PREDRAG ŠKUNDRIĆ
Yunycom d.o.o,
Belgrade, Serbia,
E-mail: predrag@skundric.com

VANJA KORAĆ
Mathematical Institute SASA,
Belgrade, Serbia,
E-mail: vanja@mi.sanu.ac.rs

ZORAN DAVIDOVAC
Mathematical Institute SASA,
Belgrade, Serbia,
E-mail: zorandavidovac@mi.sanu.ac.rs

Received: September 28th 2021
Accepted: November 20th 2021
Original research article
007:004.056.5
COBISS.SR-ID 55274249
https://doi.org/10.18485/arhe_apn.2021.17.11

TECHNOLOGICAL ASPECT OF THE GLOBAL ARCHITECTURE OF THE SECURITY OPERATION CENTRE OF AN ORGANISATION

ABSTRACT

The global architecture of the Security Operation Centre within an organisation comprehends the implementation of all modules and systems that generate security events, collect security events, store security events and analyse and react in the case of detected incidents. This paper highlights the pre-requisites for the correct implementation of the Security Operation Centre of an organisation.

KEYWORDS: SOC, SECURITY OPERATION CENTRE, INFORMATION SECURITY.

A prerequisite for the correct implementation of the architecture (its modules and systems) of a SOC (Security Operation Centre, or Organisation Security Management Centre) is to monitor the entire IT infrastructure or to monitor the part that an organisation wants to protect in the best manner possible. Therefore, we will follow the functional steps so as to adequately describe the purpose and concepts of each separate part of the architecture, which can be seen in the figure below.

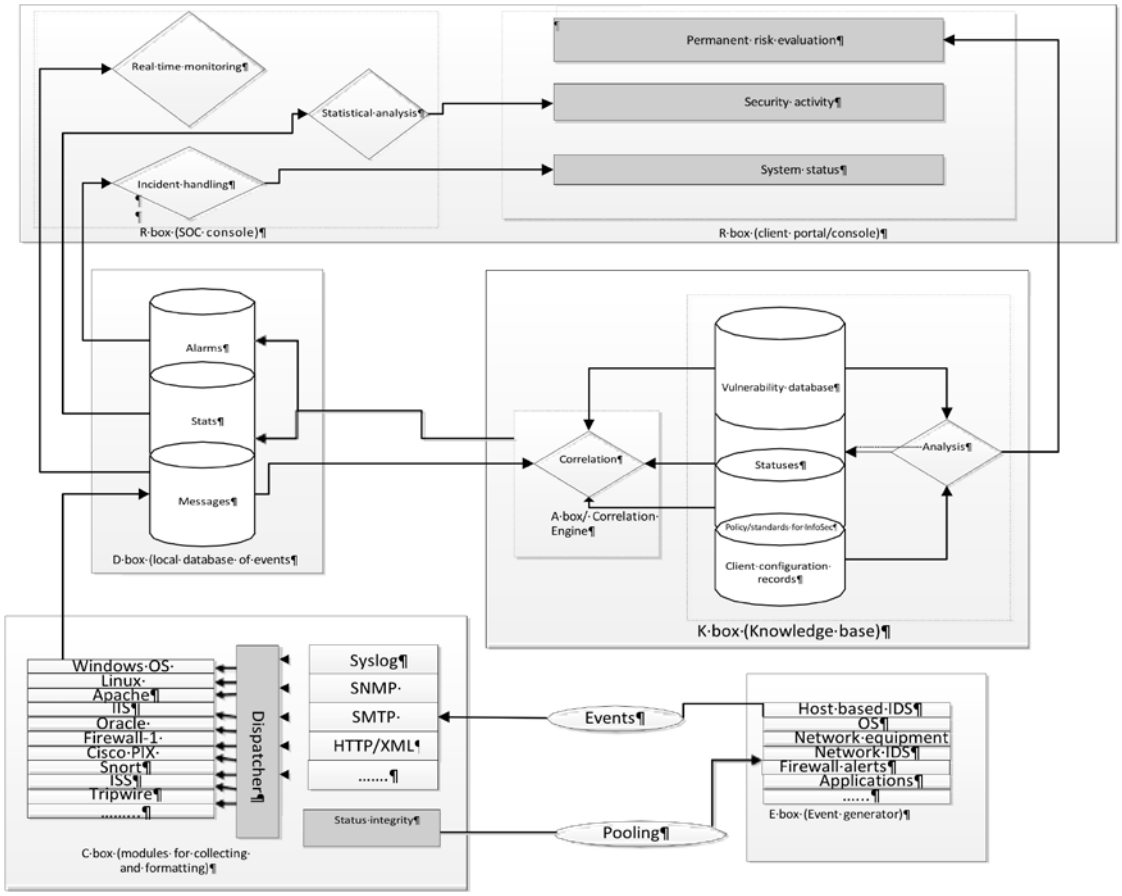
Before setting up sensors and designing correlation or analysis rules, it is necessary to assess the overall security level of the IT infrastructure that is to be monitored. This will enable us to establish whether an intrusion path can effectively lead to an intrusion into the desired system and to also establish all the potential critical points linked to such an intrusion attempt. Primarily, this comprehends the assessment of risks for the entire infrastructure, which represents the basis for defining all necessary activities for protection and. hence, the basis of the monitoring system as well (*Ganame, Bourgeois, Bidou and Spies 2006*). Additionally, a security policy should be defined regarding access clearance and allowed operations.

TECHNICAL AND ORGANISATIONAL INVENTORY

The evaluation of the security level can be divided into two parts, namely:

vulnerability assessment.

assessment of the system criticality level.



SOC architecture (Ganame, Bourgeois, Bidou and Spies 2006)

These two assessments must be performed before defining the actual level of collection of system records from a device. Also, after the collecting is finished, these two types of information should be included in some form of documentation. In our case, this type of documentation represents a knowledge base. In this implementation, this type of record is found in the SOC segment called *Client Configuration Record* (CCR). The collecting of this data can be performed in two ways, namely:

by using the Black box approach.

by using the White box approach.

The source of data for the first method of data collection is penetration testing without knowledge of the client infrastructure. This type of approach is widely used and gives results very quickly. The second method is, however, more appropriate if exhaustive data collection with a detailed list of monitored systems is desired and a detailed intrusion path generation is provided.

System criticality should be defined in accordance with the relative consequences that may occur should the system be penetrated. In order to reduce the subjectivity of such an operation, it is necessary to use standard taxonomy when determining attacks on information systems, as well as classification in accordance with valid scales, defined within the company, and if the company in question has no developed mechanisms, examples from accepted practice should be used.

SYSTEM VULNERABILITY DATABASE

The vulnerability database contains information on detected breaches in information systems, as well as data on insecure behaviour that could affect or does affect the overall security level, i.e. those that an attacker could use for an intrusion (*Ganame, Bourgeois, Bidou and Spies 2006*). The format of the database must enable the inclusion of the following vulnerability types:

Structural vulnerabilities, i.e. vulnerabilities that are specific to a given piece of software, such as buffer overflow, format string, etc. This part of the database is, obviously, the easiest one to implement, fill in and maintain. Most of these processes can be scripted, since the information is available from public sources, such as public mailing lists, recommendations for software setup and web locations. However, the level of validation and correlation (if multiple sources are used) should be mandatory, and it is necessary that a professional team should do it.

Functional vulnerabilities that depend on the configuration, operational behaviour, users, etc. These vulnerabilities differ from the previous ones because they are deeply dependant on their environment. For example, an NFS mount should be considered a functional vulnerability, since an intruder could access an account/host, which would enable him to mount a file system. Therefore, it will be assumed that many such vulnerabilities are present in systems, but they can be considered "inactive" as long as at least one of the required conditions is not met with.

Topology-based vulnerabilities, including the impact of networking and their consequences. This part of the database includes network vulnerabilities (sniffing, spoofing, etc.), as well as the impact of filters on the intrusion path. Such vulnerabilities cannot be included in the vulnerability database unless the IT infrastructure topology itself is taken into consideration.

It is necessary to emphasise here that we have considered vulnerabilities solely in terms of the technical/technological aspect of the organisation of the IT infrastructure, and not vulnerabilities in terms of the formal functioning (security policies and procedures applied within an organisation) (Škundrić *2017*).

SECURITY POLICIES

The next step in the implementation or control of the monitored system inventory is organisational, or, more precisely, the implementation and overview of the aspects of security policies that would affect the creation of events and/or report processes, or responses (*Ganame, Bourgeois, Bidou and Spies 2006*).

Policy aspects that have to be reviewed, or properly configured are the following:

authorisation process.

the process of testing and reviewing procedures.

These two aspects will provide information regarding the behaviour that sensors should send to event collectors. On the basis of this, we can conclude that events such as access to the systems by an administrator, scanning ports on the network and network segments etc., depending on the policy itself, can be treated as events important for monitoring the overall security level of the information system of a company. Aside from the above mentioned, policies can be used to prevent some employees from accessing certain system resources, and should such an attempt be made, an alarm can be raised.

All the mentioned or similar examples are part of predefined rules that must be contained within the knowledge base.

STATUS EVALUATION

The last, but not the least important part of the knowledge base is the level of security of the system that is being monitored. In the status evaluation itself, an actual event is correlated with predefined vulnerabilities defined in the system vulnerability database, as well as limitations linked to security policies. This mechanism should perform an analysis of vulnerability by system, i.e. a list of vulnerabilities that every system is exposed to, their relative impact, or criticality of the detected system vulnerabilities, as well as possible attack paths that could be used.

GENERATING EVENTS

E-boxes are responsible for creating events, in the form of system events. These boxes should be set up to generate the largest possible amount of information, or records, without affecting their normal operation. These records can be sent in real time or stored for a given period of time before being sent to C-boxes, which collect all the events generated by E-boxes. A general recommendation is that all the records be sent to the central records management system, in our case, C-boxes, in order to avoid all manner of threats.

C-boxes contain mechanisms for correlating system records (event correlation is a technique for giving meaning to a very large number of events and giving priority to specific events that are very important within that mass of information), and this is achieved by searching and analysing relationships between events. On the basis of this, we can conclude that C-boxes are responsible for the qualification and removal of unnecessary content from system records. Nevertheless, this theoretical approach to the activities of C-boxes should be accepted with caution, since it is completely or partially inapplicable in certain cases, especially in terms of system performance. A typical example of the impossibility of application would be the analysis of system records for applications, or records generated by operating systems. Unlike the previously mentioned cases, this approach can be used in the management of records created by IDS devices/systems.

The best option is filtering records at the very source of information, system records, i.e. at the E-box itself. This form of filtering would significantly reduce the number of records that would be forwarded to the C-boxes. In order to perform this activity in a successful manner, the E-box events should be qualified before generating each record. The qualification of these events is determined by two factors:

- Structural specification; in this case, some of the events will not be created if they concern some system components that are not present on the system that is being monitored. This type of filtering is typically implemented on IDS systems and firewall devices, i.e. devices used for filtering network traffic.
- Polices based on security policies; these filters are set up with the goal of avoiding the generation of events that are in accordance with the security policies of the company. A typical example are commands that can be allowed to certain users at a certain time, that is to say, ports scanning activities from certain network devices, IP addresses, etc.

Even though filtering in advance of the C-boxes reduces the number of system records that need to be processed, this approach also has several important drawbacks. C-boxes are used for collecting system records from different sensors and translating them into a standard format that can be comprehended by the system (Škundrić, Korać and Davidovac 2020: 233)

The first of the drawbacks is the very difficult maintenance of filters distributed in this manner. The consequence of this method of traffic filtering is the existence of very strict procedures that manage changes on the system, in which it must be defined that every change on the system also requires the evaluation of filters. On top of all this,

most filters are created at the application level, hence, they use a large number of different configuration files, which significantly increases the complexity of their management.

The second drawback is the real risk of cancelling certain alarms and security records.

The conclusion is that filtering systems records can result in the loss of records with which it is possible to perform an adequate forensic analysis in the case of detected problems, i.e. there is a real danger that certain system records that are not important in one moment can become of utmost importance in another.

GATHERING AND STORAGE DATA COLLECTING

The real value of gathering, collecting and analysing data, or system records, is in finding within the forest of data actual data that can have a certain value for the organisation itself (Škundrić*, Korać and Davidovac 2020: 238*).

Collecting data from heterogeneous sources comprehends the existence of two types of agents: protocol agents and those in charge of applications. The first ones gather information from E-boxes, while the others parse information in order to store it in a "pseudo-standard" format. These two modules are connected by a dispatcher. Such an architecture allows for a high level of availability and even load balancing of the system, which can be set at any level within the infrastructure.

PROTOCOL AGENTS

Basic functions

Protocol agents are designed to receive information from specific transport protocols, such as syslog, snmp, smtp, html, etc. They function as server applications and their sole purpose is to listen to the connections, i.e. the traffic coming from E-boxes and to collect data that will be available to the dispatcher.

The simplicity of such agents makes them easy to apply and maintain.

Data in its original format is most commonly stored in the form of simple files, even though direct transmission to the dispatcher via named pipes, sockets or shared memory provides better performance.

Performance and availability

An interesting part of this approach is the ease with which a large number of agents can be distributed, since they represent simple applications that do not share information between themselves. Therefore, it is possible for very large systems, even server farms, to be connected via syslog or snmp, to the SOC and to be serviced by standard HA and LB equipment. Cluster architecture is also one of the options.

The goal is to ensure a data collecting platform that can be scaled according to needs and also have high availability, regardless of which data collection protocol is applied.

Security

From the security point of view, the most important thing is to ensure the integrity of data collected by the agents. This is especially important if the data will be transmitted to the final processing point via a shared network or a network that is considered to be insecure.

By looking at the TCP/IP protocol architecture, we can conclude that most data collection protocols rely on the UDP layer of this protocol.

It seems that it is necessary to encapsulate such data into secure channels in order to ensure that it will reach the data collection agent unaltered or compromised in any other way (the CIA approach should be applied – *Confidentiality, Integrity, Availability*). Therefore, it is necessary that all three conditions are met in order to ensure security quality, but it is also necessary to

find the optimal balance, without compromising the functioning of the organisation itself (*Korać, Prlja and Diligenski 2016*). This final reason also concerns the data sent via TCP (the same as that sent via smtp or http). However, in order to maintain a high performance level and enable a better functioning of the HA and LB, it would be wise to perform data encryption operations on appropriate equipment, on both sides of the communication line (Škundrić *2017*).

DISPATCHER AND APPLICATION AGENTS

The purpose of dispatchers is to determine the form, i.e. the type of source event, and then to forward the original message to the appropriate application agent. In this case, the implementation is relatively simple when a specific pattern is found for every type of source from which the data could have been obtained.

Autonomous operations performed by a dispatcher are as follows:

listening to incoming channels from protocol agents, such as sockets, named pipes, V-system message queues (*mqueue*), etc.,

verifying pattern matching against a pattern database, which should be incorporated into the system previously so as to prevent the endangerment of system performance. This database contains patterns that are specific to each pair (E-box type, Xmit protocol), because numerous event creators use messages in different formats, depending on the transmission protocol, and

sending the original message to the agent for specific E-box applications via suitable outgoing channels.

APPLICATION AGENTS

Application agents are specific to each pair (E-box, Xmit protocol). They format messages so that they match the generic model from the message database (Škundrić *2017*).

Autonomous operations performed by agents for applying collected data are as follows:

listening to incoming channels from protocol agents, such as sockets, named pipes, V-system message queues, etc.,

parsing the original message into standard fields, and

transmitting the formatted message to the appropriate D-boxes. Any type of channels can be used here, depending on the nature of the D-box (database, connected indicators, etc.).

DATA FORMATTING AND STORAGE

Two types of data have to be formatted in a "standard" manner (i.e. homogenous and comprehensible to any SOC module): host entries and collected messages.

Host (client) entries

Unique client identification

The need for a standardised structure of client data appears in the following cases:

sensors can transmit client information in the IP address or FQDN (*Full Qualified Domain Name*) format,

multi-homing techniques provide the possibility of multiple IP addresses for the same physical system,

virtual client techniques provide the possibility of multiple FQDNs for the same physical system, and

HA and LB systems can disguise the existence of multiple systems behind a single IP address or FQDN.

The identification of clients by either their IP address or FQDN does not appear to be reliable. Moreover, in the constant need for performance, a reverse DNS lookup cannot be performed for every new (IP address) FQDN that is detected in

records.

Furthermore, it is necessary to rely on independent identification, which does not depend on the IP address or FQDN. As the only acceptable solution, a host/client token is identified.

Data analysis and reporting

Basic operations that result in the creation of alarms are:

- correlation,
- structural analysis,
- intrusion path analysis, and
- behaviour analysis.

Correlation represents a stand-alone activity that leads to the creation of contexts for records so as to conduct certain analyses later and establish whether characteristics of attacks or any malicious behaviour can be found in the records.

Structural analysis can be compared to an advanced method of matching patterns, used to determine which possible events have led or could lead to the compromising of the security of an information system. Examining the method of intrusion is the next step, the result of which is information on the exposure of the monitored system to a given attack or type of attacks, if the generic analysis is used. After that, behaviour analysis will integrate elements of the security policy with the goal of determining whether a given attack is possible or allowed.

The purpose of the listed activities is to generate alarms that are activated only in cases of oversights being found in the structure that allow certain types of intrusions (e.g.: scanning, fingerprinting, exploitation, backdooring or deleting an attack history), but also take into account defined security policies, and the criticality of the targeted systems.

Interfaces

When it comes to the SOC, there are two basic types of consoles:

SOC console,

end user port.

SOC console

The SOC console, i.e. the R-box, is primarily intended for internal analysis and usually represents unformatted data from different parts of the SOC system, such as the K-box. There are three interfaces within the SOC console:

Real-time monitoring interface, which provides data in its original form, obtained from the message process within the K-box. This approach enables the basic function of cleansing records, such as the egrep function, in order to extract certain messages that can be used for debugging, in-depth analysis or, in case of an event, re-creation.

The incident management interface represents an internal mechanism for generating and managing incident related tickets, as well as incident management. This mechanism provides quality alarms, as well as a certain amount of data that can be used in the process of debugging, i.e. control points of the information security incident management process.

The statistical analysis interface provides data in its original format that can be used for statistical purposes at a predefined interval. This interface is usually used as an input parameter for a graphic representation of specific information.

End user port

The end user port provides visual communication with the monitored information system. It is designed with the purpose of displaying different levels of reports in a format that is readable for end users, as well as complexities, depending on the person for whom the report is intended (as

opposed to the SOC console, which is intended exclusively for experts to use in cases of troubleshooting the system itself). Information that can be found at the end user port is intended for all parties involved, from the manager, the engineer, and up to the people who deal exclusively with information security. The console itself is divided into three basic parts:

Permanent risk evaluation interface – provides information on the current security levels of monitored configurations and versions of the system software. It provides information on the overall security level, vulnerability and criticality characteristics and descriptions, intrusion scenarios and patch and configuration details.

Security activity – medium-term or long-term reporting that provides general data on the type of intrusion, frequency, sources and consequences on the monitored system. At a lower level, it is used in order to determine movement and identify specific details, such as recurring attack sources or most frequently attacked services that should be monitored.

System status – represents the interface in "pseudo-real time" for the end user, which allows direct tracking of open incidents, systems that are being attacked and intrusion paths activated by intruders.

CONCLUSION

Proper responses to attacks most commonly depend on the organisation and the procedures applied by the attack response teams. The responses vary in the range from passive monitoring and collecting information up to the emergency shutdown of the attacked system by reporting the incident to the appropriate CERT (*Computer Emergency Response Team* – a team that completely resolves the problem in terms of communication with all stakeholders, as well as technical-technological changes on the system with the goal of removing the consequences. CERTs can be local and global and, depending on the organisation, they act globally or locally, with more or less technically oriented activities). Naturally, an appropriate response should be determined before an attack occurs, procedures must be validated, and then safely stored (primarily in terms of integrity) and made available to monitoring teams.

Simply put, a certain level of escalation must be defined in the SOC in order to ensure a quick and efficient response, in parallel with the use of appropriate human resources.

The first level should be those referred to as agents, i.e. technical intermediary staff that are capable of spotting events created by A-boxes. The second level should be a team of technical experts. They are responsible for analysing intrusion events which have not been defined *a priori.* Their priority is to qualify events by using the SOC console interface and to provide a temporary solution for the first level agents to apply. The third level should be a "laboratory" in which suspicious packages, system operations etc. should be re-examined so as to determine the nature of the unknown attack and to provide a fully qualified response procedure. The laboratory should also be responsible for contacting the vendors of operating systems, applications, hardware, etc. in order to design patches and/or to apply them. In its primary form, a "laboratory" represents one of the sandbox solutions.

BIBLIOGRAPHY

Ganame, A., Bourgeois, J., Bidou, R., Spies, F., 2006
Evaluation of the intrusion detection capabilities and performance of a security operation center, SECRYPT 2006, *Proceedings of the International Conference on Security and Cryptography, Setúbal, Portugal,* August 7-10, 2006.

Škundrić, P., 2017
Koncepti i implementacija Centra za upravljanje bezbednošću kompanija, Ibis instruments, Beograd.

Škundrić, P., Korać, V., Davidovac, Z., 2020
Process management within the security operation centre of an organization, Archaeology and Science 16, Centar za nove tehnologije Viminacium Arheološki institut Beograd, ISSN 1452-7448, UDK 007:004.056.5, 659.23:004.056.5, COBISS. SR-ID 29189385, p. 237-242 Beograd, 2020.

Škundrić, P., Korać, V., Davidovac, Z., 2020
Security Operation Centre Modules – Technological aspect, Archaeology and Science 16, Centar za nove tehnologije Viminacium Arheološki institut Beograd, ISSN 1452-7448, UDK 007:004.056.5, 659.23:004.056.5, COBISS.SR-ID 29249545, p. 231-235, Beograd.

Korać, V., Prlja, D. and Diligenski, A. 2016
Digitalna forenzika, Centar za nove tehnologije Viminacium, Arheološki Institut, Institut za uporednopravo: Beograd.

REZIME
TEHNOLOŠKI ASPEKT GLOBALNE ARHITEKTURE CENTRA ZA UPRAVLJANJE BEZBEDNOŠĆU ORGANIZACIJE

KLJUČNE REĆI: SOC, CENTAR ZA UPRAVLJANJE BEZBEDNOŠĆU ORGSNIZACIJE, BEZBEDNOST INFORMACIJA.

Globalnom arhitekturom centra za upravljanje bezbednošću u okviru organizacije podrazumeva se implementiranje svih modula i sistema kojima se vrši generisanje sigurnosnih događaja, prikupljanje sigurnosnih događaja, skladištenje sigurnosnih događaja i analiza i reakcija u slučaju detektovanih incidenata. Određeni nivo eskalacije mora se definisati u SOC-u kako bi se osigurala brza i efikasna reakcija, paralelno s korišćenjem odgovarajućih ljudskih resursa U radu su istaknuti preduslovi pravilne implementacije Centra za upravljanje bezbednošću organizacije.

* * *