Detection 1

Logging and Traceability

who, what, when, where, how ... why?

David Crooks
UKRI STFC
EGI CSIRT/IRIS Security team
david.crooks@stfc.ac.uk

Introduction

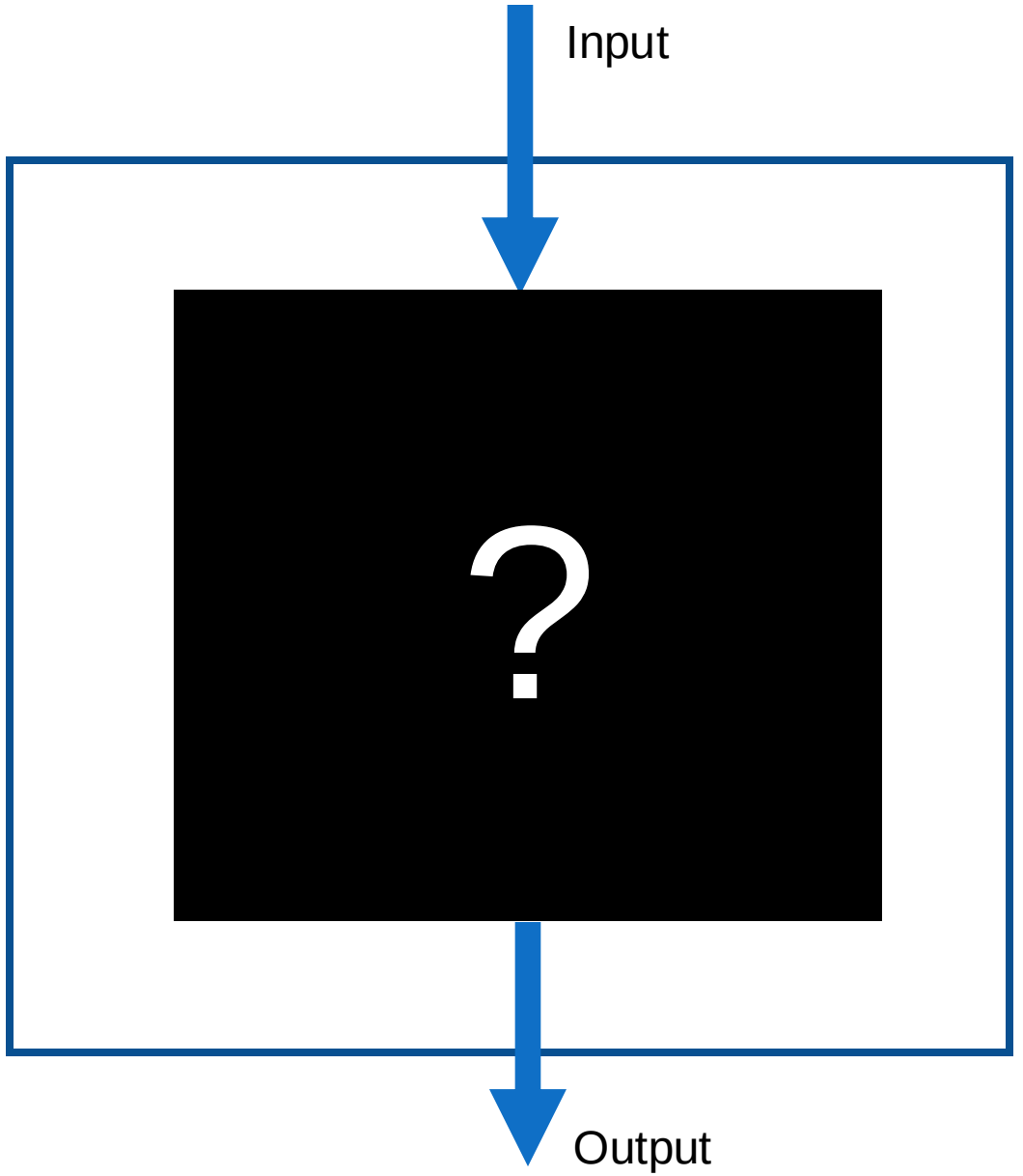
- Logging basics
- Central logging
- Data Protection
- Network logging

Why do we log?

- To know what happened **in as much detail as necessary**

- Often, security concerns are an extension of operations
  - **What** happened?
  - **When** did it happen?
  - **Where** did it happen?

  - **How** did it happen?
  - **Why** did it happen?



Examples

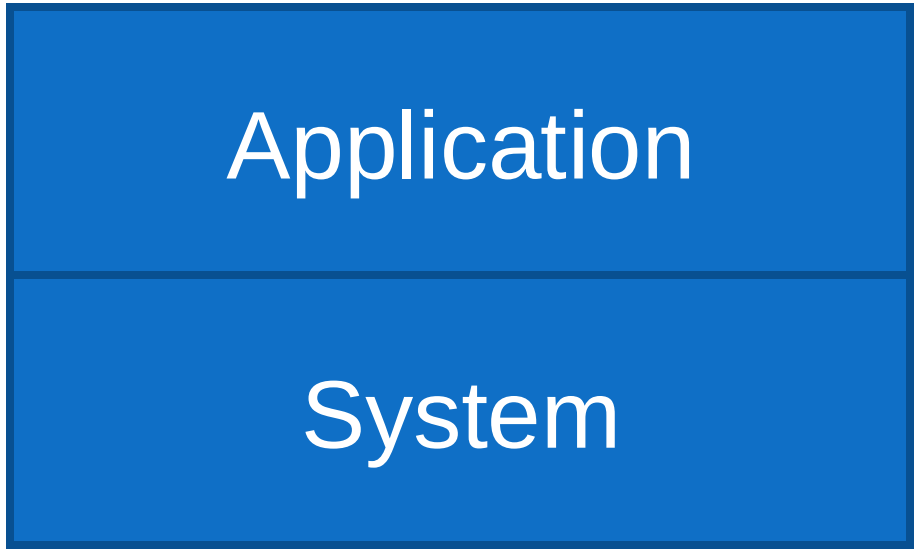
- Why did this data transfer fail?
- Why did this job only complete partially?
- Which endpoints were involved in this process?
- What did the attacker do?

Day to day life

- Logs are an integral part of our technical lives
- This ubiquity comes careful consideration

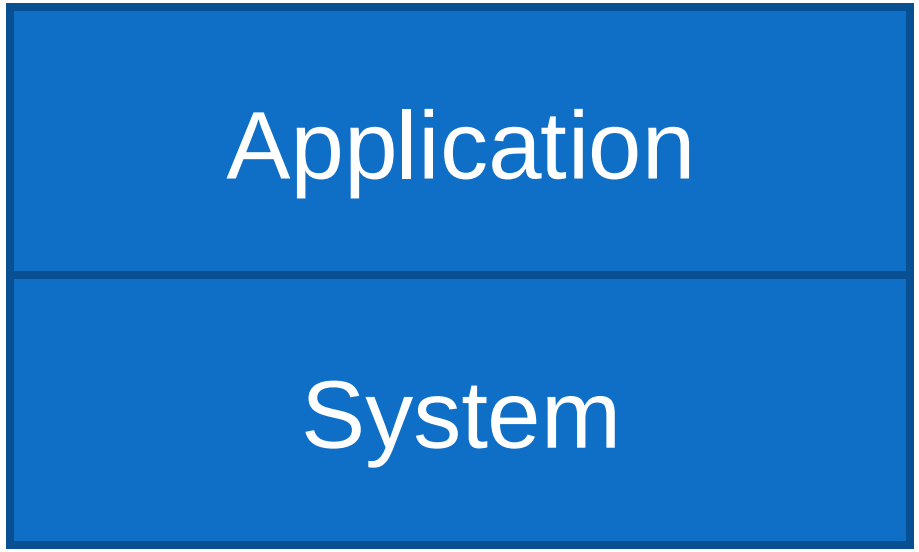
Host/service logs

- Application logs
- System logs



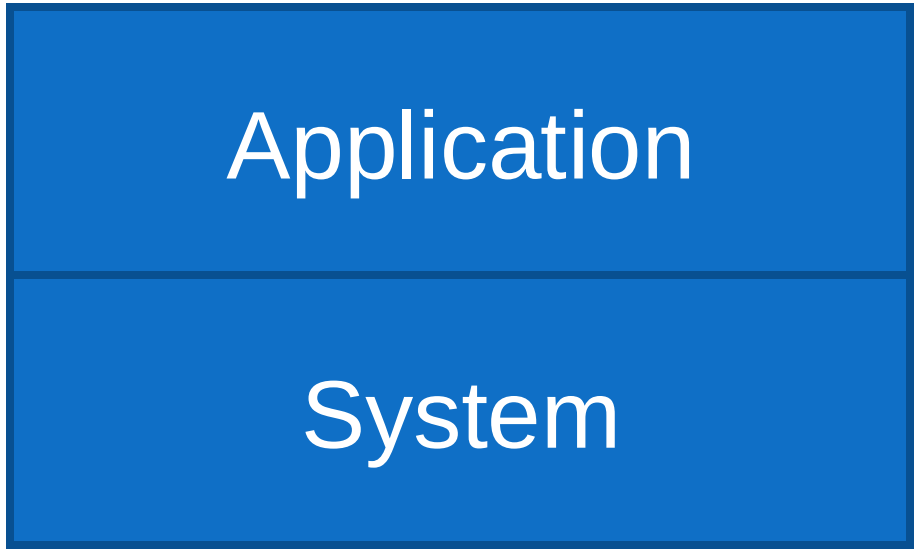
Host/service logs

- Application logs
  - Apache
  - Drupal
  - Ceph
  - Dcache
  - ...



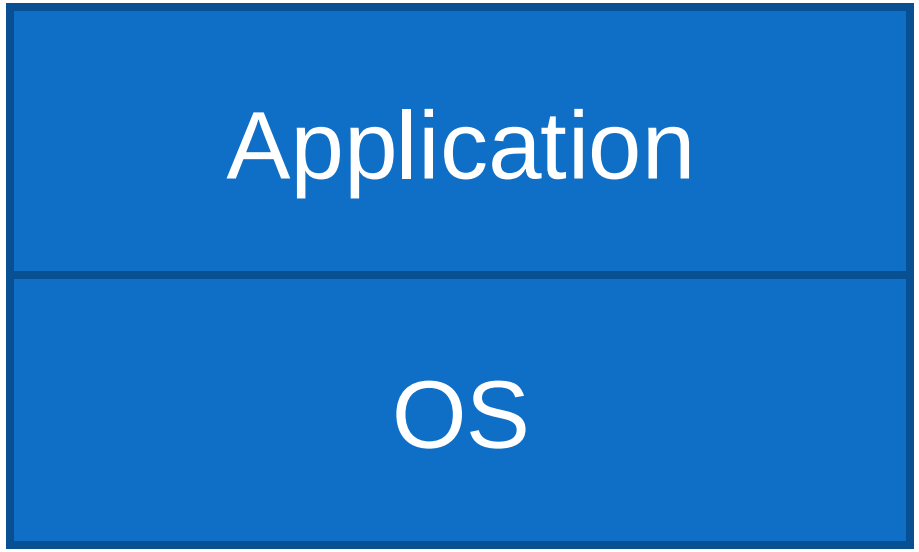
Host/service logs

- Application logs
- These depend on the service
- Talk about this again in traceability, but: service owners are best placed to understand what is useful!



Host/service logs

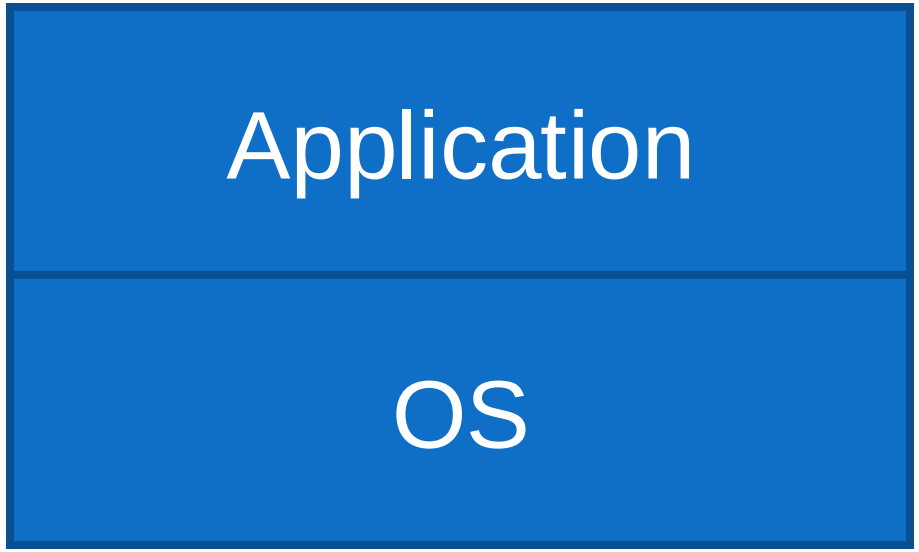
- System logs
- Give us an understanding of the behaviour of the system itself
  - Direct access via `ssh`
  - System behaviour
  - Auditing over time
- (Paths will be for RHEL Distros)



Host/service logs

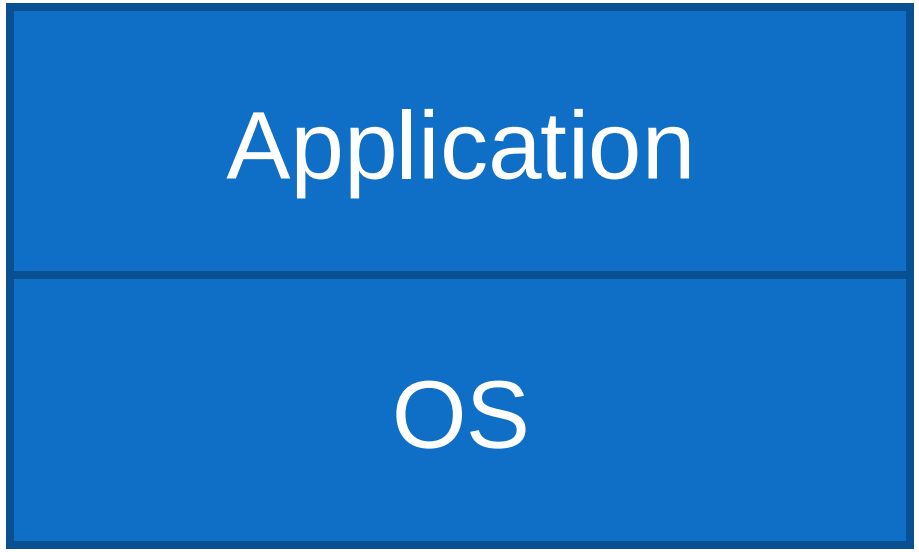
- System logs
  - `/var/log/audit.log`

```
type=USER_AUTH msg=audit(1655751006.984:3758): pid=26347 uid=0 auid=4294967295 ses=4294967295
subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=pubkey_auth rport=35186 acct="centos" exe="/usr/sbin/sshd" hostname=?
addr=A.B.C.D terminal=? res=success'
type=USER_AUTH msg=audit(1655751006.984:3759): pid=26347 uid=0 auid=4294967295 ses=4294967295
subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=key algo=ssh-rsa size=4096
fp=SHA256:48:43:a1:08:47:36:a3:69:1a:d0:72:24:58:f3:e3:07:7d:99:ce:0b:bd:d5:cd:fb:10:bc:37:18:cf:f8:4a:a4 rport=35186 acct="centos"
exe="/usr/sbin/sshd" hostname=? addr=A.B.C.D terminal=? res=success'
type=USER_ACCT msg=audit(1655751006.994:3760): pid=26347 uid=0 auid=4294967295 ses=4294967295
subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=PAM:accounting grantors=pam_unix,pam_localuser acct="centos"
exe="/usr/sbin/sshd" hostname=X.Y.Z addr=A.B.C.D terminal=ssh res=success'
type=CRYPTO_KEY_USER msg=audit(1655751006.994:3761): pid=26347 uid=0 auid=4294967295 ses=4294967295
subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=destroy kind=session fp=? direction=both spid=26348 suid=74 rport=35186
laddr=A.B.C.D 6 lport=22  exe="/usr/sbin/sshd" hostname=? addr=A.B.C.D terminal=? res=success'
type=USER_AUTH msg=audit(1655751006.996:3762): pid=26347 uid=0 auid=4294967295 ses=4294967295
subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=success acct="centos" exe="/usr/sbin/sshd" hostname=? addr=A.B.C.D 6
terminal=ssh res=success'
type=CRED_ACQ msg=audit(1655751006.996:3763): pid=26347 uid=0 auid=4294967295 ses=4294967295
subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_unix acct="centos" exe="/usr/sbin/sshd"
hostname=X.Y.Z addr=A.B.C.D terminal=ssh res=success'
type=LOGIN msg=audit(1655751006.996:3764): pid=26347 uid=0 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 old-auid=4294967295
auid=1000 tty=(none) old-ses=4294967295 ses=215 res=1
type=USER_ROLE_CHANGE msg=audit(1655751007.128:3765): pid=26347 uid=0 auid=1000 ses=215 subj=system_u:system_r:sshd_t:s0-
s0:c0.c1023 msg='pam: default-context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 selected-
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 exe="/usr/sbin/sshd" hostname=X.Y.Z addr=A.B.C.D terminal=ssh
res=success'
type=USER_START msg=audit(1655751007.145:3766): pid=26347 uid=0 auid=1000 ses=215 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023
msg='op=PAM:session_open
grantors=pam_selinux,pam_loginuid,pam_selinux,pam_namespace,pam_keyinit,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_lastbg
acct="centos" exe="/usr/sbin/sshd" hostname=X.Y.Z addr=A.B.C.D 6 terminal=ssh res=success'
```



Host/service logs

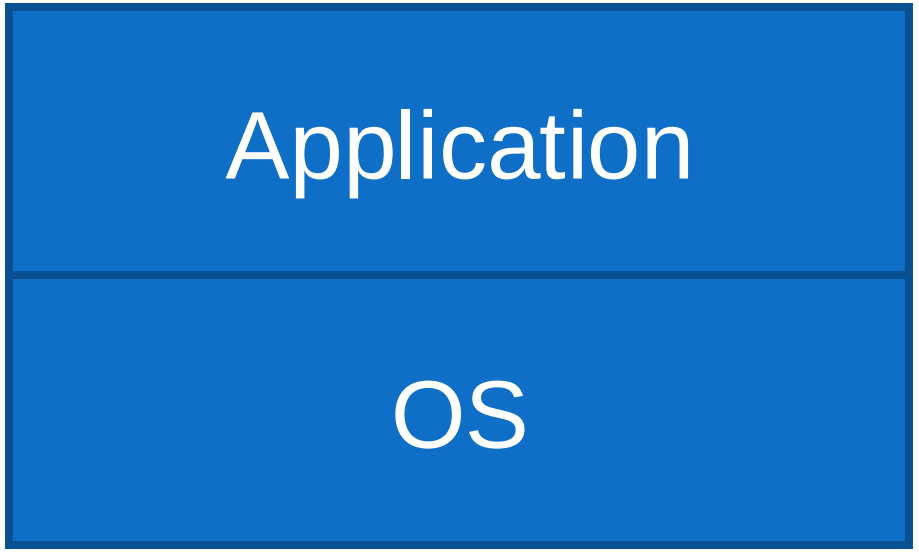
- System logs
  - `/var/log/audit.log`
- `aureport` can be used to get summary information



Host/service logs

- System logs
  - `/var/log/audit.log`

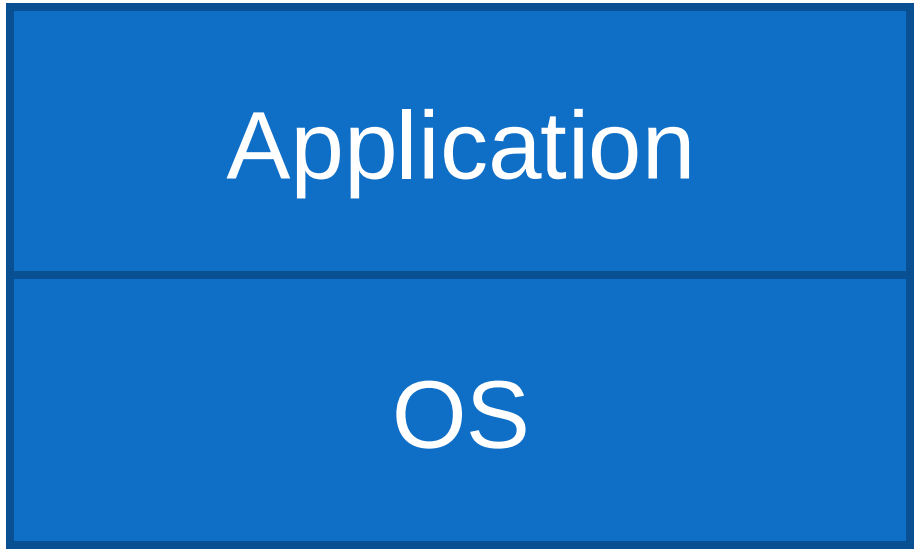
```
Summary Report
======================
Range of time in logs: 01/01/70 01:00:00.000 - 21/06/22 07:46:12.034
Selected time for report: 01/01/70 01:00:00 - 21/06/22 07:46:12.034
Number of changes in configuration: 0
Number of changes to accounts, groups, or roles: 0
Number of logins: 3
Number of failed logins: 0
Number of authentications: 9
Number of failed authentications: 0
Number of users: 2
Number of terminals: 5
Number of host names: 4
Number of executables: 4
Number of commands: 2
Number of files: 0
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 0
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 35
Number of integrity events: 0
Number of virt events: 0
Number of keys: 0
Number of process IDs: 21777
Number of events: 164767
```



Host/service logs

- System logs
  - Auditbeat

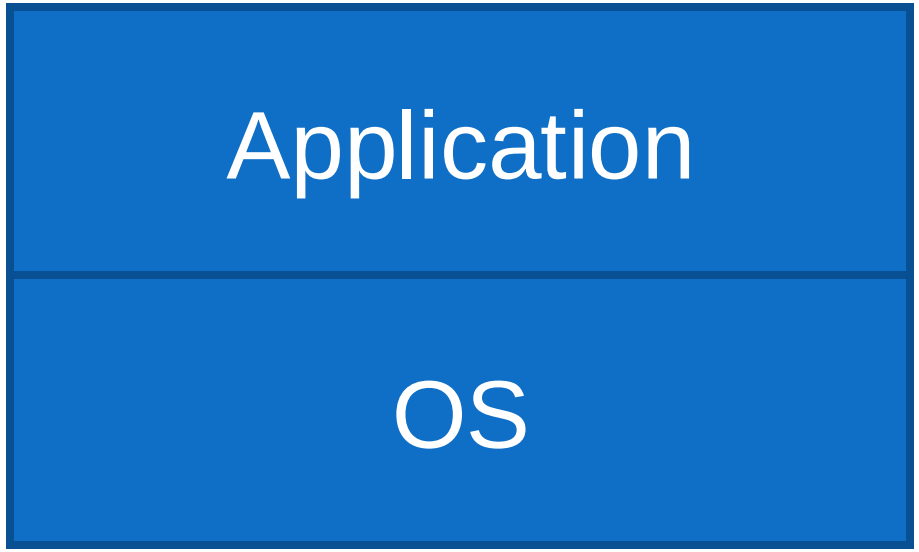
- Part of the elasticsearch Beats set of tools that can also extract and effectively parse audit data



Host/service logs

- System logs
  - `/var/log/messages`

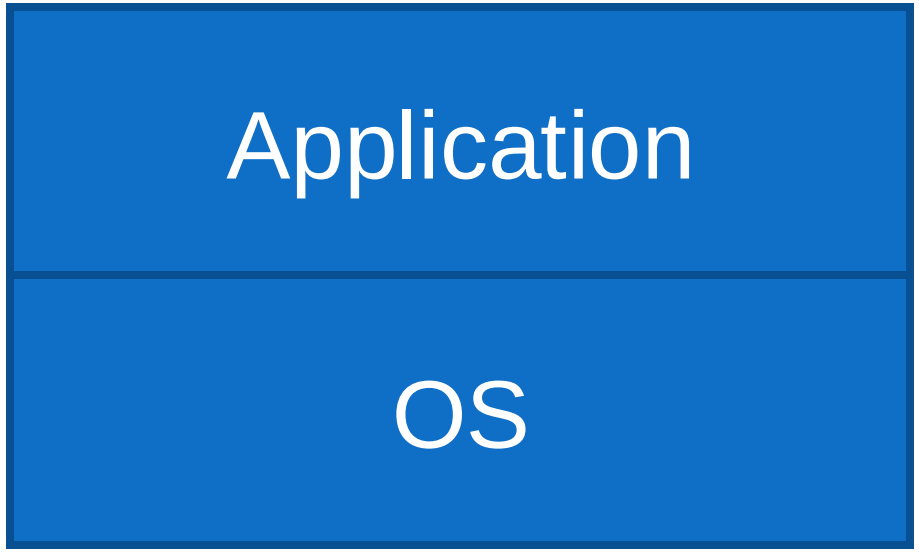
Records global log messages, system notifications including those during boot



Host/service logs

- System logs
  - `/var/log/secure`

Records successes and failures for users using `ssh` to access the system

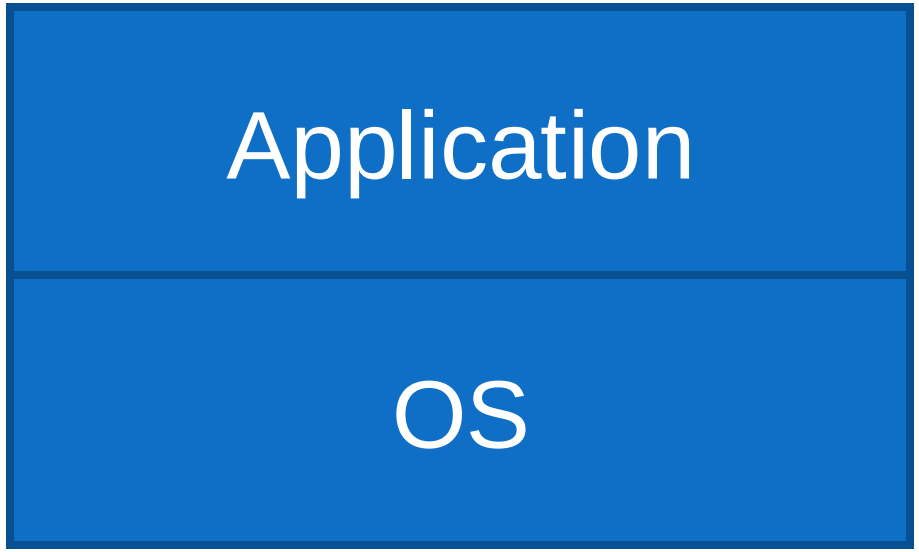


Host/service logs

- System logs
  - `/var/log/secure`

```
Jun 19 22:18:36 hostname
sshd[26877]: Accepted
publickey for user from
A.B.C.D port 60096 ssh2: RSA
SHA256:…
```

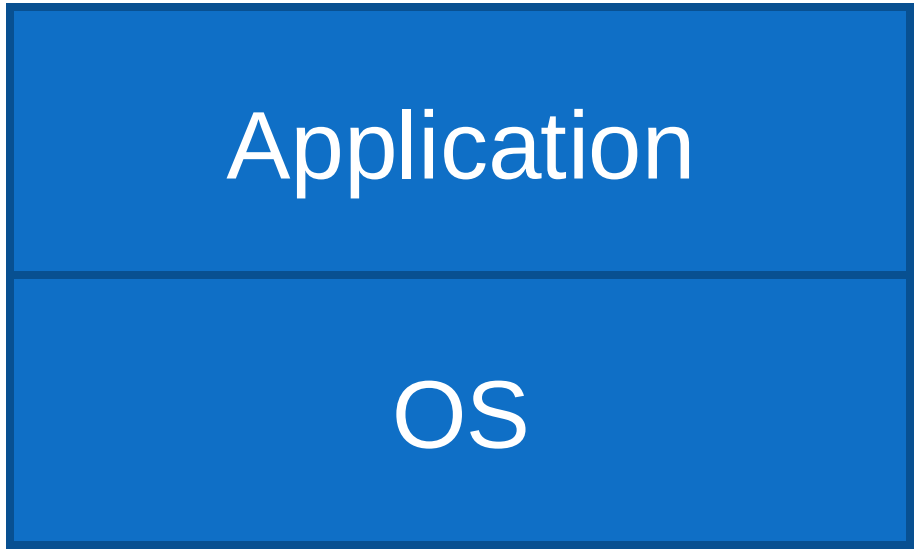
Success!



Host/service logs

- System logs
  - `/var/log/secure`

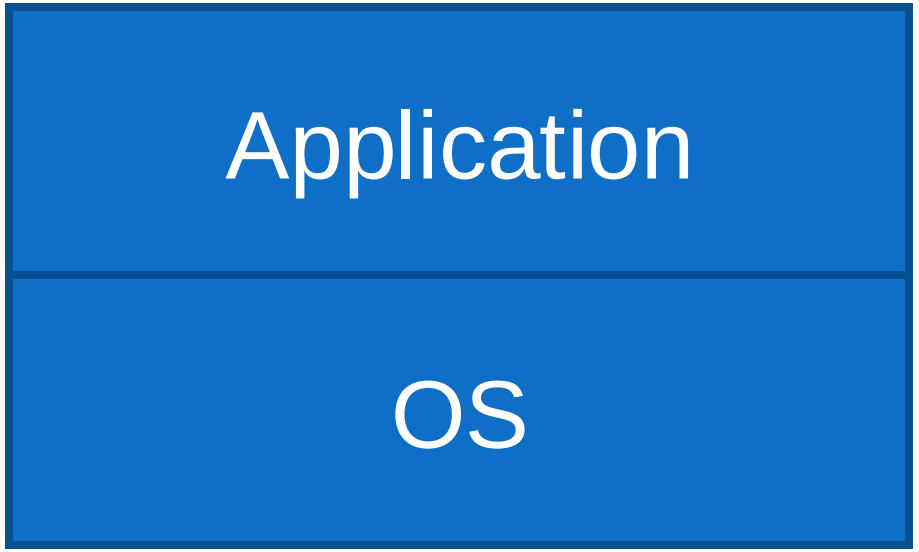
```
Jun 20 19:08:58 hostname
sshd[7555]: Invalid user admin
from A.B.C.D port 36844
```



Host/service logs

- System logs
  - `/var/log/secure`

```
Jun 20 19:08:58 hostname
sshd[7555]: Invalid user admin
from A.B.C.D port 36844
```



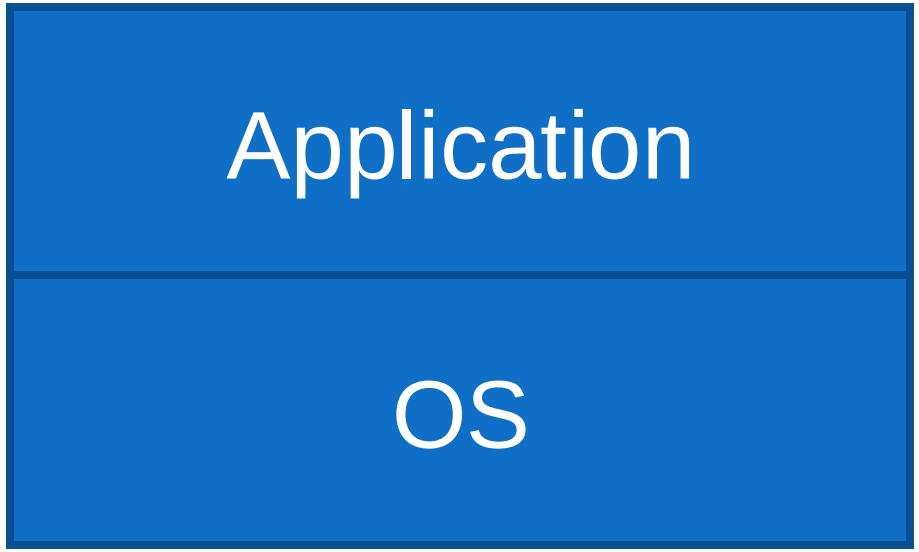
Host/service logs

- System logs
  - `/var/log/secure`

… this is why you harden your systems (although only a *real* problem if they succeed)

A primary source of checking for malicious access

Unless?



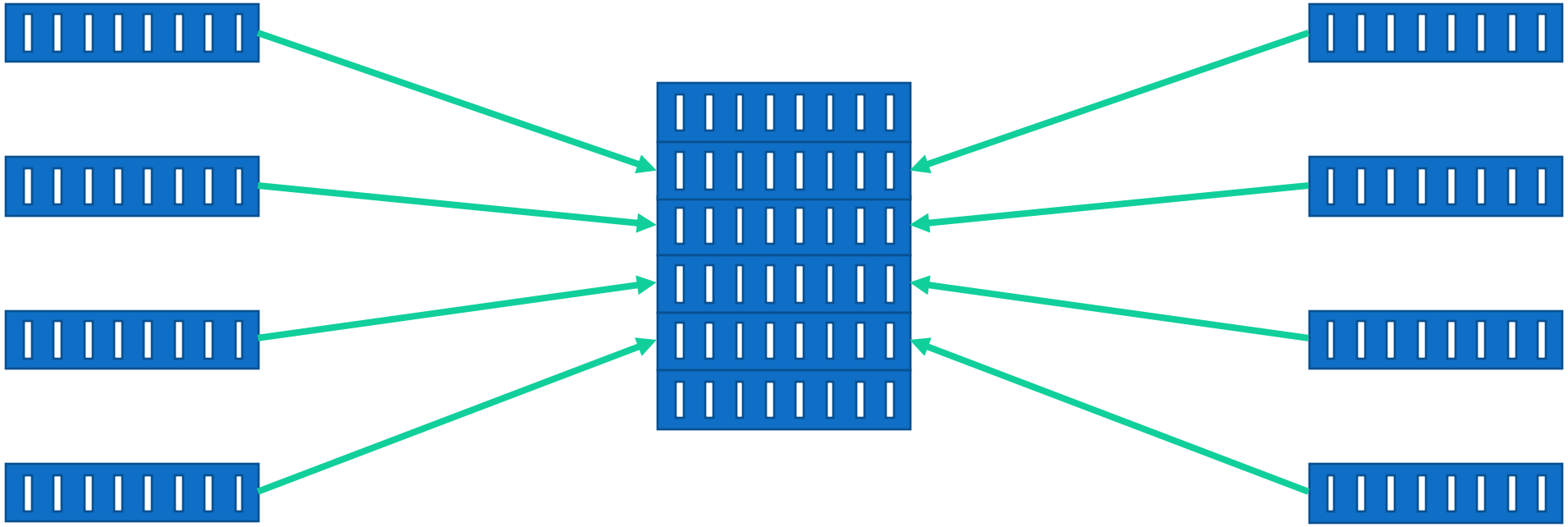
A successful attacker

- Gains access via a weak password (`password2023-2`)
- Installs a compiler, builds some code…
- … hides their tracks by truncating the logs

Central logging

- Logs are data
- Vulnerable to deletion or corruption
- Back them up!

Central logging

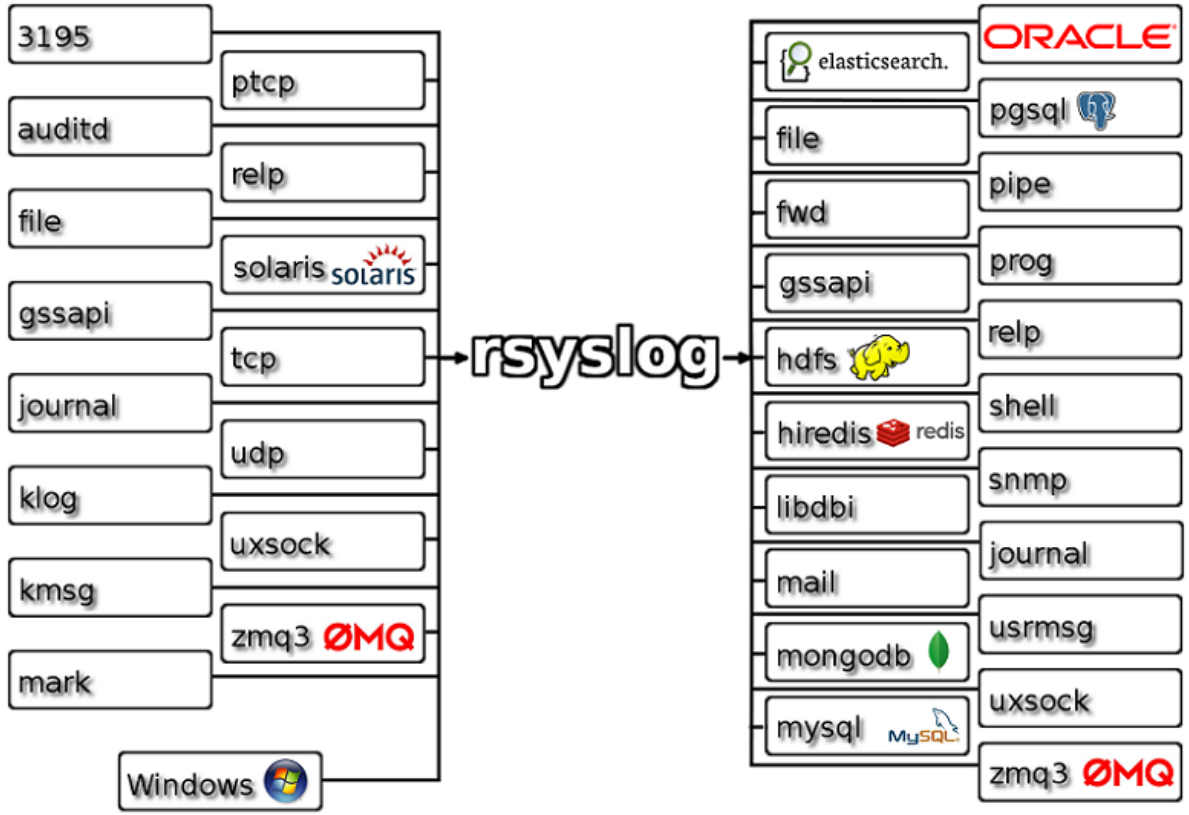


Central logging

- **One of the single most important things to do for the security of a service**
- Helps incident response
- Helps correlate logs between hosts
- Manually searching logs is time consuming and may be less effective
- Tools to aggregate and visualise logs are therefore important

rsyslog

- `rsyslog` is a well-featured logging engine
- `rsyslog` and `syslog-ng` are both feature-rich successors to the original syslog



https://www.rsyslog.com

rsyslog and other tools

- Especially at this point, storing raw logs is not the most useful
- Use a tool like elasticsearch to allow better searching an querying of the data

OSSEC/Wazuh

- OSSEC is a very nice host-based IDS that will aggregate logs in a server/client topology
- Customisable rules
- Very flexible

https://www.ossec.net

OSSEC/Wazuh

- Wazuh is a modern development of OSSEC that integrates tightly with elasticsearch

- Important when considering defence in depth – having one exactly one tool to monitor your system is **not** optimal (necessary ☺)

wazuh.

https://wazuh.com/

Wazuh/OSQuery

- Wazuh can monitor many useful things at the host level
  - File integrity + checksums
  - Configuration Assessment
  - Extended Detection and Response

- OSQuery is a nice tool that provides an SQL interface to system information

wazuh.

https://wazuh.com/

https://osquery.io

System + application logs

- Discussed some key system logs

- Application logs are best understood by their service owners: how to choose what you need?

System + application logs

- We can’t store an infinite amount of logs

- And we don’t want to

“too much data looks like noise”

Data protection

- I am not a lawyer ☺

Data protection

- We are in an era where individual privacy rights are rightly taken particularly seriously
- This is not something that should hinder our security work

- GDPR
- CERN OC11
- Development of UK data protection laws
- Working with laws in other countries

GDPR and CSIRT activities

- In GDPR and associated findings the exchange of logs for incident response is recognized as a useful activity
- https://www.first.org/blog/20171211_GDPR_for_CSIRTs
- We **do** need to be careful about what we store, why, and for how long

Access controls

- A necessary consideration for logging is that – as we've just discussed – the information could be sensitive

- Tracking who can access logs and how is therefore very important
  - Security services are just as in need of hardening as any other

Log retention

- In LHC computing context, for a long time 90 days was the retention period set by policy

- Now moving towards 180 days or more: why?

Log retention

- The number of incidents that have their beginning many months ago
- Only having logs for 90 or 180 days means we lose visibility
- 12 – or 13 – months is where we might set our sights

Log retention: practical matters

- Of course, there are practical matters
  - Logs take up room
- Central logging **also** makes capacity planning easier
  - Build to a set of services that are logged
- Continuous improvement is important

Log retention: practical matters

- Our architecture will suggest where and how many logs we can keep
- This can and should evolve over time
- Focus on sustainable development

Traceability

- For security, we want the logs that will help us piece together a set of events
  - When did someone gain access?
  - What did they do on the host?
  - Where did they go next?
  - What other hosts did they talk to?

Traceability

- Traceability is the ability for us to trace the activity associated with a particular user and/or particular workflow

- Want to be able to track the entire lifecycle
  - Initiation
  - Primary events
  - (External) communications
  - Closeout

Traceability

- Core system logs are essential; for application logs we want anything that helps piece these together
- Debug logs don’t help with this
- It is likely that this will **also** evolve over time
- Make a plan and iterate based on your risks and resources

Split traceability

- Our the, current circumstances, it is **highly likely** that the logs from a particular service – or even facility – will not be sufficient to track the activity of a user or group

- Why?

Split traceability

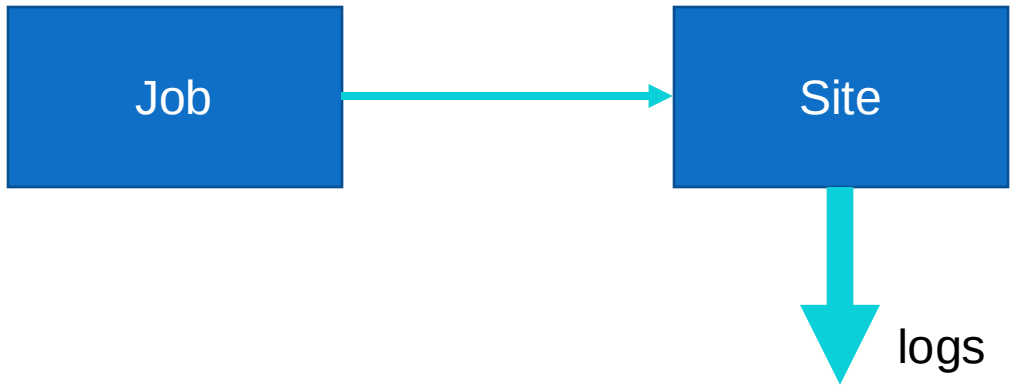
- In research and education, invariably work as part of a bigger infrastructure, federation or federation of federations

Split traceability

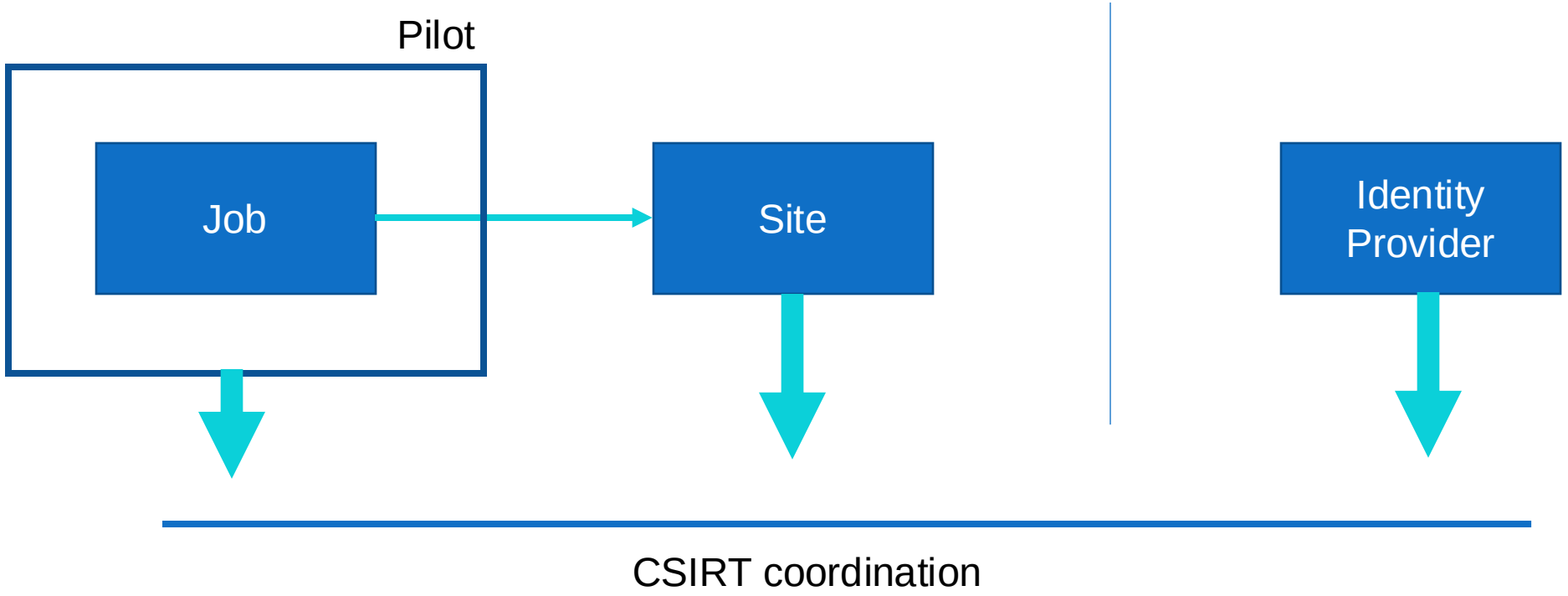
- Many (most!) of our activities involve many services composed together
  - WLCG pilot jobs
  - **Cloud services**

- We can **no longer** rely on the logs on a single host/in a single facility to assemble the full picture of a user’s activity

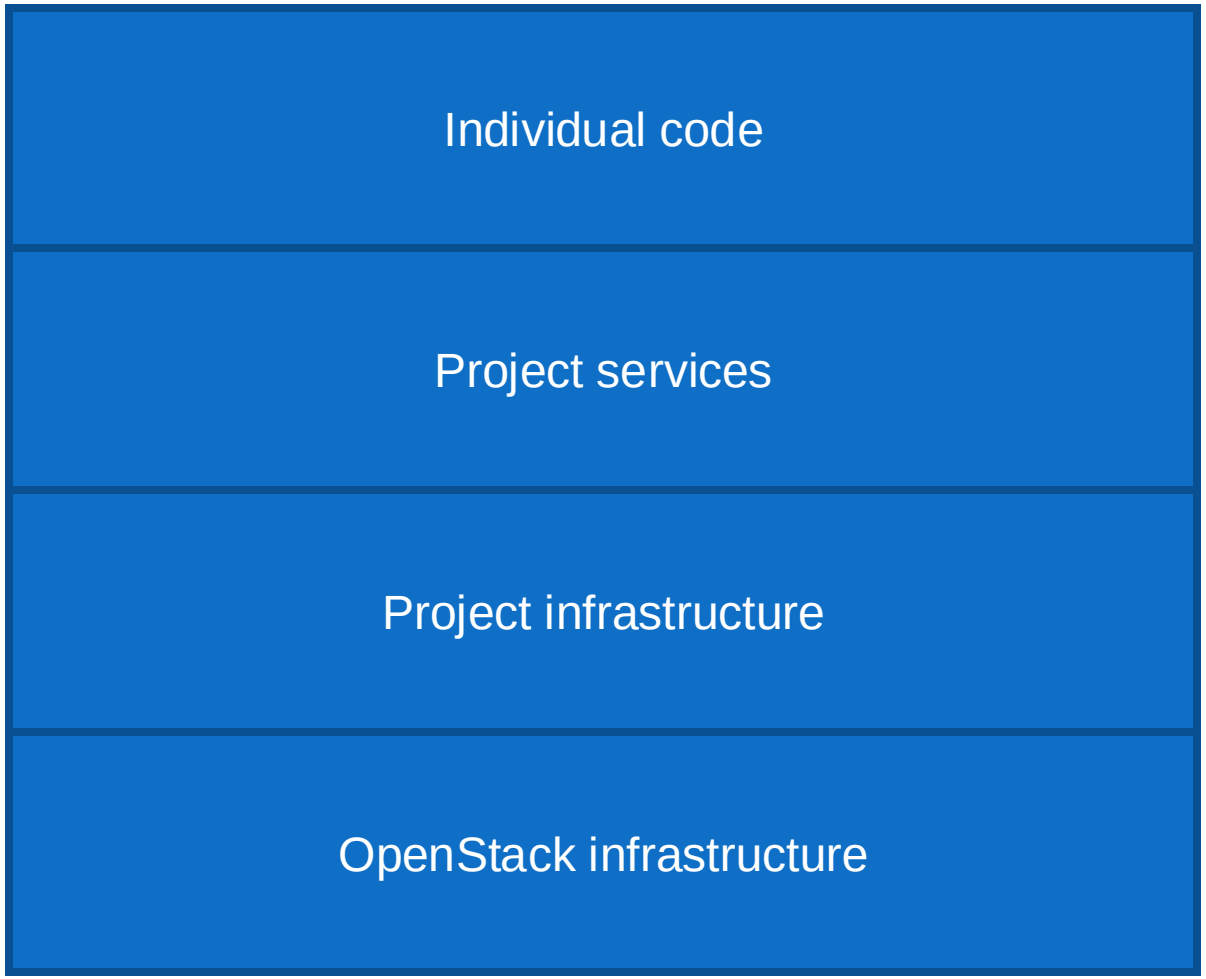
LHC case study: grid jobs: historical



LHC case study: pilot jobs: after



Cloud services



How do we evaluate our traceability?

- Planning and policy
- Collaboration and cooperation
- **Testing**
- Find use cases that are appropriate for you and try them out!

Network logging

- We’ve talked about host based logs
- What’s happening on the network?

Sources of network logs

- Routers
- Host-based generators
- Monitoring

Netflow and sflow

- Netflow and sflow are different but similar methods of storing **metadata** about network connections
  - Endpoints/duration/…
- Most switches we'll use will generate one or the other
- Can generate on-host
  - `hsflowd`

Netflow came from Cisco

sflow came from InMon

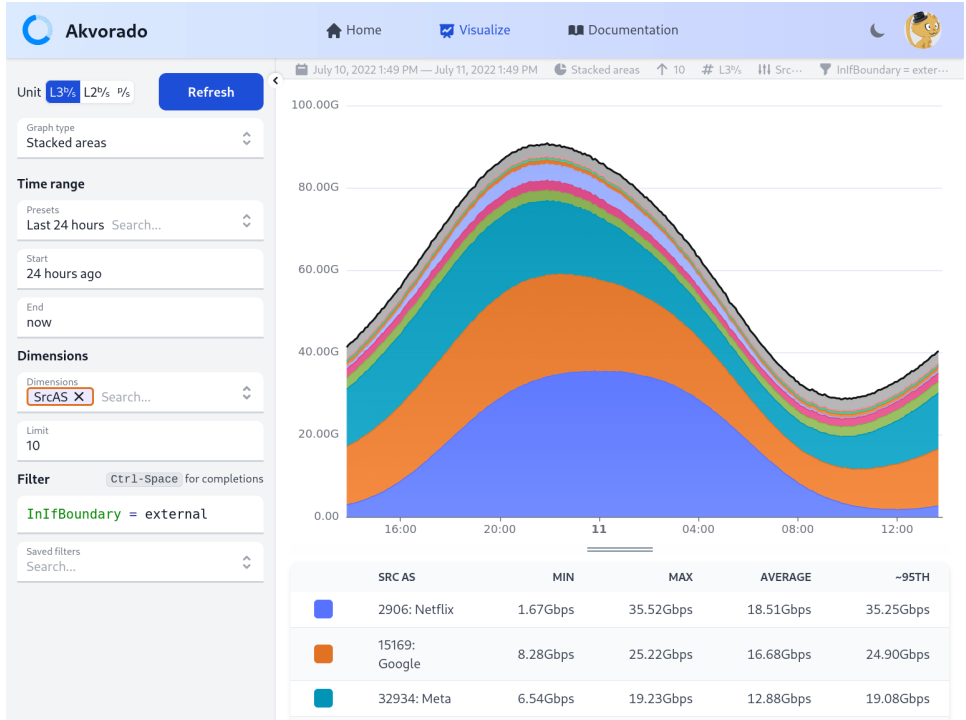
Netflow and sflow

- Pros
  - Ubiquitous
  - Easy to generate
- Cons
  - Sampled
- In general, have **sampled** data from netflow and sflow
- Useful for long term connections but **not forensically useful**

Flow collectors

- Example: Akvorado
- flow collector, enricher and visualizer

- Clickhouse backend
  - Fast and effective



new

Flow collectors

- Work presented at TNC26
- Network Flow Monitoring: An Evaluation of Modern Collector Software

	nfdump	SiLK	pmacct	IPFIXcol2	GoFlow v2	Akvorado
Initial release	2004	2003	2003	2014 (v1) 2018 (v2)	2018 (v1) 2021 (v2)	2022
License	BSD	GNU GPL 2.0-style	GPL	BSD 3-Clause, GPL 2	BSD 3-Clause	AGPL-3.0
Language	C	C	C	C++	Go	Go
Maintained by	Peter Haag Private person	CERT NetSA Carnegie Mellon University	Paolo Lucente Private person	CESNET Czech NREN	Louis P. Private person	Free French ISP
Focus	Historic data analyzation	Historic data analyzation	Network monitoring toolkit	Modular plugin-based flow processing	Flow processing pipeline entry point	All-in-one pipeline

Netflow and sflow

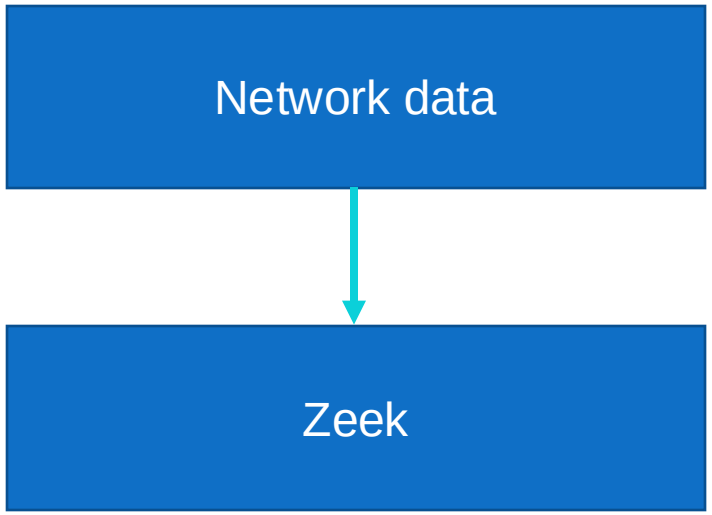
- Netflows are especially useful at a high level
  - NRENS
- You **can** produce 1:1 data, but…

Deep Packet Inspection

- Using a tool that analyses every packet it sees will yield rich information
  - Metadata
  - File information
  - Certificate information…
- Can’t see inside encrypted streams unless you do decryption

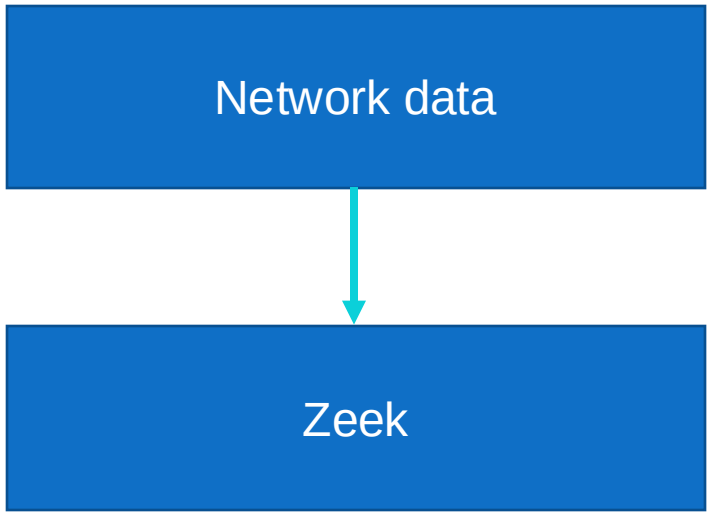
Zeek

- Zeek is an example of a current network IDS in broad use in the US and EU
  - Ingest data by taking tap of network traffic
  - Optical, port spanning or packet broker
- Single threaded, works by running a set of scripts against each packet
  - Scale out by building a zeek batch farm



Zeek

- This gives us forensic level results
  - Every packet is tracked
- But this is computationally expensive
  - Need care in choosing deployment
- More on this soon



Conclusions

- We need to retain logs that describe the activity of our users and services
  - For long enough to perform forensics
  - Following our legislation
  - Pragmatically for our environment

Conclusions

- We can identify logs in our services that will help with this
  - System
  - Application
- We can centrally log these
  - And **should!**

Conclusions

- We need to consider traceability in being able to piece together the events related to a particular user or activity

- This is **very likely** to require composition with other sources including other sites

Conclusions

- In addition to host-based logs, we can also log traffic from our networks
- {Net,s}flow generally give sampled, high-level metadata
- Deep packet inspection gives us greater, forensic detail
  - But more computationally intensive

Questions?