UKRI Science and Technology Facilities Council

Securing the RAL Site

Image © STFC Alan Ford

Outline

- Landsape
- Motivations for RAL
- WLCG SOC WG
- Applying the model to RAL
- Positives/Challenges
- Next Steps

Science and Technology Facilities Council

Landscape

- The current threats for research communities include sophisticated, motivated and well funded actors
- A SOC is a powerful tool for monitoring and alerting threats
- SOC augments security capabilities
- Important to share intelligence within the community

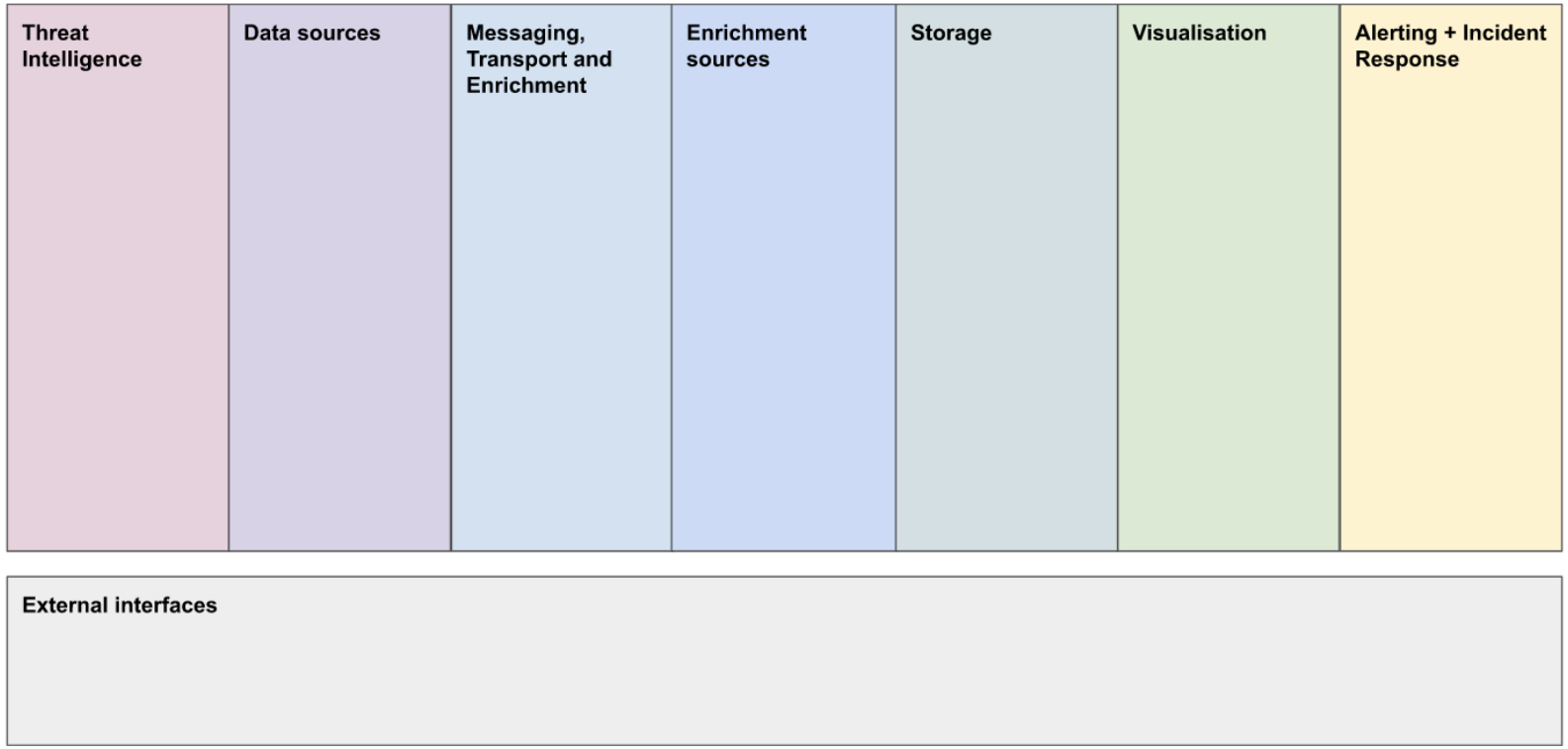
Science and Technology Facilities Council

Implementation at RAL

- General motivation
  - Increase in priority of cybersecurity projects within STFC over last few years
  - Coupled to modern landscape and clear understanding of potential impact
- Specific motivation
  - Work already planned to deploy SOC capabilities for RAL Tier1 following original SOC WG design
  - Opportunity with modest increase in funding to cover entire campus due to network topology
  - Intention both to improve cybersecurity posture for Harwell campus (RAL) and act as pilot for other STFC sites
    - Alongside other SOC development work in the broader UK Research and Innovation context

WLCG SOC WG

- Workshop 14th – 18th August
- Participants from UKRI, University of Durham, CERN, University of Chicago and University of Michigan, JISC
- Docker deployed Misp
- Zeek RPM's for Rocky9/el9
- Documentation

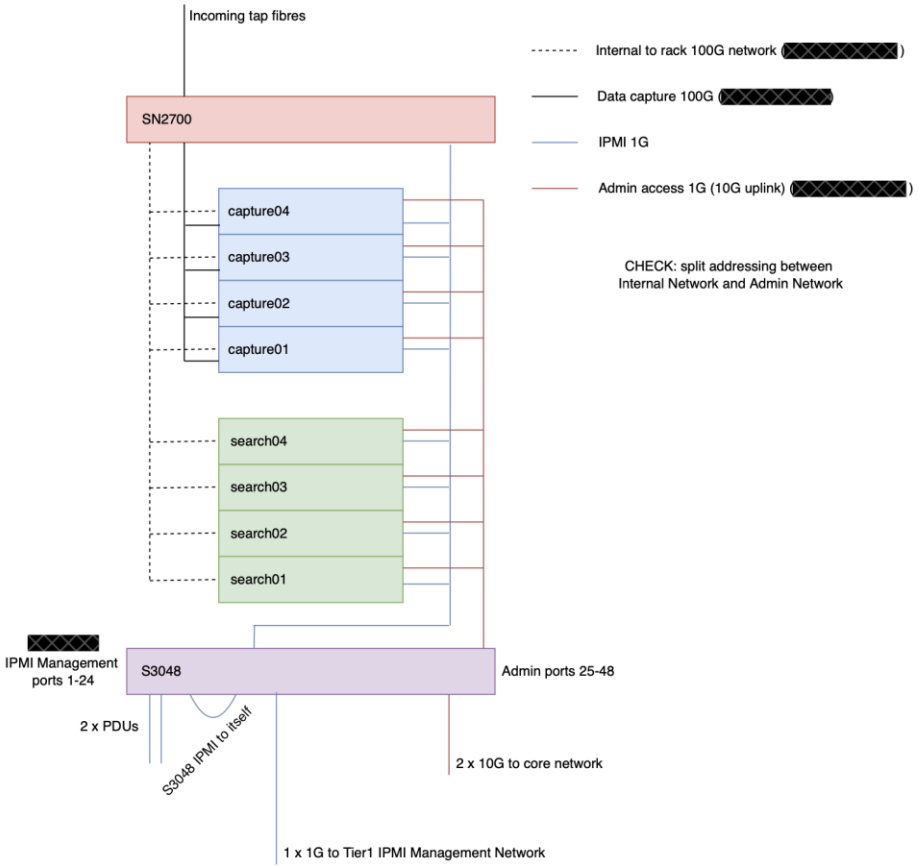


Science and Technology Facilities Council

SOC Hardware Layout

- Admin Network private VLAN behind site firewall
- Internal 100G network
- 4 capture and 4 search nodes

- Some services need to be accessible from the core site network to be useful (dashboards…)
  - Ports opened to loadbalancers and made available from there
  - Other services on the Tier-1 VMWare cluster and have monodirectional traffic from the SOC VLAN



S3048 VLANs

- IPMI: addressing in Tier1 IPMI range ([redacted])
- Admin access: addressing in SOC range ([redacted])

SN2700 Configuration Blocks (assume all 100G)

- 10 ingest ports from optical taps
- 4+ egress ports to capture nodes
- 8 internal high thoughput network ports
- 1 IPMI connection to S3048

Science and Technology Facilities Council

Applying the Model: Monitoring Taps

- High performance 100Gb Mellanox SN2700 switch, running Cumulus OS
  - Ingests the intercepted Janet and OPN network traffic (with separate Rx and Tx links)
  - Load-balances and forwards packets across our Capture hosts for analysis
- Selected ports are grouped into logical interfaces called a ”bonds”, operating in “balance-xor mode”
- A “symmetric” hash value is calculated for each incoming packet, based on source and destination IP address >> result determines which physical port of the bond will egress the packet
  - The direction a packet travels between two hosts is irrelevant in the generation of its hash value
  - Ensures all packets in a given “conversation flow” are consistently routed to the same Capture node in our SOC (crucial for Zeek performance)

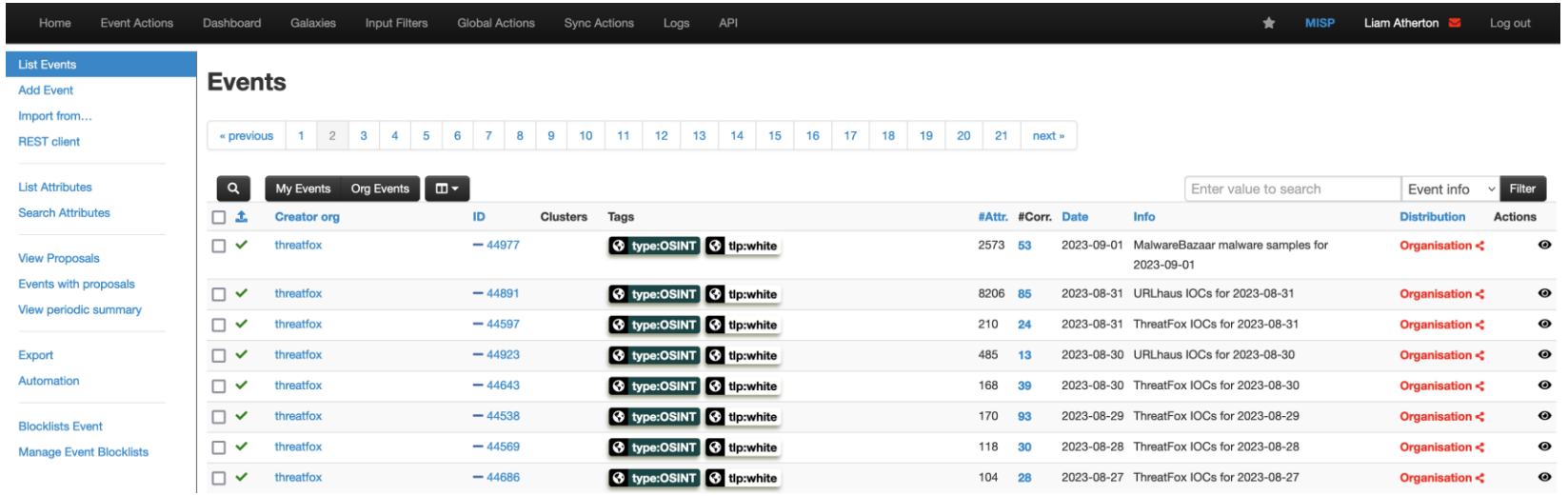
1/2	3/4	5/6	7/8	9/10	11/12	13/14	15/16	17/18	19/20	21/22	23/24	25/26	27/28	29/30	31/32
Cap01 Rx	Cap02 Rx	Cap03 Rx	Cap04 Rx	Rx	Rx	100G→4x25G splitter	100G→4x25G splitter	100G→4x25G splitter			Janet Rx	Janet Rx	Janet Rx	Janet Rx	OPN Rx
Cap01 Tx	Cap02 Tx	Cap03 Tx	Cap04 Tx	Tx	Tx	Blocked	Blocked	Blocked			Janet Tx	Janet Tx	Janet Tx	Janet Tx	OPN Tx

Science and Technology Facilities Council

- “Capture” ports (output to capture01-04): 8 x 100Gig black DAC cables – ports 1-8
- Data ingest of the 10 tapped fibres: 5 yellow fibre tap cables, into 10 x 100Gig transceivers – ports 23-32 (4 Janet links and 1 OPN link)
- Ports for internal high-throughput network: 3 x 100Gig → 25Gig cable splitters – ports 11-16 (1 port used = 1 port blocked)

Applying the Model: Misp

- MISP hosted in a more permissible network environment than the rest of SOC to allow corporate access
  - This is because we don't want routing to SOC rack from outside the department
  - To be hosted in the Tier1 VMWare cluster.
- Deployed using Docker compose, with each component (web, database, modules) in its own container
  - https://github.com/JiscCTI/misp-docker



Science and Technology Facilities Council

Threat Intelligence

Applying the Model: Zeek (Bro)

- Runs on SOC Capture Nodes: (2x Mellanox ConnectX-6 dual 100Gb NICs, 2x AMD 7H12 64-core CPUs, 1TB RAM)
  - Specialized hardware for Zeek monitoring, which we run in ”cluster” mode with 2 main “worker” processes
- Workers each listen on a designated network interface, and reserve processor threads to perform packet analysis
  - Two threads per CPU need to be left free for the host’s OS, Zeek’s manager/proxy/logger, other misc. processes
- Network cards have on board encryption compute power and 63 available “ring buffers” for speeding up the pipeline
  - Packets symmetrically hashed by network card immediately >> result determines which “Rx ring” packet goes to
- We enable 62 of these ring ”channels” per network interface to match 1-to-1 with the thread count of the Zeek workers
  - Channels manage direct storage of packets in system memory, and send out identifying ”interrupt request” (IRQ)
  - IRQs are hardware signals that trigger a CPU response, i.e. fetching a network packet to read (via “socket buffer”)
  - We then map every channel’s IRQ integer id-value to a specific processor thread, connecting the two elements
  - Zeek’s “af_packet” plugin leverages native features of Linux sockets to load balance the fanned-out traffic across multiple processing threads attached to a single worker (standard Zeek workers don’t have multithreading)
  - This feature of processing traffic across multiple NIC hardware queues is called Receiver Side Scaling (RSS)
  - Linux’s irqbalance service, which normally balances IRQs across CPU threads dynamically, needs to be disabled
- This configuration ensures all packets from any given monitored connection end up being processed by the same worker/core, enabling Zeek to benefit from cached memory during analysis of network sessions (increases efficiency and quality of monitoring)
  - Zeek interprets information from extracted content and transaction data
  - Custom scripting feature lets users modify the default criteria and methods of traffic analysis
  - Outputs streamlined, descriptive set of categorized logs

Data sources

- Currently monitoring both OPN RX and TX taps on a single Capture node host
- Random sample snapshot of network levels ⇣ and Zeek's capture_loss.log ⇢

```
[ZeekControl] > capstats    (10s average)
Interface                          kpps       mbps
-------------------------------------
localhost/af_packet::enp33s0f0 1002.4     11803.4
localhost/af_packet::enp33s0f1 658.4      7731.3

Total                          1660.8     19534.7
[ZeekControl] >
```

- Result: avg. capture loss very low, mostly << 0.01%
  - Almost no traffic being dropped by switch or node interfaces, host kernel, etc
  - Still some room to improve certainly
  - When very busy and varied Janet traffic comes into play, greater challenges are expected
  - This will require further analysis, tuning and modification of Zeek + host configuration

Science and Technology Facilities Council

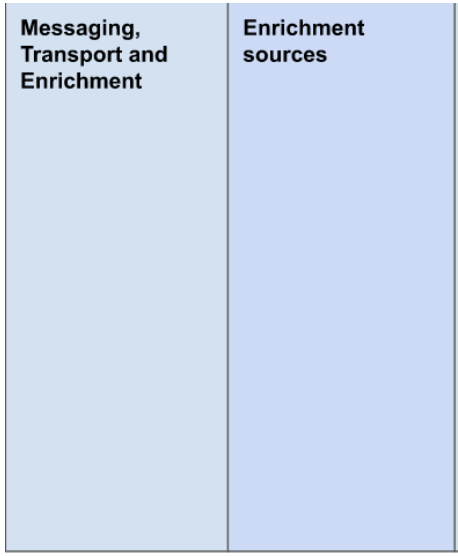
```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   capture_loss
#open   2023-10-15-14-01-18
#fields ts              ts_delta        peer            gaps    acks    percent_lost
#types  time            interval        string          count   count   double
1697374877.883803       900.000128      worker-2-8      2       185139  0.00108
1697374877.903965       900.000055      worker-1-15     0       71369   0.0
1697374877.985283       900.000055      worker-2-15     1       189492  0.000528
1697374877.937105       900.000054      worker-2-10     0       241021  0.0
1697374878.006315       900.000056      worker-2-36     0       88719   0.0
1697374877.836066       900.000055      worker-1-38     1       193079  0.000518
1697374877.882363       900.000057      worker-1-47     5       85392   0.005855
1697374878.144724       900.000054      worker-2-43     5       271265  0.001843
1697374878.078505       900.000057      worker-1-1      15      88004   0.017045
1697374877.834727       900.000055      worker-1-12     1       65145   0.001535
1697374878.095059       900.000194      worker-1-40     0       111174  0.0
1697374878.177380       900.000055      worker-2-28     0       57866   0.0
1697374877.969132       900.000055      worker-1-39     0       36952   0.0
1697374877.601313       900.000053      worker-1-32     4       81201   0.004926
1697374877.923157       900.000126      worker-1-25     5       180657  0.002768
1697374877.740190       900.000056      worker-1-13     0       87047   0.0
1697374878.278332       900.000306      worker-2-38     3       24110   0.012443
1697374877.990455       900.000127      worker-1-41     0       16867   0.0
1697374878.355018       900.000055      worker-2-49     0       45175   0.0
1697374878.215765       900.000055      worker-1-58     0       46994   0.0
1697374878.452757       900.000055      worker-1-18     1       450702  0.000222
1697374878.487202       900.000216      worker-2-3      0       10064   0.0
1697374878.233573       900.000057      worker-1-31     0       26457   0.0
1697374878.015261       900.000056      worker-1-8      0       90117   0.0
1697374878.332967       900.000457      worker-1-5      2       24861   0.008045
1697374878.428053       900.000055      worker-2-25     7       210745  0.003322
1697374878.134070       900.000083      worker-2-18     5       32533   0.015369
1697374878.433071       900.000055      worker-1-17     1       25276   0.003956
1697374878.181051       900.000251      worker-2-41     3       94084   0.003189
1697374878.369854       900.000055      worker-1-27     1       70302   0.001422
1697374878.014689       900.000056      worker-1-51     7       295801  0.002366
1697374878.283399       900.000213      worker-1-19     0       5897    0.0
1697374877.841475       900.000054      worker-2-17     3       271166  0.001106
1697374878.307842       900.000058      worker-1-59     8       67550   0.011843
1697374878.106441       900.000057      worker-1-23     1       182082  0.000549
1697374878.640479       900.000056      worker-2-7      1       10455   0.009565
1697374878.550051       900.000055      worker-2-51     2       68625   0.002914
1697374878.122468       900.000250      worker-1-55     3       51650   0.005808
1697374878.616983       900.000056      worker-2-14     0       151978  0.0
1697374878.342385       900.000640      worker-2-30     2       21778   0.009184
1697374877.853966       900.000054      worker-1-16     3       267073  0.001123
1697374878.424653       900.000055      worker-1-21     0       489109  0.0
```

- Rightmost column shows each worker's recent % of avg lost traffic, calculated using TCP sequence numbers
- Configured as only 2 workers, but each separate thread acts like its own process

Data sources

Applying the Model: Kafka

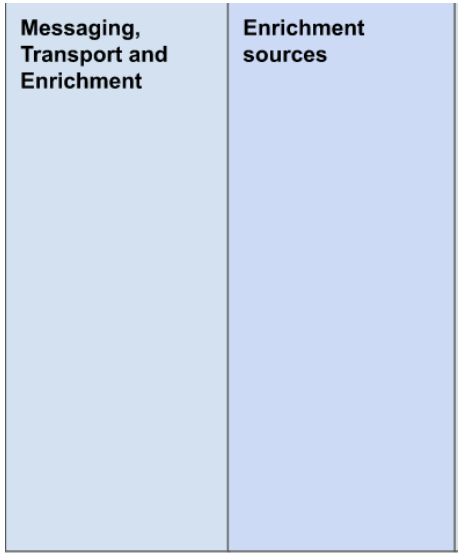
- Kafka collects Zeek logs and can be used to enrich data
- Using Kafka 3 (no zookeeper required) via Bitnami Docker image: https://github.com/bitnami/containers/tree/main/bitnami/kafka
- Using zeek-kafka plugin (https://github.com/SeisoLLC/zeek-kafka), a fork of the Apache Metron Bro plugin.
- Zeek acts as a producer for Kafka
- Can publish to distinct topics for each type of zeek log
- Offers a range of enrichment possibilities such as custom scripts
- Configuration still in development
- Will run on head node



Science and Technology Facilities Council

Applying the Model: Logstash

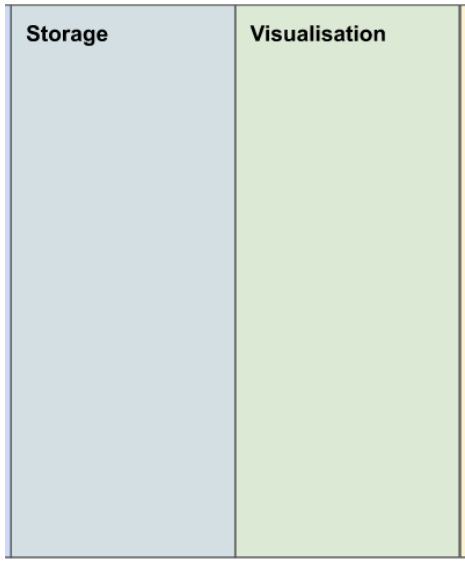
- Since no plugin to write data to OpenSearch directly from Kafka, Logstash is our method to bridge the gap.
- Using Docker to deploy
- Using the Kafka input plugin and OpenSearch output plugin
- Configuration still in development.
- Will run on head node



Science and Technology Facilities Council

Applying the Model: Opensearch

- Runs on search nodes within the SOC rack
- OpenSearch 2.4
- Docker compose deployment
- In production with no data currently
- 2 OpenSearch containers + 1 Dashboards container per host (4 hosts total)
- Role-based access control provided by IRIS IAM (an instance of Indigo IAM)
- How to expose OpenSearch Dashboards from within a private SOC network?
    - Our solution is to amend firewall rules so that the dashboards port on each search node can communicate with only a departmental load balancer. This is made available over the RAL VPN to give staff access.



Science and Technology Facilities Council

Applying the Model: Elastalert and Scripts

- Work in progress
- Scripts for
  - long-lived network connections
  - many repeated connections from same IP
  - long dns queries
  - larger than normal data-transfers
  - Uncommon user-agent strings in http log
  - IPs outside of known subnets
  - Expired x509 certificates

Alerting + Incident Response

Positive: Zeek

- Collaborating with other security engineers from NIHEF/CERN
  - ..and slowly starting to bring more to the table
- Learning opportunity in network systems, mostly
- Satisfying to get it up and running at such a high standard, without yet breaking or failing (at least for OPN links)
- Optimistic start

Positive: Hackathon

- WLCG SOC Hackathon
  - JISC made a new MISP deployment
  - Zeek RPM’s for Rocky9/RHEL9/SL9
- Good to have a regular WG for SOC deployments
  - (next one maybe at CERN?)

Challenge: SSO

- Integrating services with Single-sign-on providers
  - Had to switch from Indigo IAM to Keycloak to get MISP login working, though was ok for OpenSearch.

Positive Challenge: Configuration Management

- SecOps Archetype, minimum trust config
- Not built on configuration management assumptions
  - No SSH as root
- Challenging assumptions breaks unexpected things
  - Assumption in account creation that the admin is root for ownership of ssh keys file
    - Config now based on ownership by each user
  - Sudo config defined groups differently in different areas

Next Steps

- Deploy Misp on VMWare host
- Deploy logstash and kafka to head nodes
- Zeek ingesting Janet links
- Discussions on how to integrate into our existing processes
- Elastalert/zeek scripts
- FIR (Fast Incident Response)

UKRI Science and Technology Facilities Council

Thank you

Science and Technology Facilities Council @STFC_matters Science and Technology Facilities Council