To SOC or not to SOC ?

For environments that are secure by design, a 'full-fat SOC' is not always required.

David S

So, you are implementing a large digital project, and have followed the GOV.UK Service Manual through each phase of the project.

You're now ready to go live, but the accreditor for the project needs to know if the service has 'protective monitoring'. Which means the team needs to spend time and effort procuring (or setting up) a Security Operations Centre (SOC).

Or does it?

Building a SOC is a task that requires a lengthy investment of both time and money. This blog considers whether it's possible for organisations to design systems in such a way that a 'full-fat' SOC is not required.

How does a SOC work ?

A SOC will often be run by a separate team to the operational staff. This helps provide a separation of duties between those that manage the digital services, and those monitoring logs for security.

The SOC itself usually has specialised tooling, the most common being the SIEM (security information and event management) tool. It takes logs and data sources as input, performs some correlation and rule checking, and then outputs alerts for triaging. Licencing for SOC tooling can be expensive, typically the more enterprise-focused solutions tend to cost the most (and often require annual renewal).

A SOC team will typically comprise:

- security analysts triaging events and alerts

- security engineers managing the tooling (as well as onboarding projects to the SOC)
- some form of management and admin functions

and in some instances

- a threat intelligence function

Experiencing a security incident where there are **no logs** to help work out what happened – and having to report this to seniors – is a bad place to be. In some situations, logs may be required for legal and forensic purposes, so proving a chain of custody around their handling and integrity is important.

Depending on the SIEM tool, some offer cryptographic integrity and validation of logs, to determine whether logs have been tampered with. A SOC is often a great strategy for ensuring logs are:

- collected safely
- accessible by authorised individuals
- based on the organisations policies
- not simply collected for a rainy day (or forgotten)

However, setting up your own SOC can be expensive and time consuming, so one option is to outsource the SOC. However, external SOC analysts can lack the knowledge of (for example) a project's architecture, and may not be able to respond to alerts as promptly as an internal SOC.

GPG13: the elephant in the room

CESG, the NCSC's predecessor, published a 'good practice guide' (GPG) on protective monitoring. Known as GPG13, people tasked with dealing with risk often used GPG13 as a requirement prior to a service going live. In some scenarios, this turned protective monitoring into a checkbox exercise, with the marketing material for SOC tooling describing their product as 'GPG13 compliant'.

The problem was exacerbated as applications and services started to diversify in their architecture and technology stacks, because there was no incentive for service owners to identify and monitor risks that were not documented as part of GPG13.

GPG13 was deprecated before the NCSC was formed. However, we still get customers quoting GPG13 in their assurance process, and occasionally get asked when replacement guidance is going to be published. To differentiate between 'the GPG13 approach to protective monitoring' and the NCSC's current approach, I will be using the term 'security monitoring' to describe the approach considered for services built using cloud-native services.

How does the cloud change things ?

The UK government introduced a 'Cloud First' policy in 2013, with the goal of getting government departments to consider cloud first solutions before considering traditional on premises deployments.

So how does this shift to cloud adjust our security requirements? For deployments that are a lift-and-shift from on-premise infrastructure to managed IaaS, this does not reduce the requirement for a SOC, as there is no inherent shared responsibility model to reduce operational responsibility. By following and implementing the NCSC’s cloud security guidance, organisations can increasingly turn to the monitoring tooling available from the cloud provider (rather than relying on SIEM, SOCs, or specialised security personnel).

A key advantage of a SOC is being able to identify and alert on risks that are specific to the environment. This requirement doesn't go away with security monitoring, and it's still important to identify and monitor for any custom alerting.

What, no SOC ?

So what sort of approaches are already being used as an alternative to a full-fat SOC? Here are some that government projects are currently adopting:

- **Completely cloud-native architecture**
  Deployed services that use only 'cloud-native' services, leveraging the advantage of tying in tight identity management around service usage, as well as not having the operational overhead of deploying, patching and troubleshooting services that would have traditionally been deployed on VMs.
- **Zero touch production**
  Deployed services where engineers never have direct access to production services (other than by tightly monitored and audited break-glass solutions). This has resulted in less risk to the system and a reduction in security monitoring use cases.
- **Environment separation**
  The use of separate cloud accounts for functions that should remain segregated. For example, accounts that store security monitoring logs and services that should not be accessible (even during a break-glass event) are hosted in separate cloud accounts, with strict access controls and alerting in place.
- **Simplify log collection**
  Cloud-native services offer their own logs (as well as services to consolidate and analyse them). As security monitoring requirements are simplified, so is the logging of events. Logs are now in consistent formats, and can be collected and stored in the cloud. Some cloud providers have options that tackle the integrity of logs, for example storing checksums that can be used to validate log integrity and may be useful during an audit.
- **Replacing SIEM**
  Government departments that already have a central SOC find onboarding services a time-consuming task. By keeping their architectures simple, some have been able to replace the requirement for a SIEM by extending their cloud-native logging solution, and building their rules and alerting directly into the platform. Secure development practices mean the operations team maintains responsibility for security, so no security team is needed. Incident

Management is crucial and knowing what to do and their responsibility is important.

- **Break glass**
Sometimes direct production access is required. This could be due to an operational need such as investigating performance degradation, or could be a requirement as part of a security incident. In some projects, the operations team are responsible for investigating and running incidents, and this is enabled by well-documented process and procedures, good access and change control as well as tight auditing of the system. Access is time-limited, and all events are audited and monitored by other individuals in the team.
- **Validating logging**
All of this is to no avail if the logging stops working, and no one notices. Services can be injected with canary tokens and alert/error if the token is not observed within a time window.

Final thoughts

Where services are being built at OFFICIAL, 'cloud first' should be the focus, and this should include all the above areas highlighted to help tighten up and **simplify** the security monitoring requirements.

For AWS (for example), you have tightly controlled, cloud-native services such as GuardDuty, Security Hub, CloudTrail, CloudWatch. So rather than deploying a SIEM, you enable your operations teams to own the monitoring, working in environments that are simple and secure by design. By giving operations teams more ownership of the service, we will find security monitoring becomes templated for re-use, much like happens now for deploying infrastructures.

In evaluating whether a project needs a SOC, you should consider the functions you're relying on a SOC for and whether those are covered in other ways:

- **Do you need to have logs to investigate an incident should it happen ?**
Cloud-native services (if correctly configured) can do this, and you will need

to consider log retention and the roles that can access or delete logs.

- **Is the requirement to detect attacks as they happen ?** Cloud-native/serverless architectures will be far more limited in the nature of attacks possible, as components are typically single purpose with underlying infrastructure taken care of. By evaluating the sorts of attacks that remain possible in your architecture, you can set up specific alerts.
- **Is there a requirement to manage incidents ?** As with the above, the nature of alerts will be different in a serverless environment, and the operational team for the service will likely be more able to identify suspicious behaviour than would a general SOC analyst. The important step then becomes ensuring the operational team know how to escalate suspicions and have tested that process.

The NCSC believe that SOC and protective monitoring-based systems still have a place in the security toolbox. For some enterprise IT systems (such as endpoints) and traditional IaaS based architectures (and systems at higher classifications than OFFICIAL), it remains a requirement to provide reactive monitoring of the system. There are also benefits to centralised SOCs where government departments can identify broader attacks that are probing multiple services used by the organisation.

David S, Senior Security Architect

WRITTEN BY

David S
Senior Security Architect,
NCSC

PUBLISHED

12 July 2023

WRITTEN FOR

Public sector

Cyber security professionals

Large organisations

PART OF BLOG

NCSC publications