PROCEEDINGS OF SCIENCE

Building a minimum viable Security Operations Centre for the modern grid environment

David Crooks*
Science and Technology Facilities Council, UK Research and Innovation
E-mail: david.crooks@stfc.ac.uk

Liviu Vâlsan
CERN
E-mail: liviu.valsan@cern.ch

The modern security landscape affecting grid and cloud sites is constantly evolving, with threats being seen from a range of avenues, including social engineering as well as more direct approaches. It is vital to build up operational security capabilities across the Worldwide LHC Computing Grid (WLCG) in order to improve the defence of the community as a whole. As reported at ISGC 2017 [1] and 2018 [2], the WLCG Security Operations Centres (SOC) Working Group (WG) [3] has been working with sites across the WLCG to develop a model for a Security Operations Centre reference design. We present the current status of a minimum viable SOC design applicable to a range of different WLCG sites, centred around a few key components.

The design uses the Zeek [4] Network Intrusion Detection System for monitoring what is happening at the network level in strategic locations: for example at border between the local cluster and external networks, the border between different local network domains or at core infrastructure nodes. The MISP [5] Open Source Threat Intelligence Platform is used to share information regarding relevant security events and the associated Indicators of Compromise (IoCs). By feeding IoCs from MISP into Zeek we have a platform that allows the community to share threat intelligence that is immediately actionable across the entire grid.

The logs produced by Zeek are processed using the Elasticsearch, Logstash, Kibana (Elastic) stack for real time indexing and visualisation. This provides sites with a powerful tool for incident response and network forensics. The alerts raised by Zeek are further aggregated, correlated and enriched by an advanced notification processing engine. This ensures that most false positives are automatically whitelisted while at the same time reducing the total number of raised alerts that need to be managed by the computer security team of each site. By enriching these alerts and adding context of what happened around the moment the malicious activity was detected, the time needed to handle these alerts is greatly reduced.

We present possible deployment strategies for all these components in a grid context as well as the integration between them. We also report on the current status of work on integrating other sources of data, in particular using netflow / sflow, into this model.

Lastly we discuss how making use of these SOC capabilities distributed across the participating sites can lead to increasing the operational security across the entire grid.

International Symposium on Grids & Clouds 2019, ISGC2019
31st March - 5th April, 2019
Academia Sinica, Taipei, Taiwan

*Speaker.

PoS(ISGC2019)010

1. Introduction

In this paper we present the initial model as developed by the WLCG SOC Working Group for a minimally viable Security Operations Centre, applicable to a range of different WLCG sites. The goal of this model is to synchronise threat intelligence with a remote source, ingest security monitoring data, store it in a searchable repository and visualise it, enrich this data with threat intelligence, and alert based on any consequent correlations.

In the process of building up this initial model, the Working Group took inspiration from the CERN SOC, a large scale system used in production at CERN. The block diagram of the CERN SOC, as of February 2019, can be seen in figure 1.

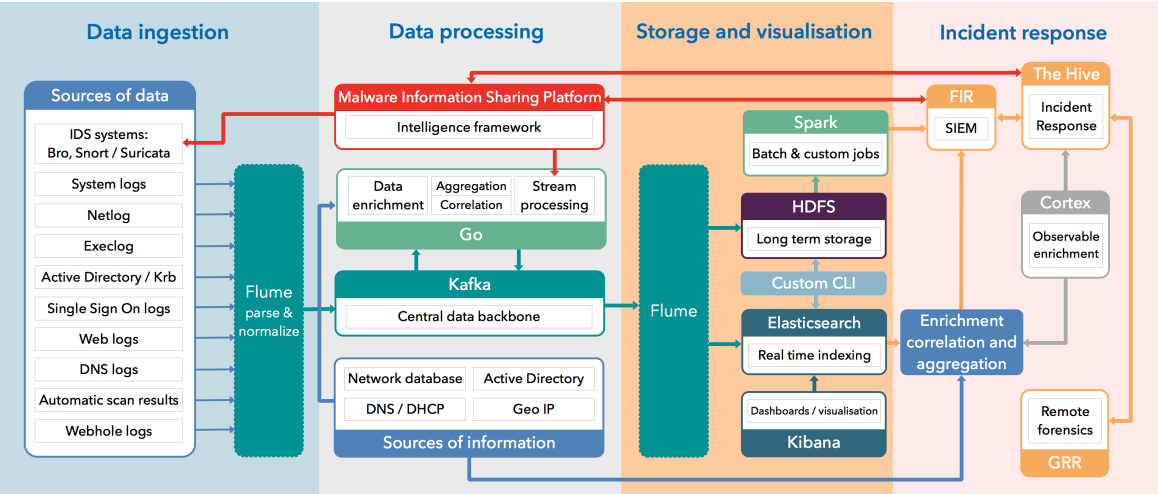

Figure 1: CERN SOC stages diagram (as of February 2019)

This paper focuses on the technical aspects of building such a minimally viable Security Operations Centre, more specifically the different open source software components and pipelines required in the formation of the initial model discussed below. Other important aspects of this work including policies, workforce aspects and other issues will be discussed in a future paper.

2. Overview and design goals

A high level diagram of the WLCG SOC WG initial model can be seen in figure 2. The WLCG SOC reference design is based on 4 different stages:

- Data sources and threat intelligence
- Data pipelines
- Storage and visualisation
- Alerting

Besides the essential building blocks (discussed in sections 3.1.3, 3.3.1 and 3.3.2), most of the stages include a number of optional building blocks. The Working Group has taken this approach

in order to provide the required flexibility so that sites of varying sizes could easily adapt the model to their specific set of needs and local configuration. In order to have a viable SOC deployment, implementation of at least one of the optional components in each of the stages is required.

In order to demonstrate the integration between the different SOC components of the initial model, the Working Group has produced a demonstrator, which will be discussed in section 5.

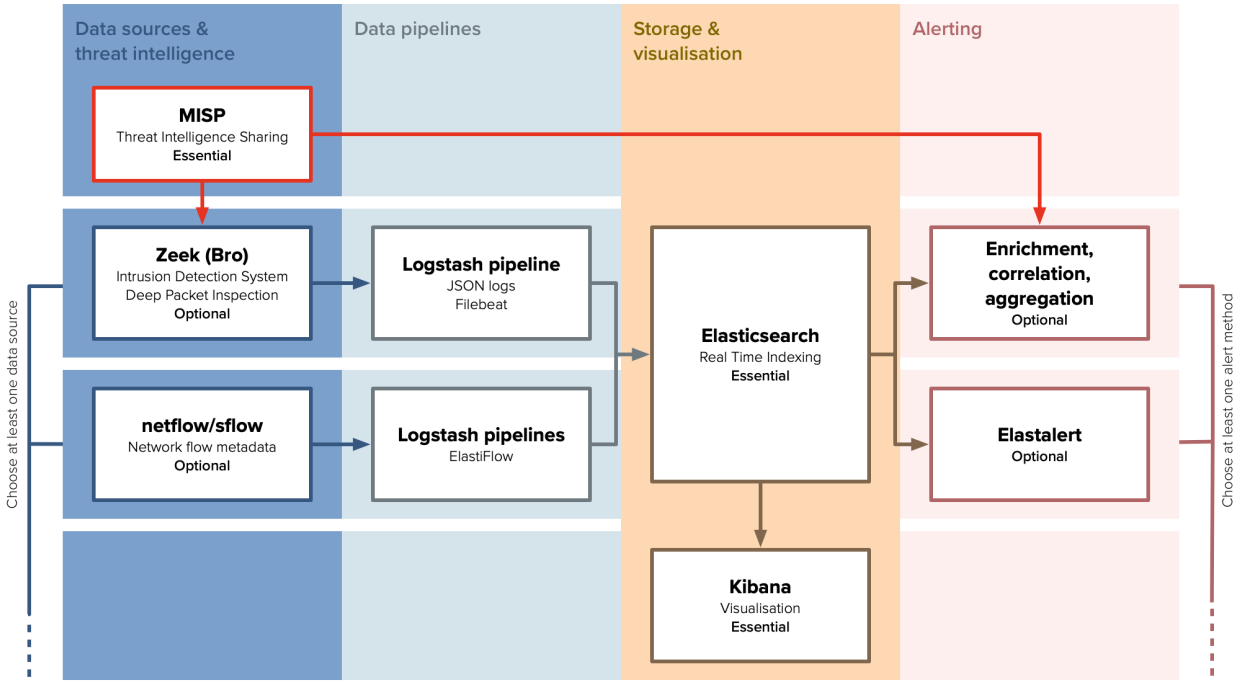

Figure 2: WLCG SOC Working Group Initial Model

3. Stages of the initial model

3.1 Data sources and threat intelligence

Given the fact that all security threats aimed at grid environments contain a network component for remote control and command capabilities, we see network based monitoring as a core component of the SOC initial model. In addition, modern virtualised grid environments can reduce visibility into running processes, which also emphasises the importance of network monitoring.

3.1.1 Zeek

Zeek [4] is a powerful network analysis framework, which provides and extends the functionality typically found in a traditional Network Intrusion Detection System. Used by major companies as well as in the research and education sector, Zeek has grown from a research project over two decades ago to provide a general network traffic analysis platform while emphasising network security monitoring.

Zeek has been selected as a key component for the initial SOC model because of the following properties and features that it provides:

- *Network security monitoring*: Zeek comprehensively logs network traffic and provides detailed network level traceability information.
- *Adaptability*: Zeek's built-in domain-specific scripting language enables the deployment of site-specific monitoring policies. This gives the flexibility for different sites to customise Zeek to their particular needs and local configurations.
- *Efficiency*: Zeek has been developed for monitoring high-throughput networks and is used operationally by a variety of organisations, including CERN.
- *Dynamic protocol analysis*: Zeek comes with analysers for many protocols out of the box, enabling inspection of high-level application layer protocols. Detection is dynamic and does not rely on the port numbers used.
- *Open interfaces*: Zeek interfaces with other applications for real-time information exchange. This makes it easy to ingest threat intelligence data from MISP and to have Zeek logs ingested by a data processing pipeline.
- *Alerting and threat intelligence*: Zeek comes with its own built-in threat intelligence framework and correlation engine, able to handled tens of thousands of indicators. A flexible alerting framework is responsible for notifying the security team whenever Indicators of Compromise (IoCs) are sighted in the network traffic.
- *Open source*: Zeek is licensed under a BSD license, allowing for free, unrestricted use.

3.1.2 NetFlow and sFlow

While Zeek provides a very powerful network analysis framework for efficiently extracting detailed network traffic metadata up to application level protocols, many grid sites already have the ability to collect NetFlow [6] and sFlow [7] data.

In an effort to devise an initial model as widely applicable as possible, leveraging as much as possible capabilities already present at the sites, the Working Group has opted to include NetFlow and sFlow as optional sources of networking information.

Given the overlap between the different mechanisms for collecting network traffic information, it is not necessary to collect NetFlow / sFlow data in the case where Zeek is used. Zeek produces logs which provide much more detailed network traceability information than NetFlow or sFlow. Nonetheless, NetFlow and sFlow can prove to be very useful sources of data for complementing Zeek logs. Given the fact that Zeek performs Deep Packet Inspection of the networking traffic, the required resources are relatively high. As such, it may not be feasible for sites to analyse their entire network traffic using Zeek. Zeek is commonly used for monitoring the network traffic at key locations, e.g. at the border between the local site network and the Internet or the network traffic going to the site's most exposed servers. On the other hand, NetFlow and sFlow can be used to collect network traffic logging information across a considerably larger area of the network, since the data is produced directly at the level of the networking devices, without the need for dedicated

compute resources. Given the limited scope of the network logging information provided by NetFlow and sFlow, the data storage needs are significantly reduced compared to the very detailed network traffic metadata logging data produced by Zeek.

NetFlow and sFlow also provide the option of sampling network traffic, further reducing the data storage needs. It should be noted, though, that sampled network traffic is not suitable for operational security purposes by itself. It should be noted that since in this scenario not all traffic is logged, such a data source cannot be relied on solely for network forensics purposes, as by itself it would paint a partial view of the security event.

3.1.3 MISP

The MISP [5] threat sharing platform is a free and open source software platform for information sharing of threat intelligence, primarily including Indicators of Compromise (IoCs).

One major reason for selecting MISP as a key component for the initial model is that it has become the de facto platform for sharing threat intelligence at a global scale, being actively used by a wide range of organisations. In addition, it also comes with the following highly desirable features:

- *Efficient and comprehensive IoC database*: MISP can be used to store technical and non-technical information about malware samples, security events and attackers.
- *Automatic correlation*: MISP automatically finds relationships between attributes and indicators from malware, attack campaigns or analysis.
- *Built-in sharing functionality*: MISP can automatically synchronise events and attributes among different MISP instances, based on several distribution models. Advanced filtering functionalities can be used to define different sharing policies.
- *Intuitive user-interface*: Users have access to a graphical interface that allows to create, update and collaborate on events and attributes as well as to seamlessly navigate between events and their correlations.
- *Flexible import / export*: Exporting to IDS (Zeek, Suricata and Snort are supported by default), as well as export / import to / from OpenIOC, plain text, CSV, MISP XML or JSON output to integrate with other systems (network IDS, host IDS or custom tools). A flexible Python API is also available for integration with 3rd party tools.
- *Collaboration*: MISP allows users to propose changes or updates to MISP events created by another party.
- *Adjustable taxonomies*: Functionality to classify and tag events, using custom classification schemes or existing taxonomies. MISP comes with a default set of well-known taxonomies and classification schemes to support standard classification as used by many organisations.
- *Sighting support*: Sightings can be contributed via the MISP graphical user-interface or via the API; this allows the exchange of observations (both true positives and false positives) of shared indicators and attributes between multiple sites.

3.2 Data pipelines

For each data source defined in the first stage, a pipeline is required to format the data appropriately and ingest it. In the initial model, all pipelines are based around the use of the Elastic stack [8].

The Elastic stack is an open source platform for the reliable ingestion of data from different sources, in a variety of different formats, allowing to search, analyse and visualise data in real time. In this section we will use the two log ingestion tools that are part of the Elastic stack: Logstash [9] and Filebeat [10]. The data storage, search and analytics engine Elasticsearch is discussed in depth in section 3.3.1, while our use of the web based visualisation tool Kibana is discussed in section 3.3.2.

As noted, the tools used for log ingestion in the initial model are Logstash and Filebeat, primarily as these form an integral part of the Elastic stack and are likely to be familiar to admins used to ingesting data into Elasticsearch. Other options are available - the CERN SOC uses Apache Flume [11], for example - however, it was felt that Logstash made the most sense to use for the reference design.

3.2.1 Zeek

At the time of writing, the following steps form the data pipeline for Zeek:

1. Initial log format: enable JSON logs in Zeek
2. Transport log data to Logstash: deploy Filebeat on the Zeek host
3. Data conditioning, enrichment and transport to Elasticsearch: deploy Logstash pipeline on a separate host

The Logstash pipeline allows for enrichment of the data prior to ingestion into Elasticsearch; an example of this would be the use of GeoIP data to give geographic locations to IP addresses.

3.2.2 Elastiflow

The suggested data pipeline for NetFlow and sFlow data is Elastiflow [12], a set of Logstash pipelines and Kibana dashboards; this software provides options for a range of Elasticsearch versions. The GitHub project gives installation instructions; once deployed on a host with the relevant Elasticsearch details added, the NetFlow or sFlow generator in use at a given site can be directed at the Logstash host.

3.2.3 Schema considerations

The most benefit from the ingested data will come when correlations can be made between common terms. A piece of ongoing work is the exploration of the schemas defined in the pipeline stage to best support this type of correlation.

3.3 Storage and visualisation

The storage and visualisation stage is the central piece of the proposed SOC model. In order to ensure reliable log ingest, storage and analytics, this is the only stage for which components are fixed, aside from threat intelligence which uses MISP.

3.3.1 Elasticsearch

At the core of the Elastic stack lies Elasticsearch, a distributed, RESTful search and analytics engine suitable for a wide range of use cases. Elasticsearch is used to centrally store data, while indexing it in real time and providing quick access to the data.

The Working Group has taken the decision of using the Elastic Stack as a core storage and analysis platform for a number of reasons, including its ubiquitous nature, ease of deployment and existing experience at many WLCG sites, as well as its ease of use and suitability for storing and analysing security logs.

The Working Group documentation website [13] contains detailed Elasticsearch deployment instructions, including a description of the Elasticsearch setup used for the CERN SOC, Elasticsearch deployment best practices as well as troubleshooting guidelines.

3.3.2 Kibana

Kibana is the data visualisation platform that comes bundled with Elasticsearch, as part of the Elastic stack. Given its tight integration with Elasticsearch and its flexibility, it is considered to be a very good fit for providing quick access to the data stored inside a SOC.

Kibana is suitable for different access patterns. Firstly it allows to discover data by querying different index patterns, with arbitrary filters and custom date intervals. Kibana can also be used for building visualisations and for bundling multiple visualisations together inside dashboards.

3.3.3 Elastic Stack Security

Given the fact that WLCG sites typically use the Elastic stack for a variety of different use cases, it is advisable to fully isolate accesses to the different categories of logs. For example, only the local Computer Security Team should be given access to the security logs. One option for achieving isolation of Kibana data accesses, visualisations and dashboards is by using the Kibana own home plugin [14]. This plugin adds multi-tenancy capabilities to Kibana, enabling user and group personal Kibana indices so that user or group specific objects are stored to separate locations.

In addition to this compartmentalisation, deployments should ensure proper access restrictions and overall cluster security. A number of different options for this are available. While the Elastic Stack does have enterprise-grade security features available (formerly packaged as X-Pack), these are currently paid features. An alternative, which provides both a paid and a free version and is in use at CERN, is ReadonlyREST [15].

3.4 Alerting

Alerting is a key functionality of a SOC. Detailed security logging information coupled with up to date IoCs can be efficiently leveraged with the implementation of timely alerts whenever IoCs are being detected.

3.4.1 Advanced correlation, enrichment and aggregation of SOC alerts

As a key component of their SOC, CERN has developed a system for advanced correlation, enrichment and aggregation of alerts. The processing of alerts is triggered at regular intervals, batching similar alerts, automatically removing false positives and enriching true positive alerts.

Different alerts are considered to be related if the respective IoCs that triggered the alerts are part of the same MISP event or of related (correlated) MISP events.

Great emphasis has been placed on dealing with false positives. The additional context from MISP is used for filtering out false positives. For example, alerts raised following a connection to a potentially malicious IP are being filtered out, if the MISP event contains a domain name linked to that IP (i.e. via a composite attribute or inside a MISP object) and the detected connection was made to a different domain that the one from the MISP event.

After the removal of these false positives, the true positive alerts that remain are enriched with additional sources of information to allow easy triage of alerts. These additional sources of information include, but are not limited to: MISP, CERN's internal networking database, reverse DNS, WHOIS information, GeoIP data and external threat intelligence services.

The alerts also contain additional information providing context around the time of the alert and the involved computing resources. For example, besides the basic network connection metadata, the generated alerts also include application specific logs (e.g. HTTP connections, certificates used for SSL / TLS connections, DNS queries made, etc).

While CERN's system for advanced correlation, enrichment and aggregation of SOC alerts is highly integrated with the different sources of enrichment data used at CERN, work has been done to come up with a streamlined, simplified version of it that can be deployed at a wide of different grid sites. In this regard, a prototype has been fully tested at the ATLAS Great Lakes Tier-2 (AGLT2) center at the University of Michigan and Michigan State University. While the streamlined version is lacking the advanced enrichment of security logs, it is still useful to use for automatically filtering out false positives as well as for correlating and aggregating alerts with the aim of reducing the total number of SOC generated alarms. The alerting module can be easily extended with site specific integrations for enriching SOC alerts with additional local sources of data (i.e. networking database mapping local IP addresses to device names).

3.4.2 ElastAlert

A second means of generating alerts has been through the use of the ElastAlert tool [16]. This tool allows alerts to be sent via a number of channels, based on targeted queries of the Elasticsearch instance. Alerts can be triggered on the basis of a number of queries, including event spikes.

4. Deployment guidelines

The deployment of this model, notwithstanding its design as a minimally viable set of components, can be complex. The Working Group documentation website [13] details the recommendations and deployment instructions for each component; this documentation set will grow as the experience of Working Group participants is added. A particular piece of ongoing work is the creation of a template project plan to help sites plan their own SOC projects - this is especially in recognition of the need for close inspection of the network topology of a given site, which will invariably differ between sites.

5. PocketSOC

PocketSOC [17] is a containerised, bench-top implementation of the model described above deployed using Docker [18]. The original use case envisaged was as means to demonstrate the workflow of the model in a contained environment, both in a workshop setting and as part of familiarising new users. In a context where the concepts of threat intelligence and the aggregation of security monitoring may not be familiar, it was felt that a method of demonstrating the intended workflow in a simple way would be beneficial. Particularly in the initial stages of deployment, the amount of data observed using a network analysis framework such as Zeek can be daunting.

A key design goal of PocketSOC was to allow the processes being used to be demonstrated using a very small amount of data specifically generated for this purpose; in this way it is hoped that the workflow can be understood separately to the normal traffic observed at a given site. An additional use is in the testing of new components; by using additional containers, new processes can be tested in isolation without risk to production services, without the need for a dedicated development cluster. A final use case, developed more recently, has been in the replaying of genuine traffic to be ingested into the model; this will be discussed in section 5.3.

The deployment of services within PocketSOC uses the documentation developed by the Working Group wherever possible. A high level overview of PocketSOC can be seen in figure 3.

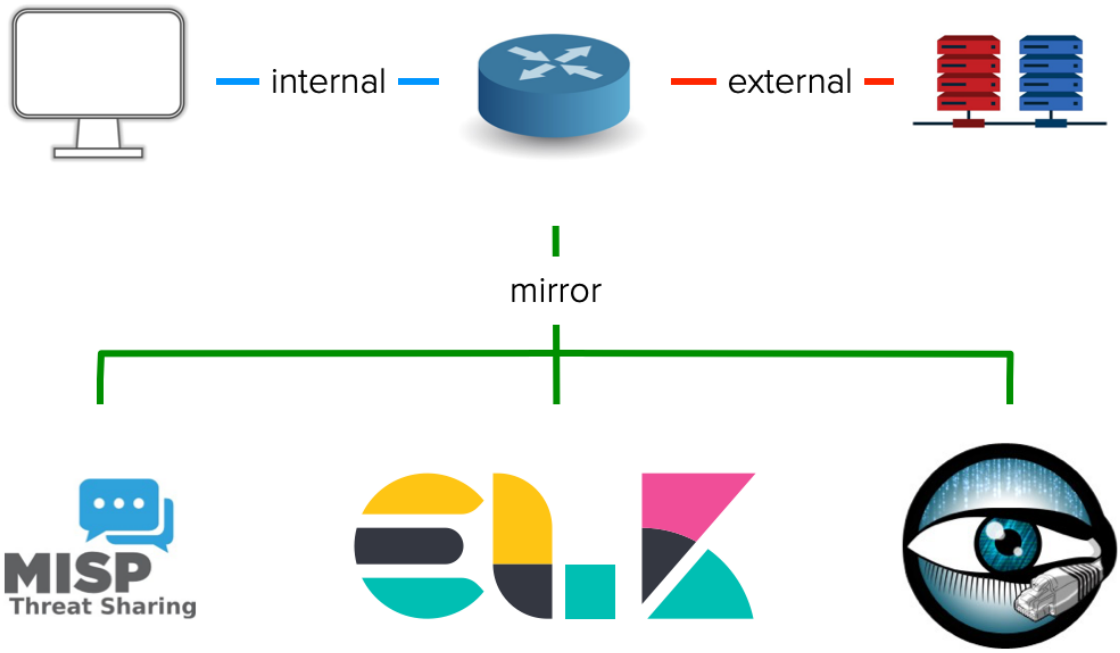

Figure 3: PocketSOC diagram

5.1 Orchestration and components

The orchestration of PocketSOC is performed using `docker-compose` [19]. The individual components currently in use are:

- Zeek: standalone Zeek instance, deployed as a single host
- MISP: dedicated, private MISP instance, not configured to sync with any other MISP instance by default
- Elastic stack: Elastisearch, Logstash and Kibana deployed as separate containers. In particular, a separate Logstash container is used for each pipeline.
- client: plays the role of a host within a site network that would access external resources
- HTTP data source: provides HTTP source of traffic
- router: provides the routing between the three docker networks (see section 5.2)

5.2 Networking

Three private networks are defined, which are connected using the `router` component.

- *Internal*: The `internal` network is intended to mimic the network region inside a site; this is where a client may be located that is accessing external resources.
- *External*: The `external` network is intended to mimic the Internet as a whole; the various demonstration data sources are located on this network
- *Mirror*: The `mirror` network is the location for the security components of PocketSOC - it was so named for the traffic being mirrored from the internal and external networks through the PocketSOC router.

5.3 Replaying traffic

The final, and most recent, use for PocketSOC is the demonstration of the use of threat intelligence with genuine traffic and MISP events. This has been carried out using `tcpreplay` [20] with `.pcap` packet capture files. The key steps in carrying out this demonstration are:

1. Capture appropriate traffic using, for example, `tcpdump` [21] or other tools.
2. Copy the resulting `.pcap` file to the Zeek container
3. Export the desired MISP event from the live instance as a MISP JSON event, and import this into the PocketSOC test instance
4. Replay the packet capture, directed at the PocketSOC Zeek instance

The traffic may then be analysed as it would be in reality, but in an isolated network environment - this includes both the appearance of the data in Kibana dashboards and also the alerts raised by the Zeek intelligence framework as a result of the import of the now local MISP event.

At time of writing, this has been used to demonstrate Zeek intelligence alerts being raised as a result of traffic from malware analysis coupled to a real MISP event. It is hoped that, with appropriate protections in place for sensitive data, this may provide a way to demonstrate the utility of this workflow in a manner that reflects realistic results.

6. Future work and conclusions

Following the establishment of this initial design, the next goal would be to have this model deployed at participating Working Group sites, including a demonstration of the entire data workflow. Threat intelligence is then the next major area of focus for the Working Group, including both the technical aspects of sharing intelligence and the formation of trust groups necessary to achieve this. Sharing difficulties are often not technical issues but a matter of social interactions, including the building of trust.

We have described in this paper the initial model to deploy a Security Operations Centre at a typical WLCG site, including a description of the key components and their place in the model. This is based on a proven design from the CERN Computer Security Team and makes simplifications based on common components such as the Elastic stack. With this work in place the Working Group can focus on the experience of deploying this model and how it can be used operationally, with the key inclusion of threat intelligence.

7. Acknowledgments

We gratefully acknowledge the work of the members of the WLCG SOC Working Group in the preparation of this paper.

References

[1] D. Crooks and L. Vâlsan (2017) WLCG Security Operations Centres Working Group. International Symposium on Grids & Clouds 2017 (ISGC 2017), BHSS, Academia Sinica, Taipei, 5-10 March 2017

[2] D. Crooks and L. Vâlsan (2018) Harnessing the Power of Threat Intelligence in Grids and Clouds: WLCG SOC Working Group. International Symposium on Grids & Clouds 2018 (ISGC 2018), BHSS, Academia Sinica, Taipei, 16-23 March 2018

[3] WLCG Security Operations Centres Working Group, https://wlcg-soc-wg.web.cern.ch

[4] Zeek, https://www.zeek.org

[5] MISP, https://www.misp-project.org

[6] NetFlow, https://netflow.us

[7] sFlow, https://sflow.org

[8] Elastic stack, https://www.elastic.co

[9] Logstash, https://www.elastic.co/products/logstash

[10] Filebeat, https://www.elastic.co/products/beats/filebeat

[11] Apache Flume, https://flume.apache.org

[12] Elastiflow, https://github.com/robcowart/elastiflow

[13] WLCG Security Operations Centres Working Group documentation website, https://wlcg-soc-wg-doc.web.cern.ch/wlcg-soc-wg-doc

[14] Kibana plugin: Own Home, https://github.com/wtakase/kibana-own-home

[15] ReadonlyREST, https://readonlyrest.com

[16] ElastAlert, https://github.com/Yelp/elastalert

[17] PocketSOC, https://gitlab.cern.ch/wlcg-soc-wg/pocketsoc

[18] Docker, https://www.docker.com

[19] Docker Compose, https://docs.docker.com/compose

[20] tcpreplay, https://tcpreplay.appneta.com

[21] tcpdump, https://www.tcpdump.org