Overview of the SOC Technologies

Data Collection and Analysis

In order to perform a useful analysis, we need to monitor and acquire relevant data. This type of data can be: *event logs, network packets, network flows.* It may also be useful to include hash values of important files, like */etc/passwd*.

The basic questions to follow:

- Which elements should you monitor?
- What data should you collect and in what form?
- What level of logging should you enable on each element?
- What protocols should you use to collect data from the various elements?
- Do you need to store the data that you collected and for how long?
- Which data should you parse and analyze?
- How much system and network overhead does data collection introduce?
- How do you associate data collection requirements with capacity management?
- How do you evaluate and optimize your data-collection capability?

Many SOC fail because they tend to collect and store bad, unsynchronized information, therefore the events triggered will be false positives.

The SOC data management workflow will be similar to the diagram below. This has to be defined in the SOC design phase.

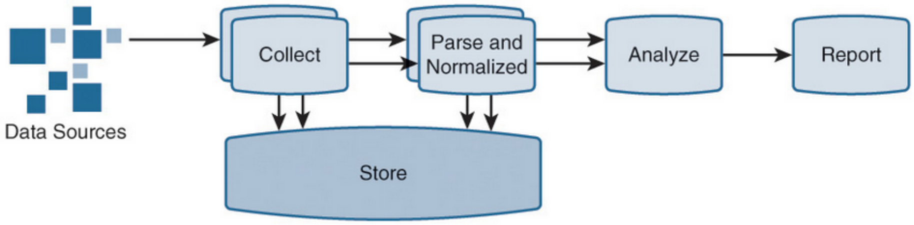


Figure 1: SOC Architecture

Data can be stored in a plain text file, in a relational database, or in a distributed file system (HDFS). Data can be analyzed in several ways, including a statistical-based anomaly detection way or machine learning techniques.

Once the data has been collected, we have to parse it so that we can interpret it and understand it at a future point in time.

Data Sources

Logging messages are considered the most useful type of data to acquire, but, depending on the granularity of the required logging structure, you might also need to store network packets. Every object stored has advantages, in terms of information, and disadvantages, in terms of storage and management.

Some examples of the virtual and physical devices that can provide logs are:

- Security elements such as firewalls, IDS, IPS, Anti virus, web proxies, and malware analysis tools.
- Network elements such as routers, switches, wireless access points and controllers.
- Operating Systems, such as UNIX, GNU/Linux, OS X, Windows.
- Virtualization platforms such as VmWare, VirtualBox, KVM, Hyper-V
- Applications such as web servers, DNS servers, e-mail gateways
- Databases
- Physical security elements, such as security cameras
- Systems used in process and control networks, such as SCADA and DCS

Data Collection

Depending on the data source, we can collect it by pushing the data to the collector or pulling the data by the collector. It is important to realize that we need a unified time between all of our devices, therefore, one of the most common ways in which we can ensure this, is by using a central timing server using the Network Time Protocol (NTP).

The Syslog Protocol

> Provides a message format that enables vendor-specific extensions to be provided in a structured way, in addition to conveying event notification messages from a syslog client (originator) to a syslog destination (relay or collector). The syslog protocol supports three roles:

- **Originator:** Generates syslog content to be carried in a message
- **Collector:** Gathers syslog messages
- **Relay:** Forwards messages, accepting messages from originators or other relays and sending them to collectors or other relays.

To configure a syslog system correctly, we have to use technologies like *rsylog* in UNIX or install syslog-like programs in Windows.

We also have to configure a set of parameters:

- *Logging destination:* The collector, relay IP addresses, or hostnames.
- *Protocol and Port:* Typically, UDP and port 514.
- *Logging severity level:* A value from 0 (Emergency) to 7 (Debug).
- *Logging facility:* Value between 0 and 23 that could be used to indicate the program or system that generated the message. The severity and logging facility values could be combined to calculate a priority value of an event, influencing the post-event actions to take.

Depending on the platform, skill set, vendor support, and acquisition and operation costs, you can use several log management solutions, from Splunk, to graylog2 and logstash.

Logging Recommendations

- In the context of security operation, log events that are of business, technical, or compliance value
- Configure clients and servers for NTP, and confirm that clocks are continually being synchronized
- Time-stamp log messages and include timezone in each message
- Categorize events by assigning logging facility values
- Limit the number of collectors for which a client is configured to the minimum required. Use syslog relays when you require the same message to be forwarded to multiple collectors. Syslog relays can be configured to replicate and forward the same syslog message to multiple destinations. This scenario is common when you have multiple monitoring platforms performing different tasks such as security, problem management, and system and network health monitoring.
- Baseline and monitor the CPU, memory, and network usage overhead introduced by the syslog service.
- Have a limited local logging facility, in file or memory, so that logs are not completely lost if the syslog collector is unavailable, such as in the case of network failure.
- On a regular basis, test that logging is functioning properly.
- Protect your syslog implementation based on evaluating the risk associated with syslog not providing confidentiality, integrity, or authenticity services.
- Ensure that log rotation and retention policies are properly set.
- Protect files where logs are stored: Restrict access to the system, assign proper files access permissions, and enable file encryption if needed.
- Read access to log files must be granted only to authorized users and processes.
- Write access to log files must be granted only to the syslog service.
- Standard system hardening procedures could be applied to operating systems hosting your logging server.

Logging Infrastructure

Other elements, such as type of data, expected storage, security requirements should be considered when designing a logging structure.

- Logging level
- System resources available to the syslog
- Usable network bandwidth available between the logging client and the logging server

Telemetry data: Network Flows

Every network connection attempt is transported by one or more physical or virtual devices. Capturing and transferring network packets is not required, desired, or even feasible. The alternative to capturing packets is to collect contextual information about network connections in the form of network flow. A flow, according to the IPFIX standard, consists of network packets that share the same arbitrary number of packet fields within a timeframe. Network flow can be setup in hardware routers and switches. It works in sampled and unsampled modes. (It takes a sample every X packets / it inspects every packet)

A flow-based security product could identify a user as being authorized to perform valid actions but still flag the unusual behavior as an indication of compromise.

Telemetry data: Packet Capture

When you need to capture packets, because of a strict regulatory measures, there are two techniques with Ethernet:

- **Port mirroring:** Uses network switches to mirror traffic seen on ports or VLANs to other local or remote ports.
- **Network taps:** Connecting out-of-band devices in the form of network taps to monitor and capture packets from point-to-point links. They are ideal for on-demand troubleshooting.

Parsing and Normalization

Copious amounts of data can't be analyzed by a network analyst, this is where parsing and normalization comes into play, in order to be able to perform a security analysis later.

Security Analysis

- *Security Analysis:* the process of researching data for the purpose of uncovering potential known and unknown threats

Event correlation is the most used form of data analysis.

- *Security Event Correlation:* the task of creating a context within which revealing relationships between disparate events received from various sources for the purposes of identifying and reporting on threats

Correlation rules are packaged in SIEM tools. The vendors provide out-of-the-box use cases, but they are most probably going to be general rules, unspecific to your organization.

Before you formalize a use case, answer the following questions:

- What methodology to use to come up with use-cases
- Logging messages to collect and from which device (Per use-case)

- Complexity of the task
- Complexity and impact relationship
- Will the rule increase false-positives?

Alternatives to Rule-Based Correlation

Anomaly-based correlation is another approach that can be combined with rule-based.

First gather data, to establish a baseline, after, the tool can analyze traffic to observe if the baseline is met or not. If it is not met, an alert will be generated.

Risk-based correlation: calculate a risk score for an event based on the content and context of an event

Data Enrichment

Adds additional context to the data received:

- Geo information, allowing to map IPs to geographical location
- WHOIS information, additional context on the IP
- Reputation information on domain names, IP addresses and e-mail senders, file hash values, and so on.
- Domain age information

Big Data Platforms for Security

As companies grow, so do their logs, and relational databases are becoming a bottleneck to manage logs. The solution is to use big data systems with a Hadoop approach, using HDFS as their core.

Vulnerability Management

- *Vulnerability management:* refers to the process of discovering, confirming, classifying, prioritizing, assigning, remediating, and tracking vulnerabilities

The most critical element of vulnerability management is being faster at protecting the vulnerable asset before the weakness is exploited

Vulnerability Announcements

Vulnerabilities are found in open and closed source every day. To identify any given vulnerability, we use the CVE (Common Vulnerabilities and Exposures) The security tool alerts you of a found vulnerability and supplies a CVE identifier. You can therefore know exactly what vulnerability the security tool refers to.

In addition to CVE, there is a vulnerability scoring system (CVSS) which specifies the vulnerabilities impact on confidentiality, integrity and availability.

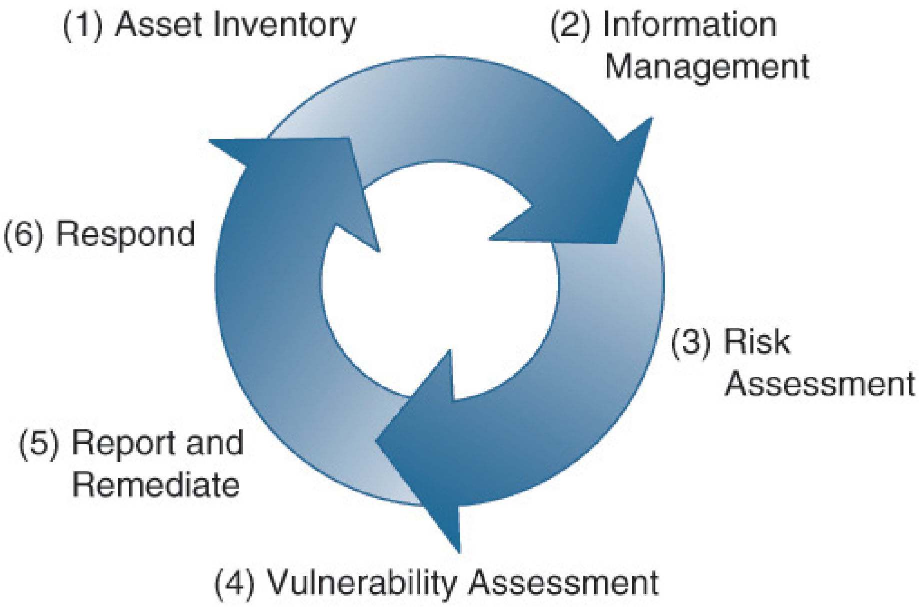
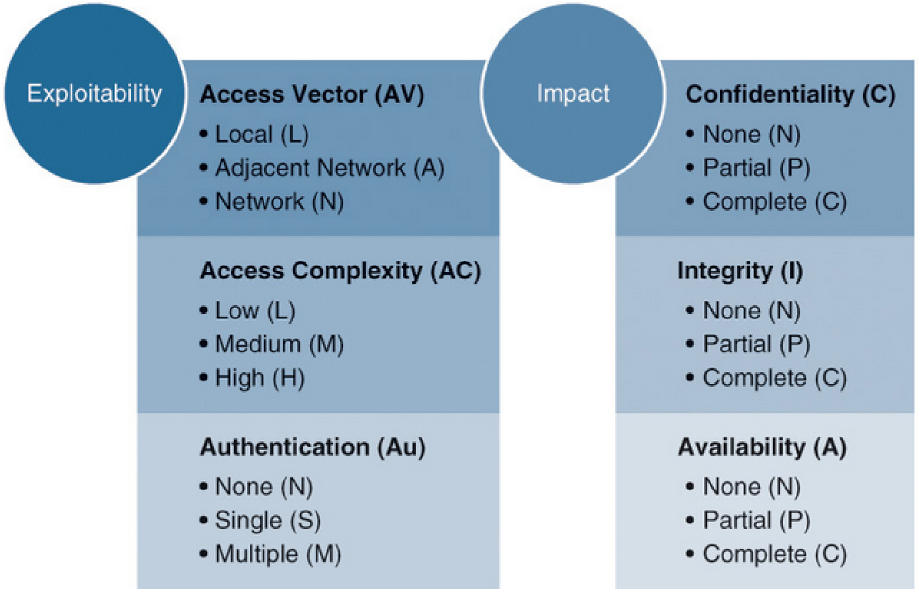


Figure 2: SANS Vulnerability Management Model



Exploitability

Access Vector (AV)
- Local (L)
- Adjacent Network (A)
- Network (N)

Access Complexity (AC)
- Low (L)
- Medium (M)
- High (H)

Authentication (Au)
- None (N)
- Single (S)
- Multiple (M)

Impact

Confidentiality (C)
- None (N)
- Partial (P)
- Complete (C)

Integrity (I)
- None (N)
- Partial (P)
- Complete (C)

Availability (A)
- None (N)
- Partial (P)
- Complete (C)

Figure 3: CVSS Base Metrics

Threat Intelligence

> Threat intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that you can use to inform decisions regarding the subject's response to that menace or hazard.

The idea of threat intelligence is to extend security awareness beyond the internal network by consuming intelligence from other sources Internet-wide related to possible threats to your organization.

Threat intelligence cycle according to foster:

1. Collection
2. Processing
3. Analysis and production
4. Dissemination
5. Planning and direction
6. Collection

Plan to consume it, and how threat intelligence will integrate with your SOC technologies and processes.

Compliance

Monitoring the compliance of your system against reference configuration templates gives you an opportunity to detect changes and existing configuration problems that could lead to a possible breach.

Many tools, such as Nessus or Nexpose provide compliance tools that enable them to remotely login to your system and analyze a reference benchmark.

Automate the system compliance process and link it to the risk management and incident response practices.

Ticketing and Case Management

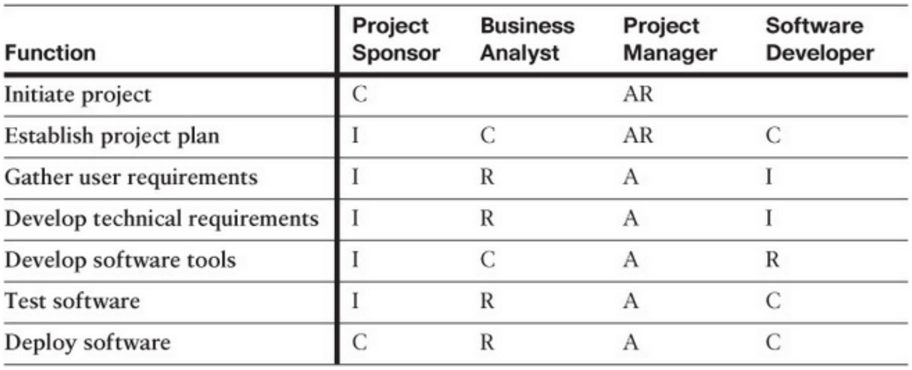
The SOC team is responsible of tracking potential incidents, from the moment they are made aware of them, until they are resolved.

A key point to consider is that remediating some incidents, may require sponsoring of some teams external to the SOC, like business teams. They should also be on board.

The RACI Matrix is a valuable tool in managing roles and responsibilities when an organization is in a state of change.

R = Responsible, A = Accountable, C = Consult, I = Inform

The steps to built the matrix are:



Function	Project Sponsor	Business Analyst	Project Manager	Software Developer
Initiate project	C		AR	
Establish project plan	I	C	AR	C
Gather user requirements	I	R	A	I
Develop technical requirements	I	R	A	I
Develop software tools	I	C	A	R
Test software	I	R	A	C
Deploy software	C	R	A	C

Figure 4: RACI Matrix

1. Identify all the processes or activities known (List on the left of the matrix)
2. List all the roles at the top of the matrix
3. Create values to reference (R, A, C, I, AR)
4. Verify that every process has a R, and that there is only one R per process.

Collaboration

The SOC team should have a central collaboration platform that stores, manages, and provides access to documents. Communication is important within the SOC, so make sure to provide access to internal messaging, websites, and documentation.

SOC Conceptual Architecture

To get the most out of a SOC, the several products, (monitoring systems, documentation, communication) should be able to coexist in a cohesive architecture.

Proposed architecture, consisting on Sources, Alerts and Actions, technologies and their relationships, recollection areas.

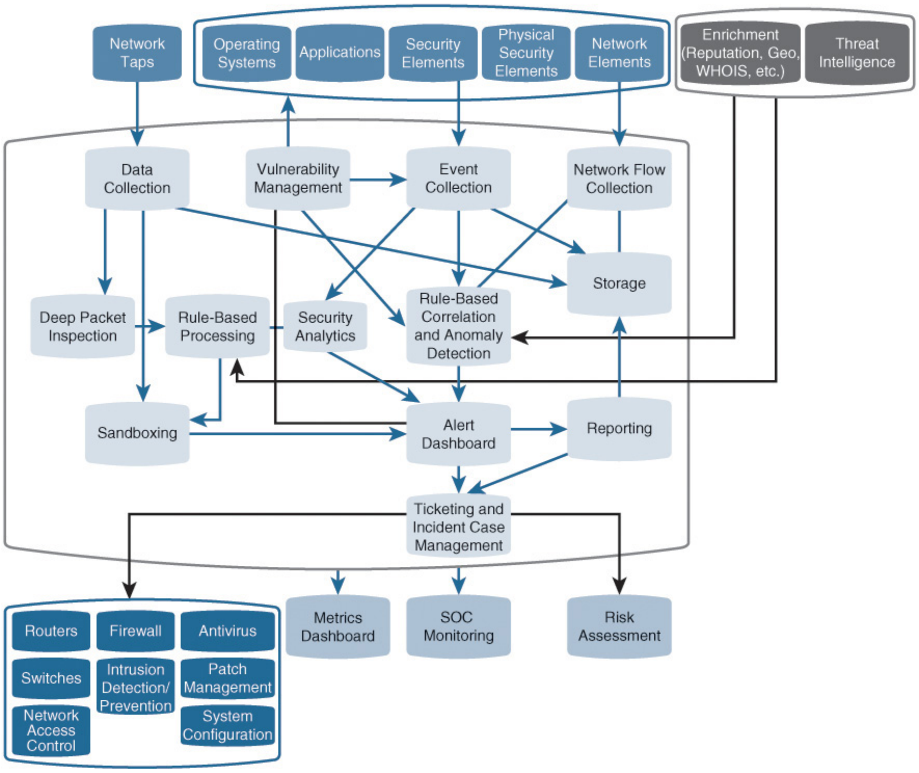


Figure 5: SOC Schema