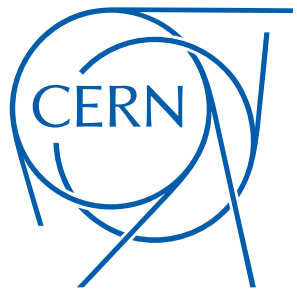
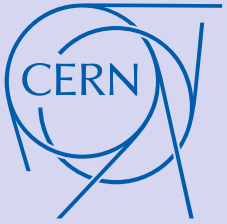
# Intrusion Detection

Vincent Brillault

CERN (European Organization for Nuclear Research)

October 2013

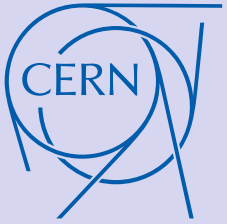






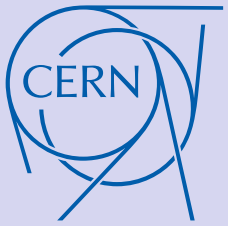



# Content





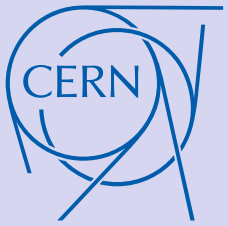

# Contents






# Network-based IDS

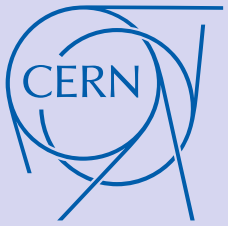
- Existing solutions: Bro, snort...
- Need correct configuration
- Miss a lot of details:
  - encrypted traffic
  - user on shared host
- Undetectable
- Need computing power depending on traffic






# Host-base IDS

- Existing solutions: OSSEC, Samhain, Tripwire
- Much more details
- Burden on every production host
- Potentially detectable





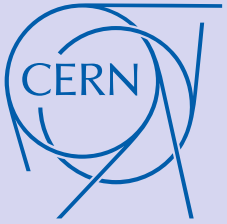

# Arm-race

## Stay hidden, detect the other

- Attacker will try to detect IDS:
  - Look for standard IDS traces
  - Look for half-hidden kernel modules
  - Look for remote logging
- *Protected* hosts could be avoided

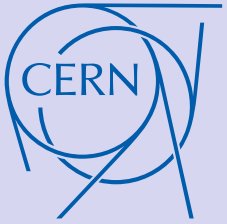
  $\rightarrow$Difficult to detect compromise

  $\Rightarrow$Undetected user credential compromise
- False feeling of security?






# Contents



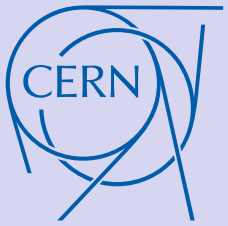



# Remote logging

## Central logging servers

Send logs to protected cluster of servers

- Allow on the fly offline analysis of logs (and alerts)
- Ciphered transport available (expensive, dangerous?)
- Integrity (post-detection analysis):
  - Local logs can be altered
  - Dedicated central server can be secured

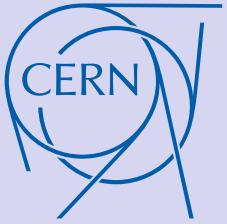





# Remote logging

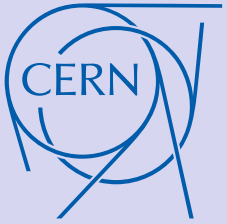
## Storage

- Disk space is cheap, compromise consequences are not
  $\rightarrow$ Store everything
- Standard big data issue:
  - Store interesting processed (login) data separately
  - Use scalable storage and processing, e.g. Hadoop
- Confidentiality issues: policies and life-cycles needed






# Contents



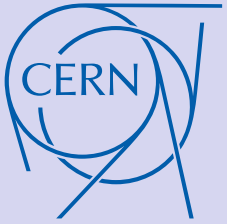



# Sources

## Base

All basic unix logs:

- system logs (syslog), including:
  - authentication logs (includes SSH and failures)
  - daemon logs
  - cronjob logs
  - mails send/received
- kernel logs
  - iptables logs
  - SELinux (if no auditd): 'denied' actions



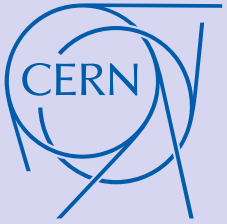



# Sources

## Network logs - network based

Traffic headers (e.g. NetFlow) from routers/firewalls

- Undetectable
- Backup for network-based IDS
- Post incident analysis
- Dedicated hosts: safe after production host compromise
- Require router support

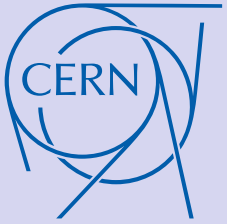





# Sources

## Network logs - network based

```
2013/07/01-09:35:36 tcp 213.251.186.118:22 -> 137.138.29.205:38109 1919 secs 1 flows 3436 packets 1572864 bytes
2013/07/01-09:35:36 tcp 137.138.29.205:38109 -> 213.251.186.118:22 1919 secs 1 flows 3312 packets 179040 bytes
2013/07/01-10:07:36 tcp 213.251.186.118:22 -> 137.138.29.205:38109 1920 secs 1 flows 3827 packets 1363149 bytes
2013/07/01-10:07:36 tcp 137.138.29.205:38109 -> 213.251.186.118:22 1920 secs 1 flows 3785 packets 201604 bytes
2013/07/01-10:39:38 tcp 213.251.186.118:22 -> 137.138.29.205:38109 1920 secs 1 flows 2677 packets 1468006 bytes
2013/07/01-10:39:38 tcp 137.138.29.205:38109 -> 213.251.186.118:22 1920 secs 1 flows 2532 packets 136560 bytes
2013/07/01-11:11:39 tcp 137.138.29.205:38109 -> 213.251.186.118:22 1919 secs 1 flows 4603 packets 283756 bytes
2013/07/01-11:11:39 tcp 213.251.186.118:22 -> 137.138.29.205:38109 1919 secs 1 flows 4701 packets 1572864 bytes
2013/07/01-11:43:40 tcp 137.138.29.205:38109 -> 213.251.186.118:22 1920 secs 1 flows 3684 packets 195552 bytes
2013/07/01-11:43:40 tcp 213.251.186.118:22 -> 137.138.29.205:38109 1920 secs 1 flows 3712 packets 1363149 bytes
2013/07/01-12:15:41 tcp 213.251.186.118:22 -> 137.138.29.205:38109 1919 secs 1 flows 3947 packets 1363149 bytes
2013/07/01-12:15:41 tcp 137.138.29.205:38109 -> 213.251.186.118:22 1919 secs 1 flows 3926 packets 207704 bytes
2013/07/01-13:16:11 tcp 137.138.144.183:25 -> 213.251.186.118:56112 0 secs 1 flows 13 packets 2641 bytes
2013/07/01-13:16:12 tcp 213.251.186.118:56112 -> 137.138.144.183:25 0 secs 1 flows 14 packets 2802 bytes
2013/07/01-12:47:42 tcp 137.138.29.205:38109 -> 213.251.186.118:22 1920 secs 1 flows 5619 packets 345676 bytes
2013/07/01-12:47:42 tcp 213.251.186.118:22 -> 137.138.29.205:38109 1920 secs 1 flows 5856 packets 2411725 bytes
2013/07/01-13:19:43 tcp 213.251.186.118:22 -> 137.138.29.205:38109 1919 secs 1 flows 6046 packets 1992294 bytes
```



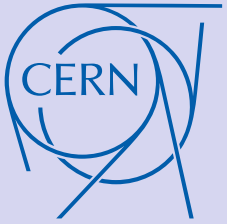



# Sources

## Network logs - host based

Kernel module 'netlog' (@CERN)

- Uses system-tap back-end: (j|k|kret)probes
- Any network high level interaction (system call):
  Listen, connect, accept, close (TCP and/or UDP):
  Local_IP:port action [distant_IP:port]
- Add uid, pid information, executable name
  → links network events and users
- Logs in kernel logs (debug level)
  → Usually not logged into disk
  → Access to dmesg should be restrict

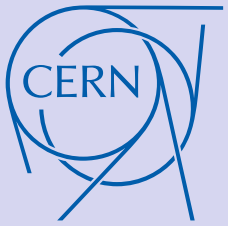





# Sources

## Network logs - host based - Example

```
netlog: [837958.514969]: ??[9464] UDP 127.0.0.1:36572 -> 127.0.0.1:53 (uid=1000)
netlog: [837958.515818]: ??[9464] UDP 127.0.0.1:36572 <!> 127.0.0.1:53 (uid=1000)
netlog: [837958.516249]: ??[9464] UDP 37.187.17.50:50628 -> 74.125.24.94:80 (uid=1000)
netlog: [837958.516268]: ??[9464] UDP 37.187.17.50:50628 <!> 74.125.24.94:80 (uid=1000)
netlog: [837958.516361]: ??[9464] UDP [2001:41d0:a:1132::1]:35540 -> [2a00:1450:400b:c02::5e]:80 (uid=1000)
netlog: [837958.516371]: ??[9464] UDP [2001:41d0:a:1132::1]:35540 <!> [2a00:1450:400b:c02::5e]:80 (uid=1000)
netlog: [837958.516594]: ??[9464] TCP [2001:41d0:a:1132::1]:32875 -> [2a00:1450:400b:c02::5e]:80 (uid=1000)
netlog: [837958.577899]: ??[9464] TCP [2001:41d0:a:1132::1]:32875 <!> [2a00:1450:400b:c02::5e]:80 (uid=1000)
```





# Sources

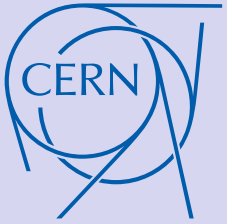
## Network logs - host based - Example

### /usr/bin/curl !

```
[837958.499946] execlog: [uid:1000 pid:9464 sid:4702 tty:pts7 filename:/usr/bin/curl]: curl www.google.fr
```

### strace:

```
socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 3
connect(3, {sa_family=AF_INET, sin_port=htons(80), sin_addr=inet_addr("74.125.24.94")}, 16) = 0
getsockname(3, {sa_family=AF_INET, sin_port=htons(52605), sin_addr=inet_addr("37.187.17.50")}, [16]) = 0
close(3)                                = 0
socket(PF_INET6, SOCK_DGRAM, IPPROTO_IP) = 3
connect(3, {sa_family=AF_INET6, sin6_port=htons(80), inet_pton(AF_INET6, "2a00:1450:400b:c02::5e", &sin6_addr),
 sin6_flowinfo=0, sin6_scope_id=0}, 28) = 0
getsockname(3, {sa_family=AF_INET6, sin6_port=htons(33626), inet_pton(AF_INET6, "2001:41d0:a:1132::1",
 &sin6_addr), sin6_flowinfo=0, sin6_scope_id=0}, [28]) = 0
close(3)                                = 0
```



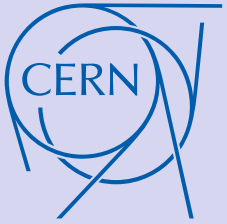



# Sources

## Execution logs

Snoopy or Kernel module 'execlog' (@CERN)

- Every file execution (execve) with arguments
- Add uid, pid, session-id information
- Potentially confidential data/passwords
- Snoopy:
  - LD_PRELOAD 'trick': not a security tool
    $\rightarrow$Can be bypassed
  - Logs into auth syslog
- Execlog:
  - Kernel module in testing @CERN
    $\rightarrow$CanNOT be bypassed
  - Logs in kernel logs (debug level)
    $\rightarrow$ Access to dmesg must be limited






# Sources

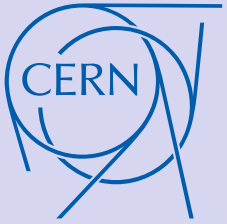
## Execution logs - Example

### dmesg:

```
[ 2142.980426] execlog: [uid:1000 pid:3234 sid:2573 tty:pts2 filename:/usr/bin/uname]: uname -a
[ 2147.401540] execlog: [uid:1000 pid:3235 sid:2573 tty:pts2 filename:/usr/bin/id]: id
[ 2155.013897] execlog: [uid:1000 pid:3236 sid:2573 tty:pts2 filename:/usr/bin/...]: ...
[ 2155.014507] execlog: [uid:0 pid:3236 sid:2573 tty:pts2 filename:/usr/bin/bash]: /usr/sbin/sshd
[ 2157.439922] execlog: [uid:0 pid:3237 sid:2573 tty:pts2 filename:/usr/bin/id]: id
[ 2175.924942] execlog: [uid:0 pid:3240 sid:2573 tty:pts2 filename:/usr/bin/nano]: nano /boot/grub/grub.cfg
```

### syslog:

```
Oct 06 14:28:11 lerya-bis kernel: execlog: [uid:1000 pid:3234 sid:2573 tty:pts2 filename:/usr/bin/uname]:
 uname -a
Oct 06 14:28:15 lerya-bis kernel: execlog: [uid:1000 pid:3235 sid:2573 tty:pts2 filename:/usr/bin/id]: id
Oct 06 14:28:23 lerya-bis kernel: execlog: [uid:1000 pid:3236 sid:2573 tty:pts2 filename:/usr/bin/...]: ...
Oct 06 14:28:23 lerya-bis kernel: execlog: [uid:0 pid:3236 sid:2573 tty:pts2 filename:/usr/bin/bash]:
 /usr/sbin/sshd
Oct 06 14:28:25 lerya-bis kernel: execlog: [uid:0 pid:3237 sid:2573 tty:pts2 filename:/usr/bin/id]: id
Oct 06 14:28:34 lerya-bis kernel: execlog: [uid:0 pid:3239 sid:2573 tty:pts2 filename:/usr/bin/uname]:
 uname -a
Oct 06 14:28:44 lerya-bis kernel: execlog: [uid:0 pid:3240 sid:2573 tty:pts2 filename:/usr/bin/nano]:
 nano /boot/grub/grub.cfg
```

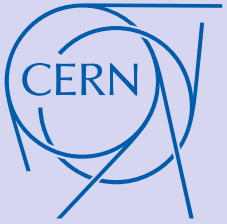





# Sources

## Execution logs - Example 2

```
host1 sshd: Accepted login for baduser from badip
 host1 snoopy: cat .history
 host1 snoopy: grep ssh
 host1 snoopy: ssh -X baduser@somewhere
 host1 snoopy: ssh -X baduser@somewhere
    host2 sshd: Accepted login for baduser from somewhere
     host2 snoopy: cat .history
     host2 snoopy: grep ssh
     host2 snoopy: ssh ...
     host2 snoopy: cat known_host
     host2 snoopy: ssh ...
     ...
    host2 sshd: Session closed for baduser
 host1: snoopy ssh ...
 ...
host1 sshd: Session closed for baduser
```



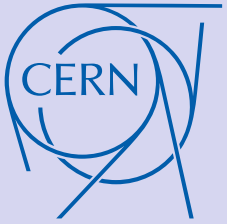



# Sources

## Audit logs

Auditd (Linux Auditing System)

- Built-in solution
- High granularity configuration (man audit.rules):
  - System calls
  - FileSystem access to given files, folders
- Can send logs over network
- aureport & ausearch



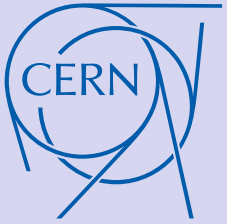



# Sources

## Kernel module loading

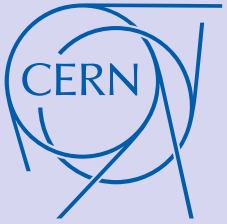
Kernel module 'Dresden' (@CERN)

- Blocks loading of [non-signed] kernel modules
- Logs in kernel logs (emergency level) failed attempts






# Contents



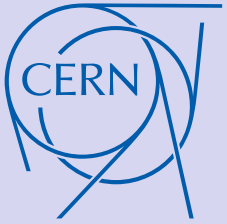



# Analysis

## Secure Shell (SSH)

Users rarely change their habits:

- Track IP source per user:
  - Save recent origin (using geo-localisation/provider)
  - @CERN: notify user on each new location
- Look for technology change:
  - SSH key $\rightarrow$ password
  - Kerberos $\rightarrow$ password
- Combine those two information (escalation?)

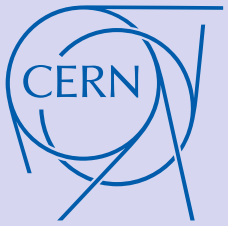





# Analysis

## Kernel Logs

- Kernel OOPS/PANIC:
  - Root escalation tries
  - Kernel tempering
- Modules blocked
  - Root escalation tries
  - Bug/Improper security measure/Unknown behavior

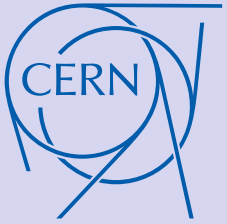





# Analysis

## Network traffic

- Watch *known* IPs
- Backlog on recent incident:
  - Network logs: traffic from/to compromised hosts
  - Host logs: users using compromised hosts

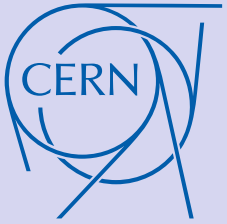





# Canaries

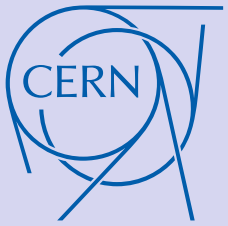
Monitor your monitoring

- Verify that the logging system is still alive
- Generate & filter false positives
- Look for missed generated events






# Contents






# Integrity checks

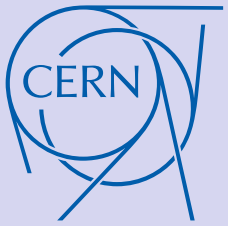
## Packages

Verify package Integrity: **rpm -Va**

- Detects malicious modification of installed RPMs
- Compatible with pre-linking
- False positives: configuration changes

→ Cronjob with filtered output:

- Ignore configuration
- Look for binary size/hash change (S,5)
- Look for link change (L)






# Integrity checks

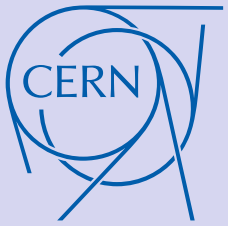
## Libraries

Attacker can use malicious libraries:

- Add a library with a bigger version in lib folders
  → Detected by **rpm -Va**:

```
$ ls -al /lib/libc[.-]*
-rwxr-xr-x. 1 root root 1902708 Aug 28 09:29 /lib/libc-2.12.so
-rwxr-xr-x. 1 root root 1902708 Oct  4 14:05 /lib/libc-2.19.so
lrwxrwxrwx. 1 root root      12 Oct  4 14:05 /lib/libc.so.6 -> libc-2.19.so
$ rpm -V glibc
....L....     /lib/libc.so.6
```

- Inject libraries (LD_PRELOAD):
  - Monitor /etc/ld.so.preload
  - Look for LD_PRELOAD environment variables
  - Monitor /etc/ld.so.conf(.d)?





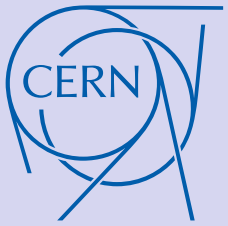

# Integrity checks

## Installed signatures/packages

- Attacker can install a malicious RPM
- **rpm -Va** does not verify this
  → Verify installed packages and their signature
- Attacker can add a key to keyring
  → Monitor keys

Get installed RPM keys:

```
rpm -qa --qf '%{name}: %|RSAHEADER?{%{RSAHEADER:pgpsig}}:{%|DSAHEADER?{%{DSAHEADER:pgpsig}}:{(none)}|}|\n'
```



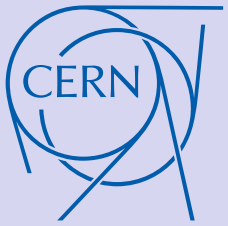



# Integrity checks

## Configuration files

**rpm -Va** does not account for configuration changes

- Use a central configuration system (e.g. puppet, quattor)
- Monitor hash of configuration files (e.g. AIDE)

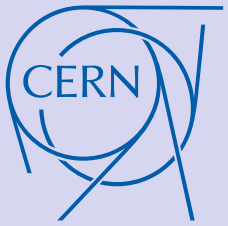





# Integrity checks

## Kernel modules

- Attacker can benefit from kernel modules:
  - Load malicious module
  - Abuse auto-loading feature
- Limit/block kernel loading:
  - Restrict to signed kernel modules (boot option or Dresden)
  - Block kernel loading (Dresden)
- Monitor which modules are loaded:
  - /proc/modules
  - /sys/module






# Integrity checks

## Forensics friendliness

Checking system destroy forensics evidences:

- Basic hashing reads file:
  $\rightarrow$ atime modification
- prelink undo/redo modifies binary
  $\rightarrow$ ~~mtime~~ ctime modification

$\rightarrow$ remount with **noatime** during checks?

Thank your for your attention

# Any questions?