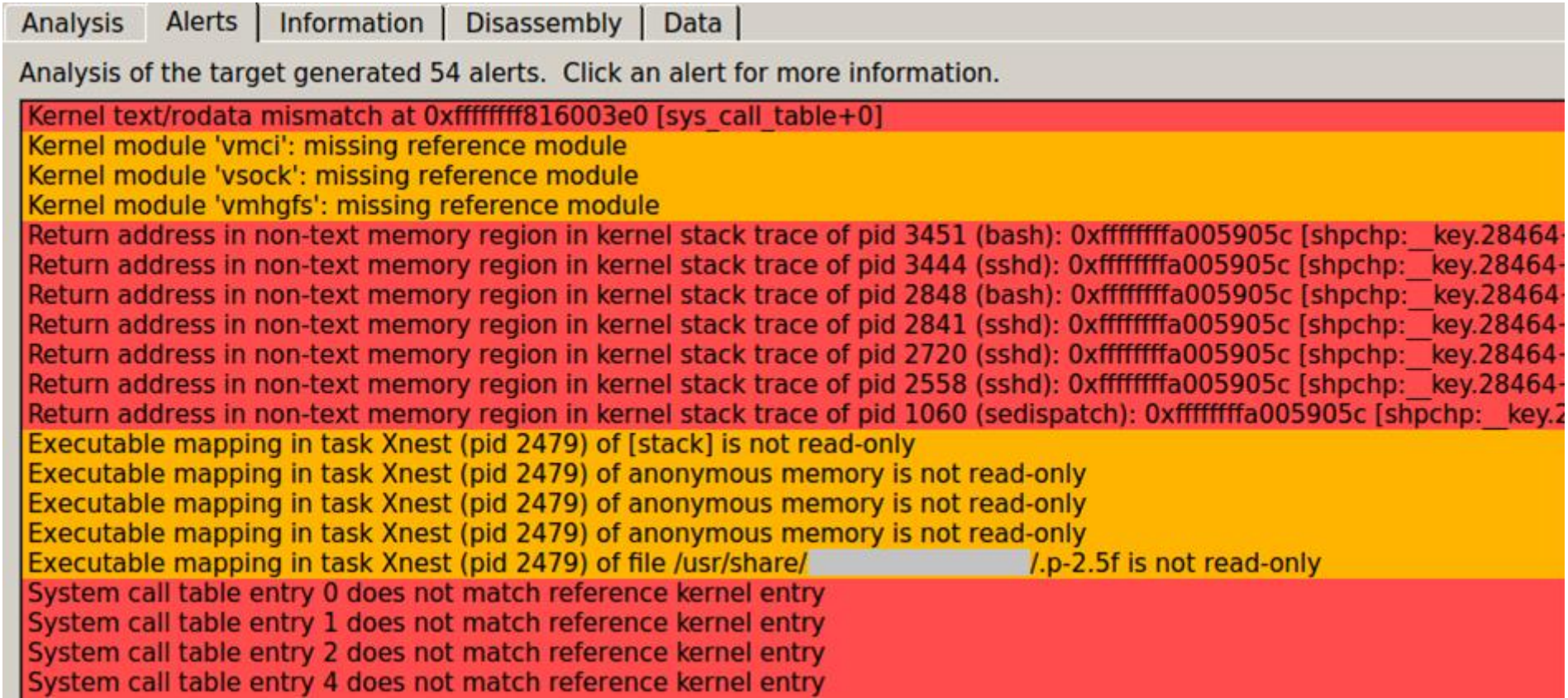
Linux Memory Forensics

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

```
% python vol.py -f Phlananx2 linux_check_syscall
Table Name                Index Address             Symbol
---------- ------------------ ------------------ ----------------------
64bit                       0x0 0xffffffffa0059000 HOOKED
64bit                       0x1 0xffffffffa0062000 HOOKED
64bit                       0x2 0xffffffffa0035000 HOOKED
64bit                       0x3 0xffffffff81115351 sys_close
64bit                       0x4 0xffffffffa00cb000 HOOKED
64bit                       0x5 0xffffffff8111aa73 sys_newfstat
64bit                       0x6 0xffffffffa00b5000 HOOKED
64bit                       0x7 0xffffffff81126170 sys_poll
<edited for length>
```

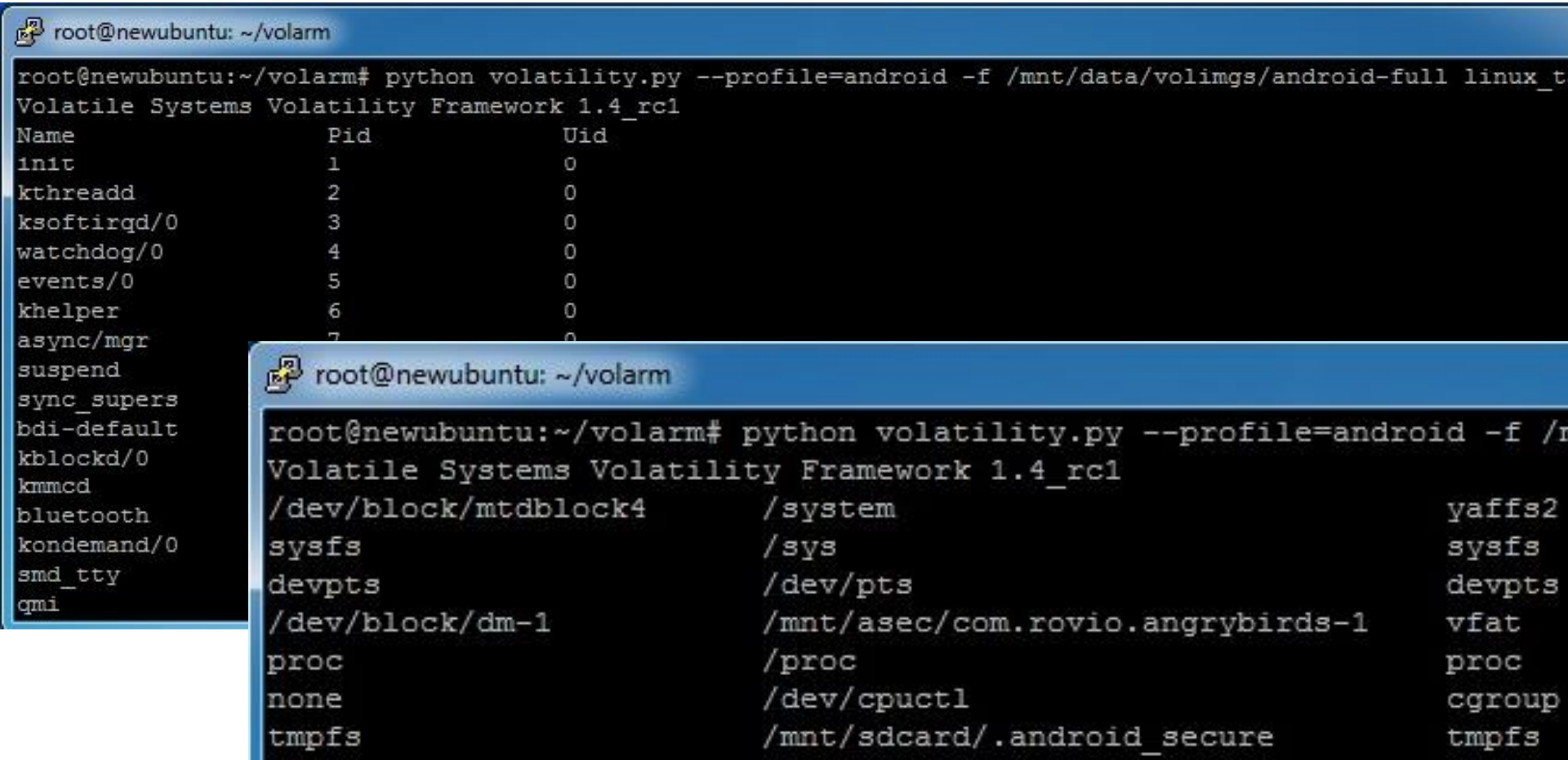
Linux Memory Forensics

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

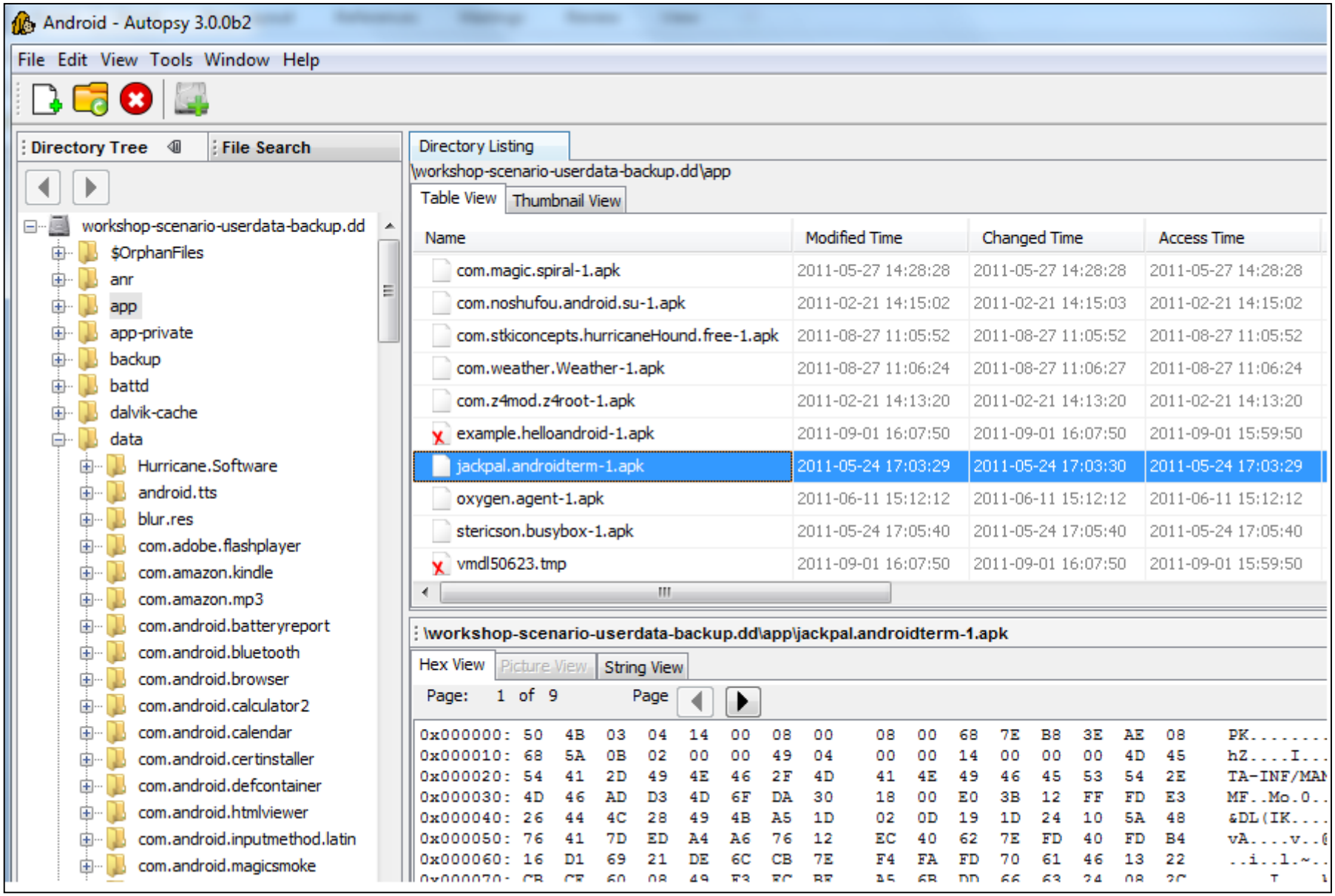


Android Memory Forensics

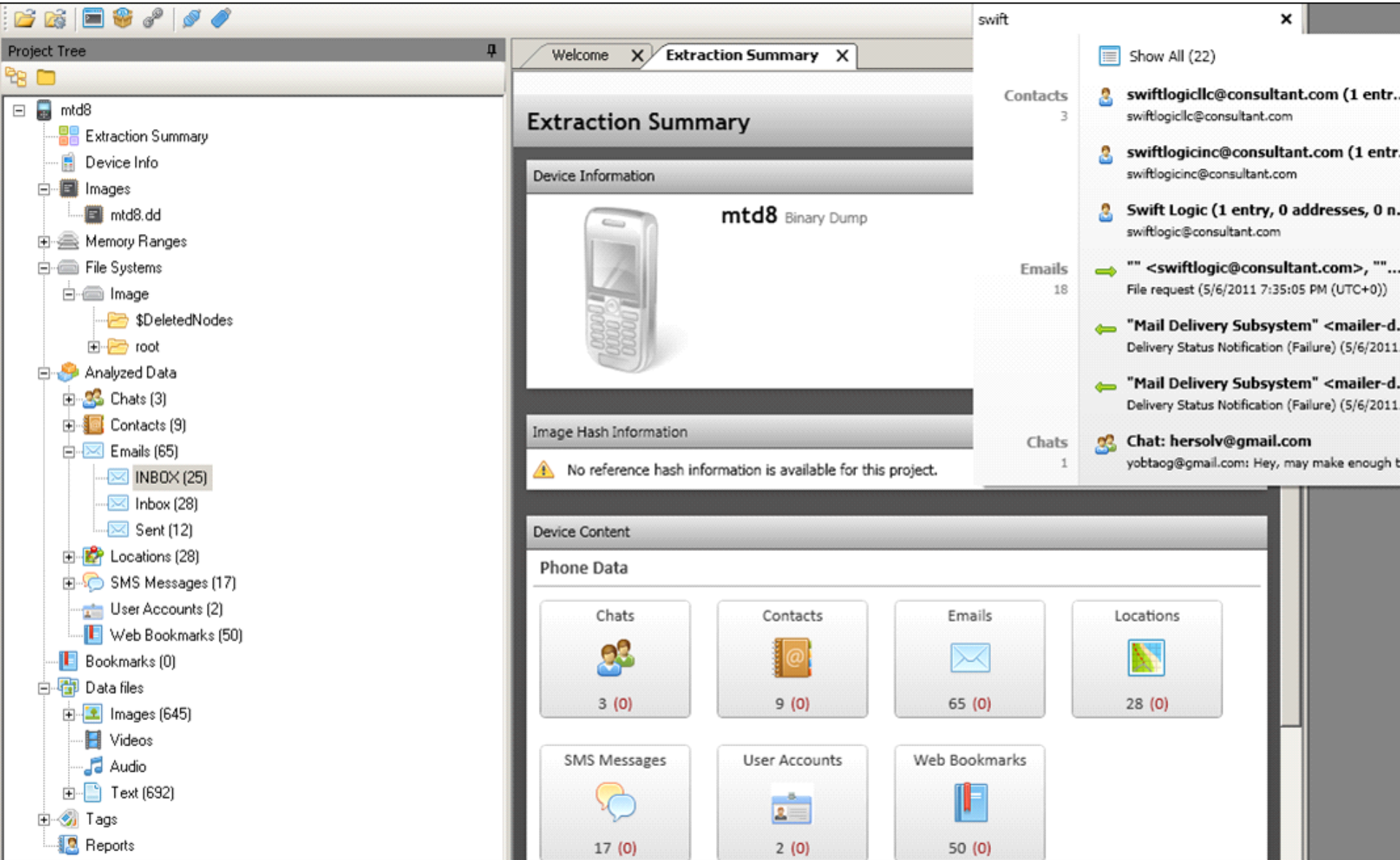
- Examination of Android physical memory
  - Volatility plugin for Android memory



Android File System Forensics



COTS Android File System Forensics



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
```

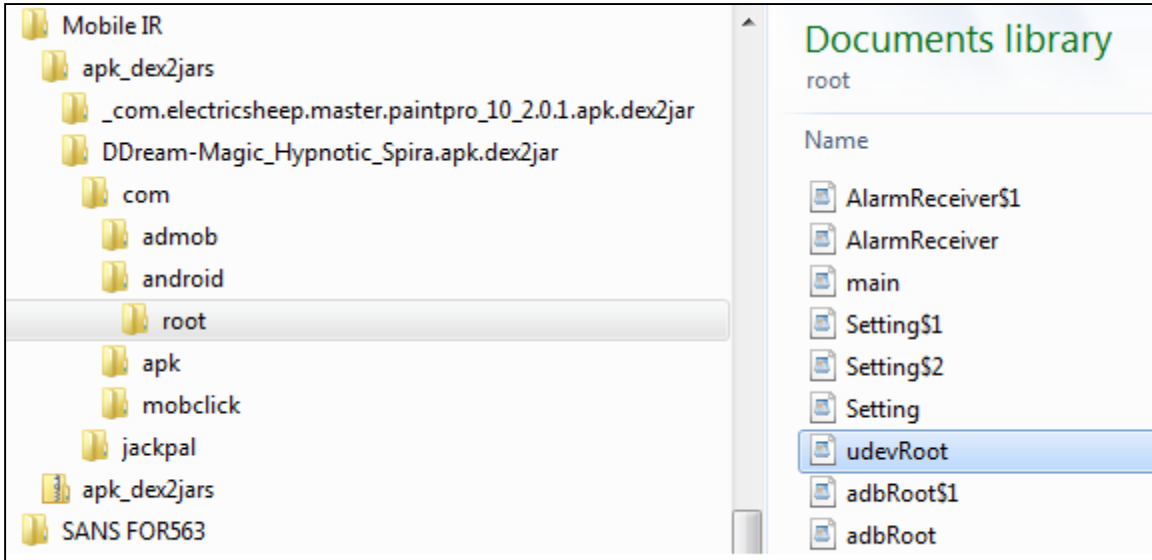
Remote Android Acquisition

- 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




```
Êþº¾   2 ┴  ├com/android/root/udevRoot     ┼java/lang/Object  └
udevRoot.java
BUFFER_SIZE   I└  ┘
FNAME_EXPLOIT  ↕Ljava/lang/String;   exploid◘
  ┴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
```

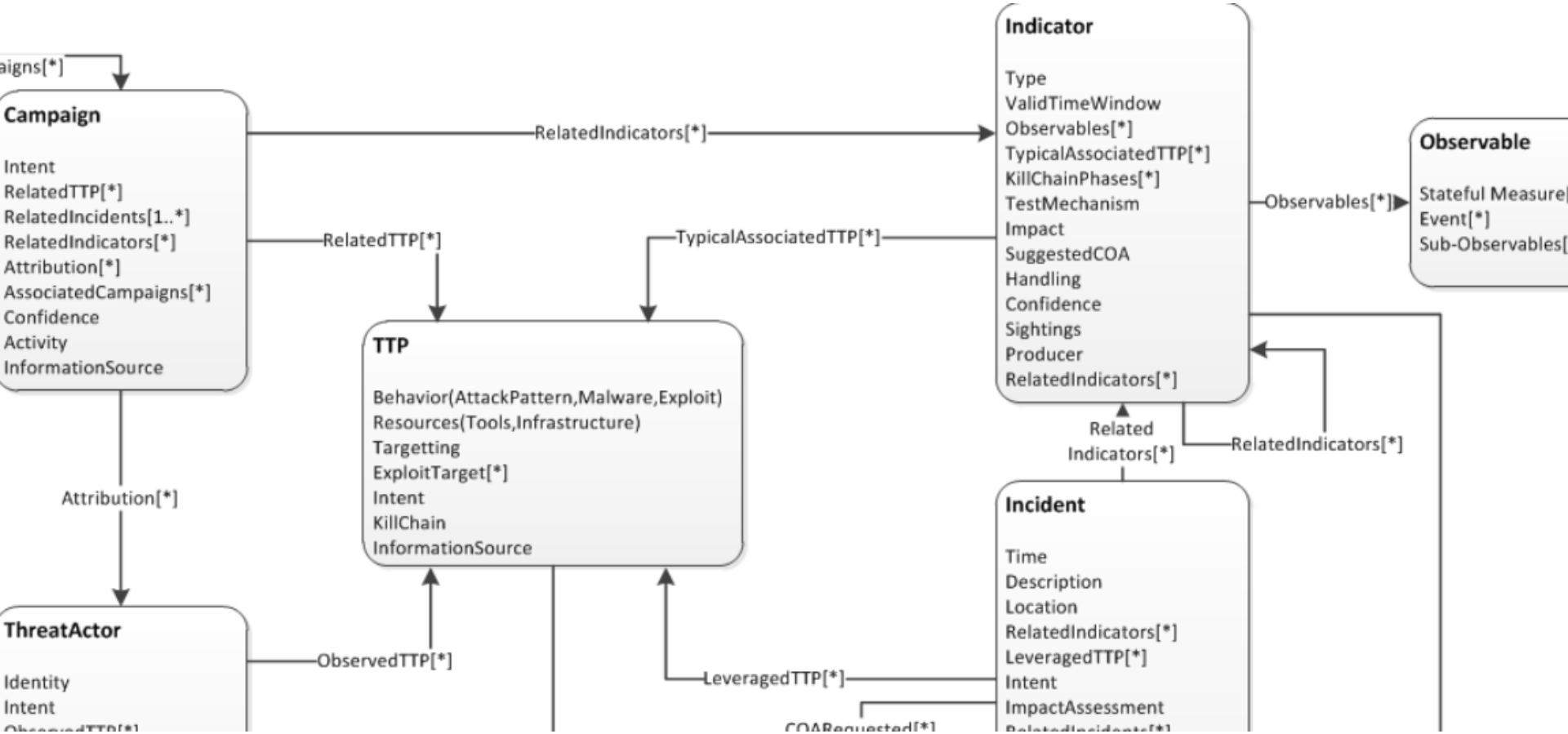
Cross Border Information Sharing

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



STIX Whitepaper - makingsecuritymeasurable.mitre.org/docs/STIX-Whitepaper.pdf

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