Science & Technology
Facilities Council

Security Incident Investigation

Mingchao Ma

STFC – RAL, UK

HEPSYSMAN Workshop

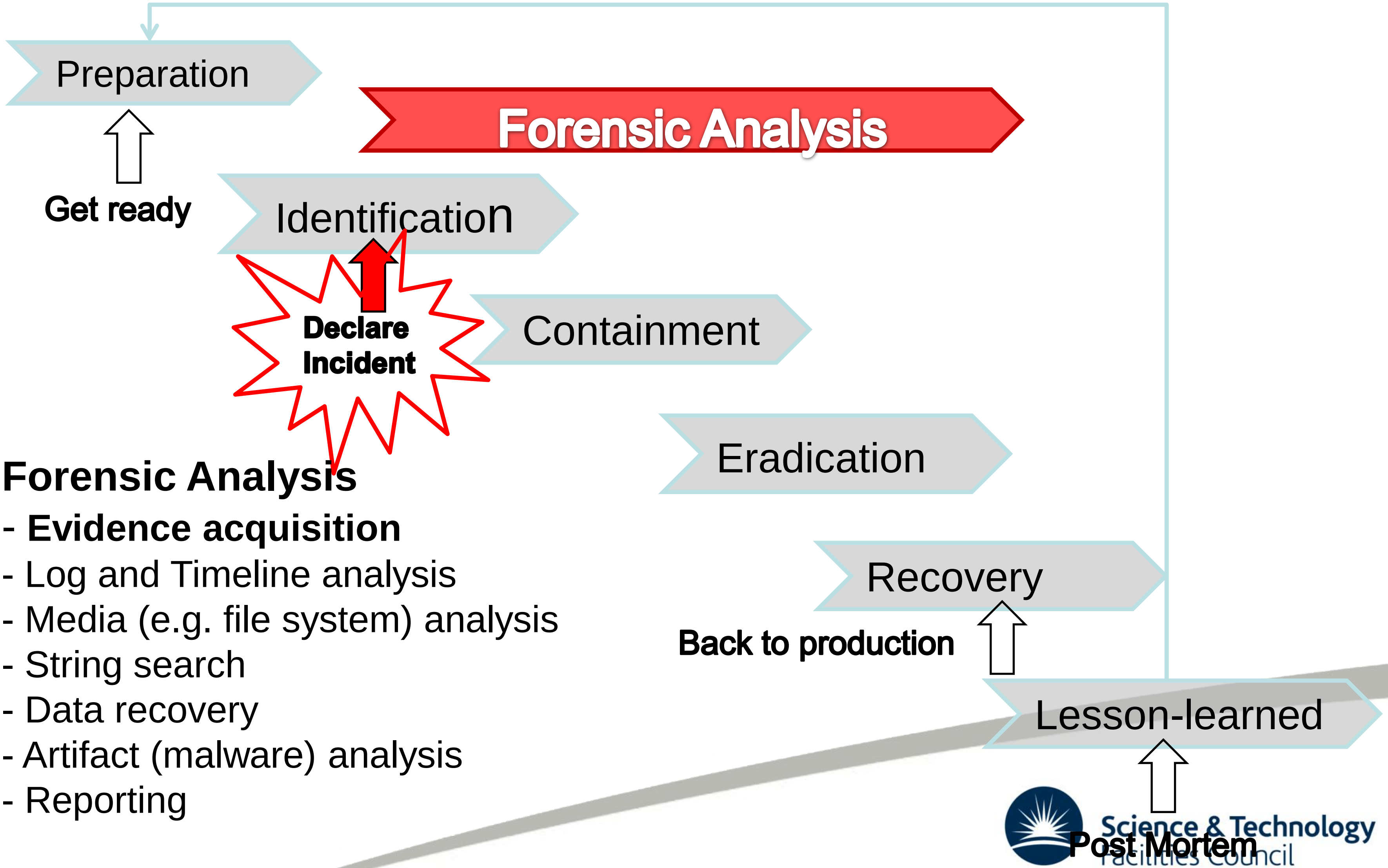
10th June 2010

Overview

- Security incident handling lifecycle
  - Based on **NIST SP800-61rev1** recommendation
  - http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf
- Aim at first responder
  - What and how to do?
- Tips and tricks on
  - **Evidence collection**
  - **Basic forensic**

It is a question of “when incident will happen”, not “if”

Incident Handling Lifecycle



Preparation

Get ready

Forensic Analysis

Identification

Declare Incident

Containment

Eradication

Recovery

Back to production

Lesson-learned

Post Mortem

Forensic Analysis

- **Evidence acquisition**
- Log and Timeline analysis
- Media (e.g. file system) analysis
- String search
- Data recovery
- Artifact (malware) analysis
- Reporting

Be warned!

- No two incidents are identical
- NO one-for-all solution, tailor it for your OWN need!
- Many types of incidents
  - DoS, Virus/Worm, Inappropriate usage, unauthorized access etc.
- Focus on “hacking scenario”
- But the principle remains the same!

Step 1 - Preparation

- Know existing policies, regulations and laws
  - **Authority** of investigation
    - Job description
    - Incident handling procedure
  - What information can be collected?
  - Privacy and wiretapping issue
- Do not violate any existing security policies
- And do not break laws!

Preparation

- Security policy and incident handling procedure
  - Policies & procedures, write them down on PAPER
  - A simple and easy-to-follow procedure is very helpful
- Building a team
  - Information about the team - "Organizational Models for Computer Security Incident Response Teams (CSIRTs) (http://www.cert.org/archive/pdf/03hb001.pdf)
- Contacts information and communication channels
  - Name, telephone, email, PGP keys etc.
- Incidents Prevention
  - Risk assessment
  - Patching, hardening, best practice, education etc.
  - Be aware of your organization's security policy
- **Known your systems before an incident**
  - Profile systems and network
  - Know normal behaviours

Toolkit – Live CDs

- Incident response toolkit
  - Linux forensic live CDs
    - Helix (no longer free ☹) - http://e-fense.com/
      - Live response, live/dead acquisition and analysis
    - FCCU GNU/Linux Forensic Boot CD
      - Belgian Federal Computer Crime Unit
      - http://www.lnx4n6.be/
    - BackTrack 4 has an option to boot into forensic mode
      - http://remote-exploit.org/backtrack.html
    - Many others
  - Will not modify the target system harddisk
    - Will not auto-mount devices on target system
    - Will not use target system swap partition
    - Build-in some well-known open source forensic tools

Toolkit - Forensic

- Any Linux system plus proper open source forensic tools
- US CERT forensic appliance (fedora)
  - A fully functional Linux VM forensics appliance
  - Linux Forensics Tools Repository (RPMs for fedora)
  - http://www.cert.org/forensics/tools/
- SANS SIFT workstation (Ubuntu)
  - VM forensic appliance
  - https://computer-forensics2.sans.org/community/siftkit/
  - Free, but registered first
- BackTrack
- Load of tools readily available

Toolkit - Forensic

- TSK + Autopsy (GUI-frontend)
  - The Sleuth Kit and Autopsy browser
  - http://www.sleuthkit.org/
  - Alternative – PSK (GUI-frontend)
    - http://ptk.dflabs.com/
- The Coroner's Toolkit (TCT)
  - http://www.porcupine.org/forensics/tct.html

Science & Technology
Facilities Council

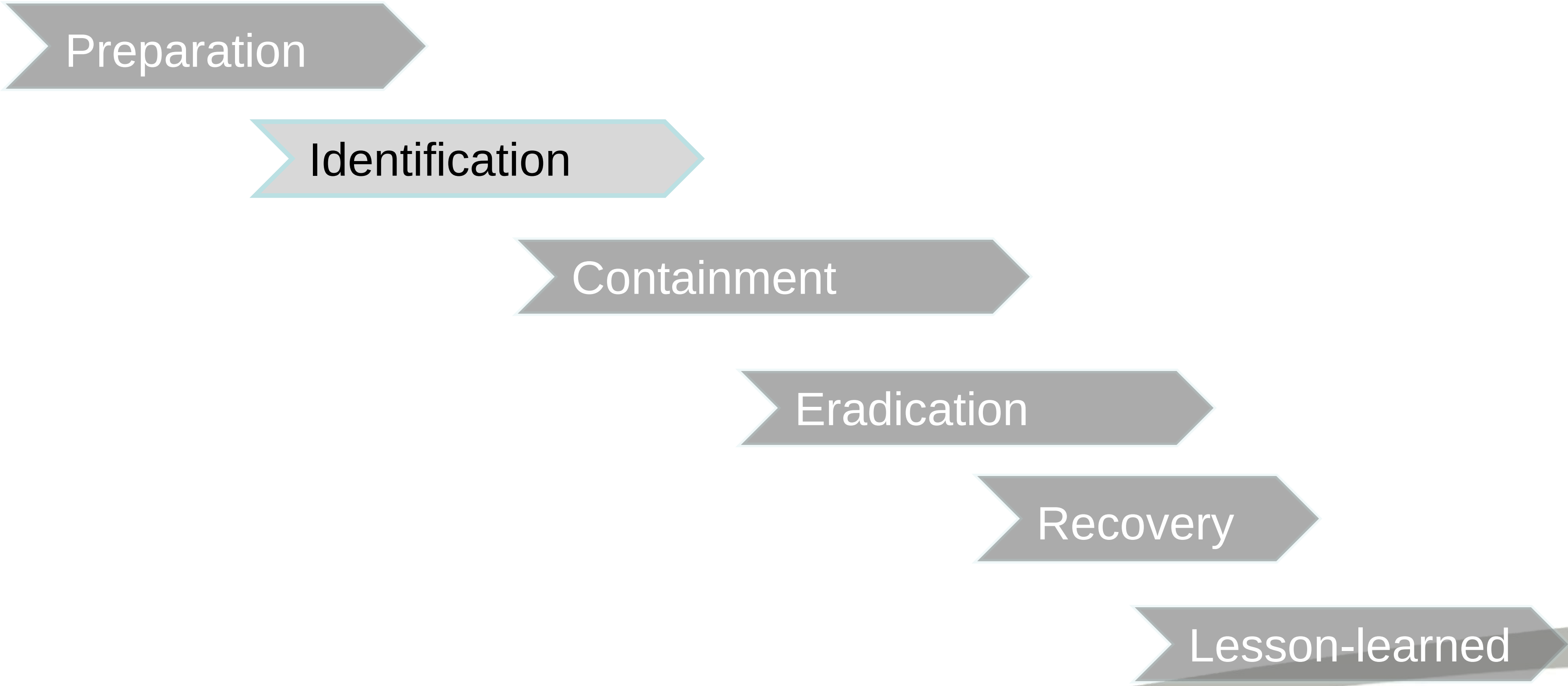
Toolkit – Network forensic

- Wireshark/tshark
- Tcpdump
- Nmap
- Snort
- P0f (OS passive fingerprinting)
- Antivirus software
  - http://www.clamav.net/
  - AVG and avast! for Linux, free!

Toolkit – Build in

- Trusted binaries - **statically compiled** binaries run from CD or USB
  - ls, lsof, ps, netstat, w, grep, uname, date, find, file, ifconfig, arp … …
- Test before use
  - different Linux distributions and kernels
  - both 32 bit and 64 bit platform
- Will not modify A-time of system binaries;
- Be aware of limitation – can be cheated as well
  - Kernel mode rootkit

Incident Handling Lifecycle



Preparation

Identification

Containment

Eradication

Recovery

Lesson-learned

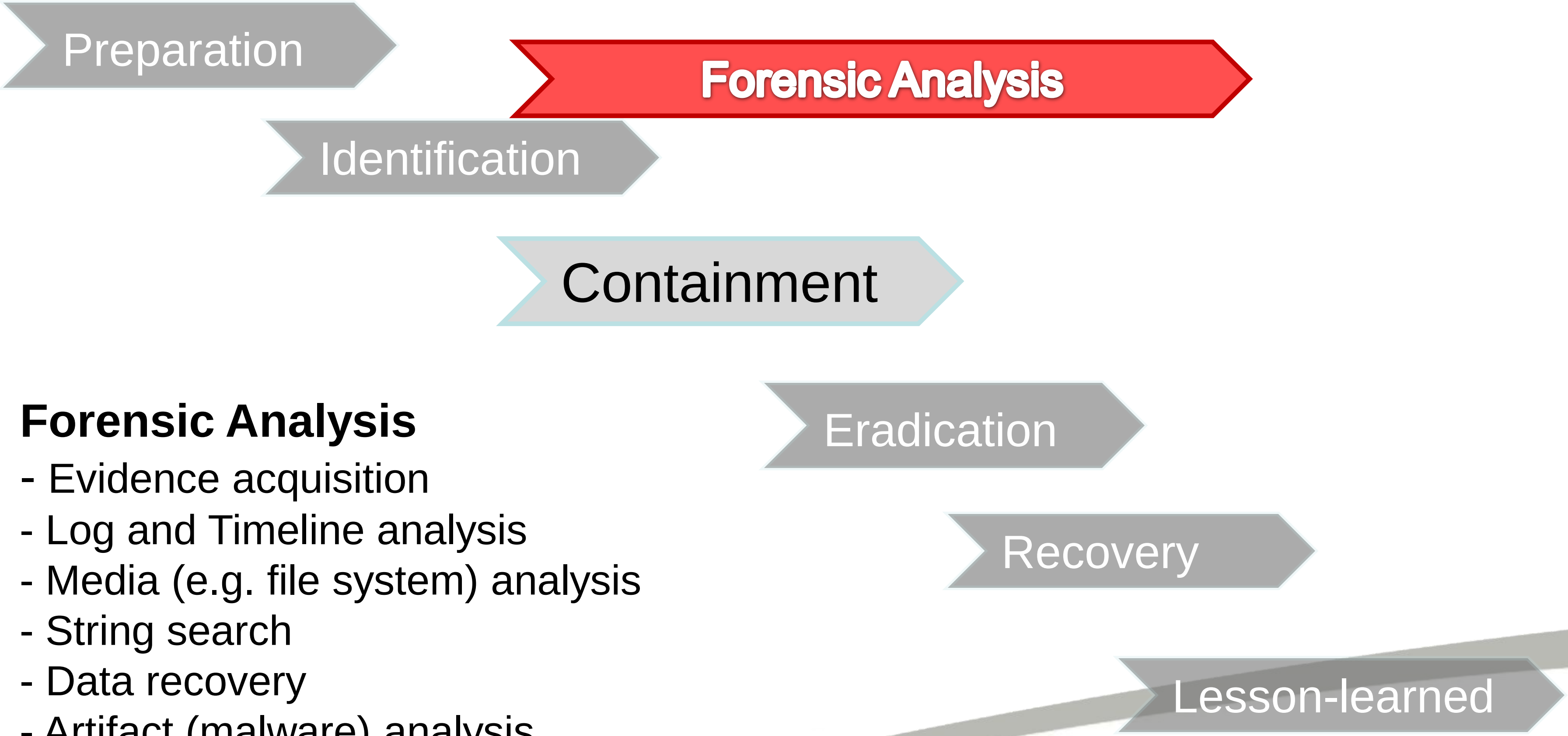
Step 2 - Identification

- Detect deviation from normal status
  - **Alerted by someone else**;
  - Host & network IDS alerts;
  - antivirus/antispyware alerts;
  - Rootkit detection tools;
  - file integrity check;
  - System logs;
  - firewall logs;
  - A trusted central logging facility is essential;
  - Correlate all information available to minimise **false alarm**

Identification

- Declare an incident once confirmed
  - Make sure that senior management is informed
  - Notification – who should be notified?
  - EGEE CSIRTs: PROJECT-EGEE-SECURITY-CSIRTS@in2p3.fr
- Following incident handling **procedures**
  - EGEE incident response procedure
  - https://edms.cern.ch/document/867454

Incident Handling Lifecycle



Preparation

Forensic Analysis

Identification

Containment

Eradication

Recovery

Lesson-learned

Forensic Analysis

- Evidence acquisition
- Log and Timeline analysis
- Media (e.g. file system) analysis
- String search
- Data recovery
- Artifact (malware) analysis
- Reporting

Step 3 – Containment & Forensic Analysis

- Prevent attackers from further damaging systems
- Questions to be answered!
  - Online or Offline?
    - Pull the network cable?
  - Live or Dead system?
    - Pull the plug?

Forensic Analysis

- Start up forensic analysis process once incident has been identified
  - Aim to obtain forensic sound evidences
  - Live system information
    - Will lose once powered off
  - Bit by bit disk image
  - Logs analysis
  - Timeline analysis
  - Data/file recovery
- Collect volatile data FIRST, if possible!

How to collect evidences

- Volatile data collection
- Hard disk image
- Where to store evidences?
  - Attach a USB device
  - Transfer data over network with *netcat*

Evidence workstation (192.168.0.100):

```
# ./nc –l –p 2222 > evidence.txt
```

Compromised host:

```
#./ lsof–n |nc 192.168.0.100 2222
```

Volatile Data Collection

- Aim:
  - Collect as much volatile data as possible
  - But **minimise** footprint on the target system
- In the order of most volatile to least
  - Memory
  - Network status and connections
  - Running processes
  - Other system information
- Be warned: system status will be **modified**
- Document everything you have done
- Be aware of the concept of “chain of custody”
  - Maintain a good record (a paper trail) of what you have done with evidence

Volatile Data Collection?

- System RAM
  - Raw memory image with *memdump*
    Available at http://www.porcupine.org/forensics/tct.html
  - Hardware-based memory acquisition?
  - Virtual Machine
    - Take a snapshot
- Network Information
  - open ports and connections
    - *lsof* and *netstats*
    - *Nmap*
- Process information
  - Running processes with *ps*
  - Process dumping with *pcat*
    - Available at http://www.porcupine.org/forensics/tct.html

Science & Technology
Facilities Council

Other volatile data

- System Information
  - – System uptime: *uptime*
  - – OS type and build: *uname –a*
  - – Current date/time: *date*
  - – Partition map: *fdisk -l*
  - – Mount points: *mount*
  - – … …?

What to do with memory image?

- Linux memory dump
  - Very limited option (at least with open source tools)
  - Strings search for IP, email or strange strings etc
  - Can be used to cross check with evidence found in file system/logs
  - Some ongoing researches in open source community

Collect Evidence – Disk Image

- Bit by bit disk image
  - Capture both allocated and unallocated space
- Do not use gzip/tar or normal backup tools
  - Lose unallocated space
  - Can’t recover deleted files
- How to do it?
  - Live system vs dead system image?
  - Full disk vs Partition?

Disk Image

- Live system image vs Dead system image?
  - – Helix Live CD or FCCU Live CD
  - – Or USB
  - – Writeblocker?

Disk Image

- Full disk vs. Partition?
- Full disk if possible
  - Get everything in one go
  - Can copy host protection area - HPA (after reset)
  - Might not be feasible
    - RAID system: too big, RAID reconstruction?
- Image only partition
  - OS partitions

Disk image

- Linux *dd* command
  - Full disk
    - *dd if=/dev/sda of=/mnt/usb/sda.img bs=512*
  - Partition
    - *dd if=/dev/sda1 of=/mnt/usb/sda1.img bs=512*
- Enhanced *dd* – e.g. *dc3dd or dcfldd*
  - *http://dc3dd.sourceforge.net/*
  - *http://dcfldd.sourceforge.net/*
  - *dcfldd if=/dev/sourcedirve hash=md5 hashwindow=10M md5log=md5.txt bs=512 of=driveimage.dd*
- *dd_rescue*
  - *http://www.gnu.org/software/ddrescue/ddrescue.html*

Science & Technology
Facilities Council

What to do with disk images?

- Mount disk image/partition to the loop device on a forensic workstation in READ ONLY mode
  - *mount -o loop, ro, offset=XXXX disk_image.dd /mnt/mount_point*
- Partition information can be obtained
  - *sfdisk –l disk_image.dd*
  - *fdisk –lu disk_image.dd*
  - *mmls –t type disk_image.dd*
    - In the TSK toolset
- Either work on the whole image
  - Use the “offset”parameter
- Or, split the image to individual partitions and then mount them separately
  - *dd if=disk_image.dd bs= 512 skip=xxx count=xxx of=partition.dd*

Science & Technology
Facilities Council

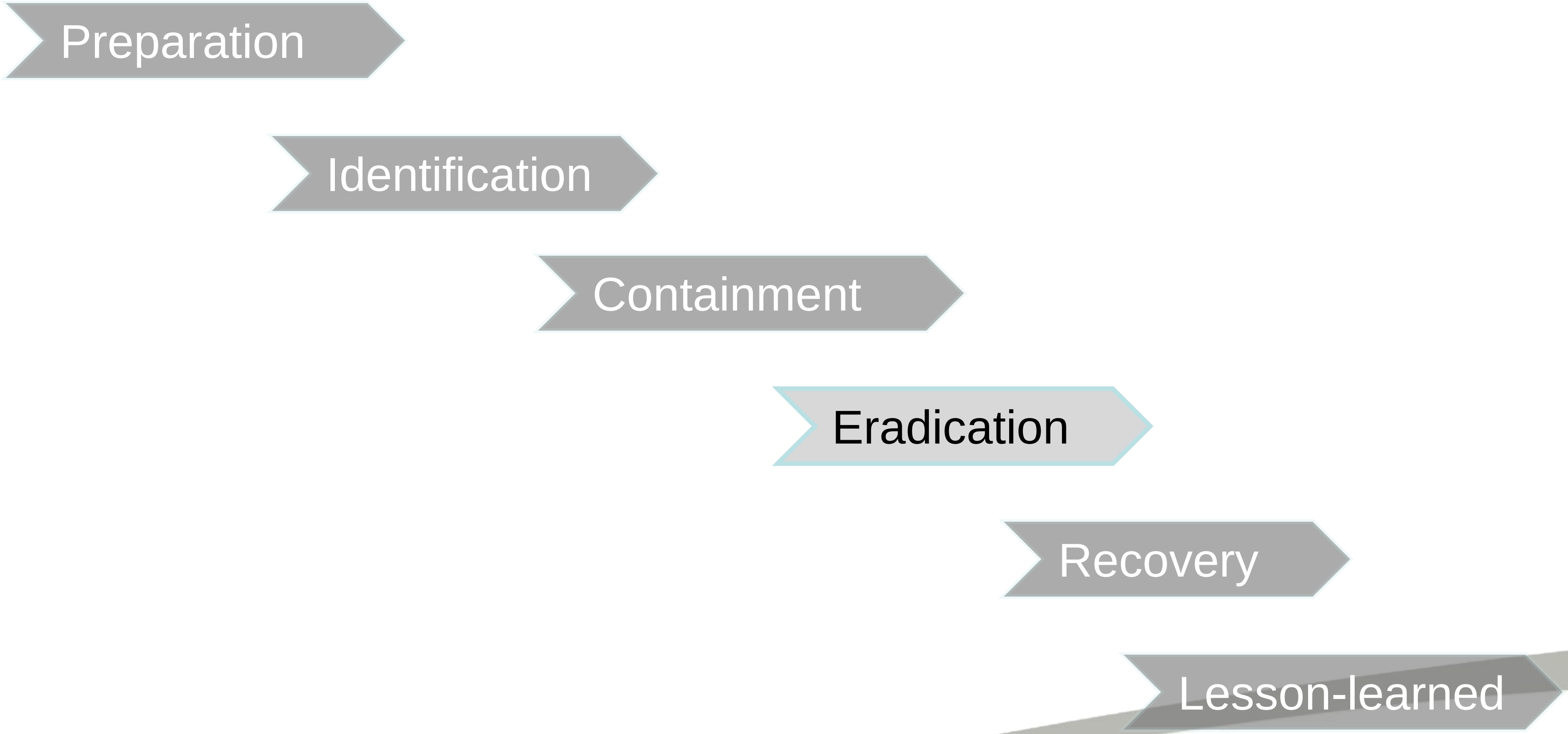
Evidence Collection

- Memory dump;
- Network status;
- Process dump;
- Other system information;
- Disk images;
- Forensic analysis done on the images NOT on the original disk;

After Evidence Collection

- Mount disk/partition images on a trusted system
- Timeline analysis with *TSK*
  - What had happened?
- Media (e.g. file system) analysis with TSK
  - What was modified/changed and or left?
- String search on both allocated and unallocated areas with *strings*
- Data recovery with *TSK*
  - What was deleted?
- Artifact (malware) analysis
  - To understand the function of the malware
- Sharing findings with relevant parties

Incident Handling Lifecycle



Preparation

Identification

Containment

Eradication

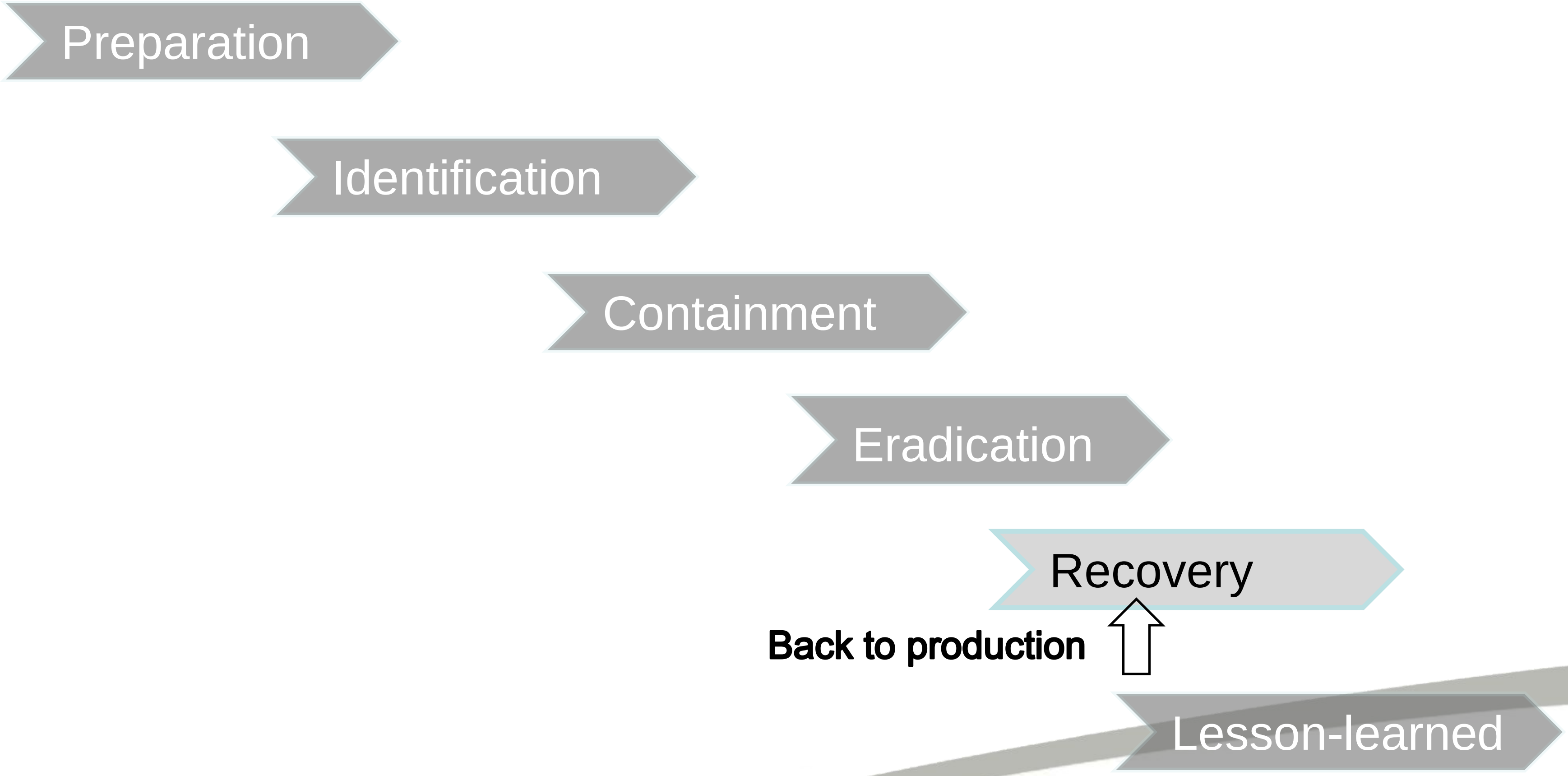
Recovery

Lesson-learned

Step 4 Eradiation

- Remove compromised accounts
- Revoke compromised credentials
- Remove malware/ artifact left over by the attackers
- Restore from most recent clean backup
- If root-compromised, **rebuild** system from scratch
- Harden, **patch** system to prevent it from re-occurrence

Incident Handling Lifecycle



Preparation

Identification

Containment

Eradication

Recovery

Back to production

Lesson-learned

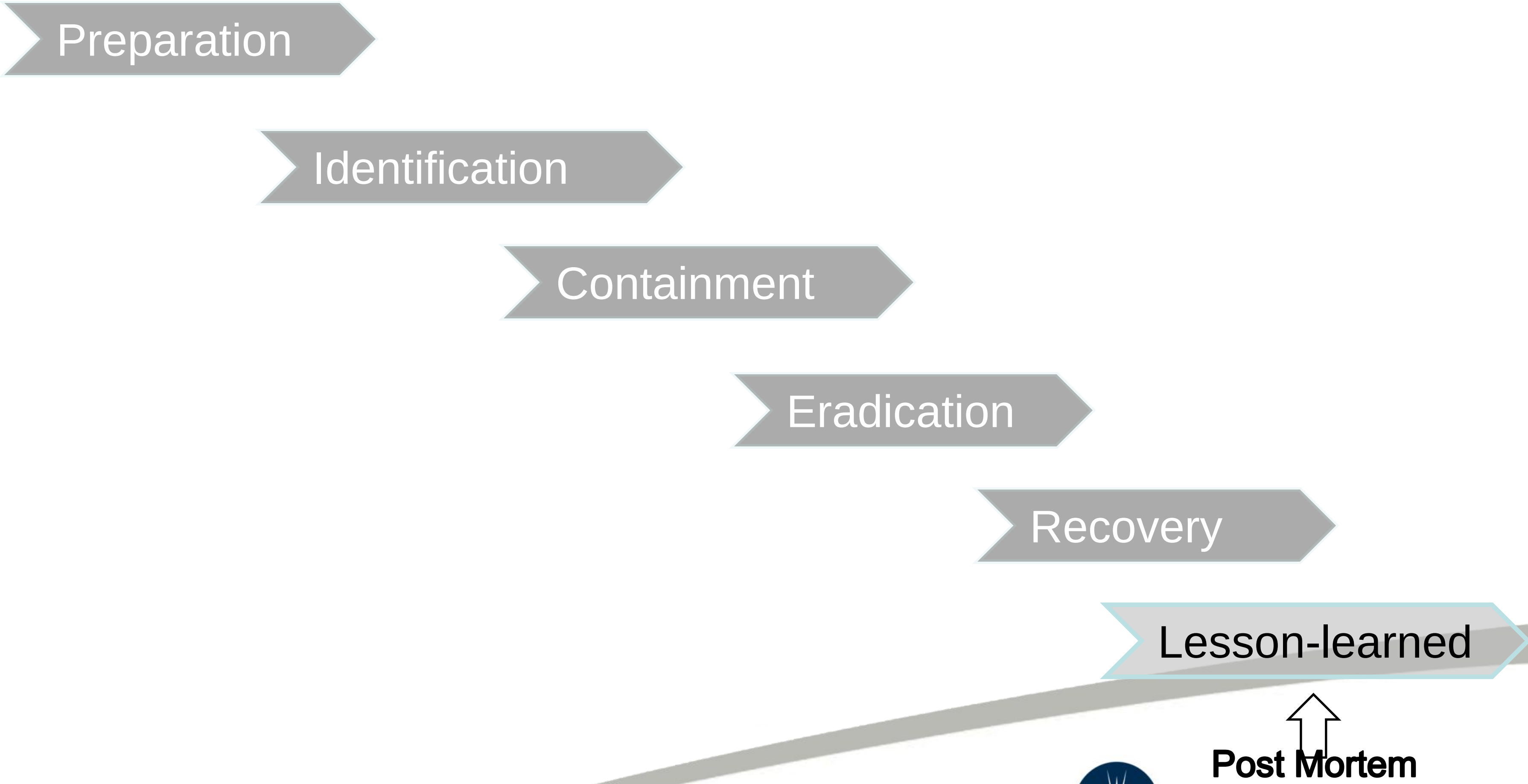
Science & Technology
Facilities Council

Recovery

- Put system back to production in a control manner
- Decision should be made by management
- Closely monitoring the system

Science & Technology
Facilities Council

Incident Handling Lifecycle



Science & Technology Facilities Council

Step 6 – Lesson learned

- Know what went right and what went wrong
  - Learning & improving
  - A post-mortem meeting/discussion

Science & Technology
Facilities Council

Thanks

DEMO

How to detect rootkit in a live Linux system?

The rootkit

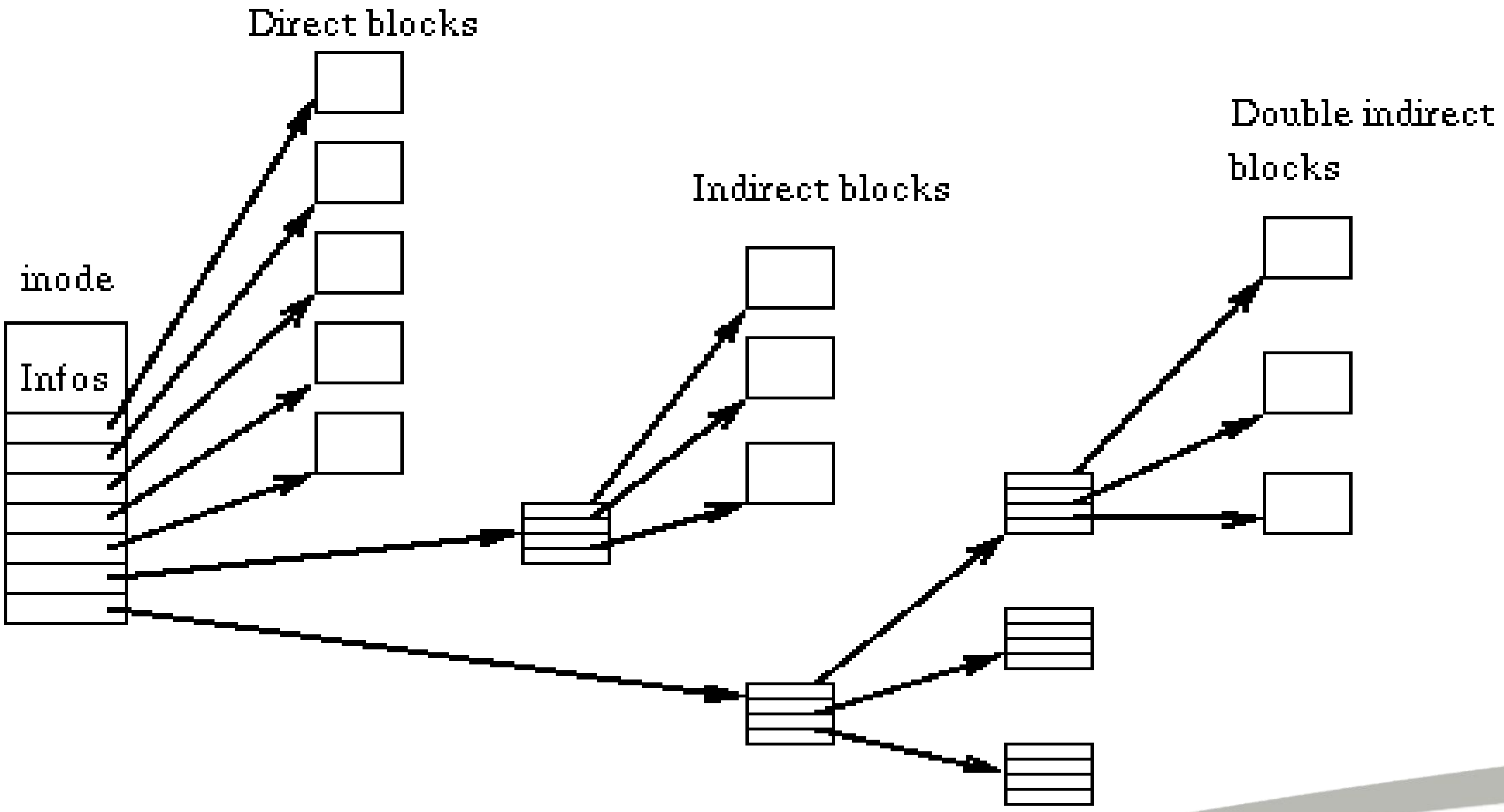
- Captured in last year incident
- Kernel mode rootkit with sniffing backdoor
- Hide itself and relevant files from normal detection
- Can survive from system reboot
- Protected with password

DEMO

EX2/EX3 file system premier

- Superblock
  - – Block size, number of blocks, number of Inodes, number of reserved blocks, number of blocks per group, number of Inodes per Group
- Block Groups
  - – All blocks belong to a Block Group
  - – Begins from block 0, after reserved blocks
  - – Each Block Group
    - Superblock backup
    - Group Descriptor Table
    - Block Bitmap, Inode Bitmap
    - Inode Table, Data Blocks

EX2/3 Meta Data structure



Directories

- Directory itself is a file
- A sequence of entries
  - – Inode number
  - – File name
  - – Size of file name

Byte Offset	Inode Number	File Names
0	80	.
16	8	..
32	1674	init
48	69	fstab
64	1978	passwd
80	115	group
…	…	…