INNOPOLIS UNIVERSITY

SYSTEM AND NETWORK ENGINEERING

SoC for $0

Author:
Akhmed GASANOV
Dmitry FOFANOV
Sergey GREBENNIKOV
Tlhologelo MPHAHLELE

Supervisor:
Konstantin URYSOV
Teacher Assistant:
Anatoly TYKUSHIN
Kirill SALTANOV

March 9, 2018

***Abstract*—The lack of security in information systems is due to the lack of desire to spend money on this. Therefore, free solutions will always have high demand. There is a reasonable challenge - is it possible to provide security of information systems for free? It becomes quite feasible task, thanks to enthusiasts who invest their efforts in open source projects.**

I. Introduction

A security operations center ("SOC") is a facility where enterprise information systems (websites, applications, databases, data centers and servers, networks, desktops and other endpoints) are monitored, assessed, and defended. A SOC is related to the people, processes, and technologies that provide situational awareness through the detection, containment, and remediation of IT threats. SOCs typically are based on a security information and event management (SIEM) system which aggregates and correlates data from security feeds such as network discovery and vulnerability assessment systems; governance, risk and compliance (GRC) systems; web site assessment and monitoring systems, application and database scanners; penetration testing tools; intrusion detection systems (IDS); intrusion prevention system (IPS); log management systems; network behavior analysis and Cyber threat intelligence; wireless intrusion prevention system; firewalls, enterprise antivirus and unified threat management (UTM).

A. Description of the problem

The main problem considered in the research work is the lack of a method for building security operations center for free, which guarantees the safety of an enterprise's information systems. It is required to investigate the technical ways of implementing such a method. Its implementation should help the company to improve the security of its systems.

B. Research Question

Following from the introduction, the research questions are defined as:

- What are the current best practices to build SOC(proprietary and open source)?
- Which current open source technologies for SOC are used and what are their benefits?
- How effective is the designed solution?

By summarising these questions, the following main question is formed:

"How to build an effective SOC with open source technologies?"

II. Theory

At the epicenter of a successful SOC should be operational excellence driven by well-designed and executed processes, strong governance, capable individuals and a constant form of self-reflection and improvement against cyber adversaries. As much as SOC might be a technical aspect of any organization for SOC to succeed and add value it needs to support business objectives and effectively improve an organization's risk posture. A well-designed and implemented SOC can utilize and maximize existing security infrastructure in an organization such as(anti-virus, IPS, and IDS). By combining existant and often isolated security features SOC can extend the benefit of these individual applications throughout the organization.

To improve the security posture of an organization SOC must be both active and proactive whilst carrying out everyday tasks such as vulnerability management.

III. The actually conducted research

A. Collected Information

During the analysis, valuable articles and complete projects were found but they were overly technical on how to build SOC. However, a satisfactory solution was not among the found documents.

One of the most successful examples of the open source SOC was provided by Cisco company as the OpenSOC project [9] with the aim to help organizations to manage the log data in their security strategy. It is done by integrating a number of elements of the Hadoop ecosystem such as Storm, Kafka, and Elasticsearch. OpenSOC gives a platform with a full-packet capture indexing, storage, data enrichment, stream processing, batch processing, real-time search, and telemetry aggregation.

As an open source solution, OpenSOC allows creating an incident detection tool as using the framework: organization can customize their incident investigation process.

Thus the Cisco "OpenSOC" project repository [10] with the collection of submodules looks attractive to use, however, our goal was to develop our own open SOC and use it in specific virtualized infrastructure.

Another Github open solution for SOC was found the "Elasticsearch, Logstash, Kibana with Curator and Beats support" project [11] which was ready to run Docker configuration for setup ELK stack fast - it is handy and quick to deploy, pretty scalable (horizontally), but not modifiable under our infrastructure and topology.

One more open source project published on Github is "quick-elk" [12]. Just by cloning the repository and running the installation bash script one can automatically download last versions of Elasticsearch, Logstash, and Kibana and install them in a local directory (Kibana installed as an Elasticsearch plugin). The main drawback from our perspective was that installation had the lack of instruments required for our SOC.

"Suricata-Elasticsearch-Logstash-Kibana" project [13] contains Suricata-ELK interacting system, but again in Docker that is usually pretty handy, but in our case when it is just a part of the entire structure it does not fit.

The best documented from our point of view tree-like article was found "Build Open Source Security Operations Center (SOC)" [14]. In this paper step-by-step instructions on how to deploy SOC using open source tools providing all the requirements and dependencies. The SOC offered consists of:

- Full Packet Capture (FPC) as a Moloch java application for sniffing and creating PCAPs files, traffic metadata storing and session data representing;
- Network Intrusion Detection System (NIDS) as a Snort application like a standard (Suricata optionally, as it is a fork of Snort);

- Netflow network session metadata ELK application set for collecting data such as source IP address, destination port, number of bytes transferred; plus collecting which receives and processes the exporter output; and a viewer;
- Host-based logs application for embedded devices, like home routers;
- Host Intrusion Detection System (HIDS);
- Anti-Virus (AV) logs like Malware detection - Cuckoo;
- Proxy web and mail logs system.

By looking at multiple visualisation tools for attacks the best we found that fit our requirements was GeoIP [15] because it visualises the different locations from which attacks are originating from. Cybersecurity GeoIP attack map tracks the logs and parses IP addresses and port numbers to show the attackers in real-time. The project first was developed by Sam Cappella, who created a cyber defense traffic visualizer for the 2015 Palmetto Cyber Defense Competition and then rearranged by Matthew Clark under the GNU General Public License.

During the deploying the project we actively were using and were guided by the hard copy set of lectures and seminars in Information Security course of the Moscow Physical-technical University (MIPT) kindly provided by Professor Mike Steinberg, which, in turn, actively involved in the development of the security sector in open source.

B. Architecture

For the virtual test environment, computers are required that match the following hardware configuration:

- Processor: 64-bit Intel Virtualization Technology (Intel VT) or AMD Virtualization (AMD-V) processor, with 2.8 gigahertz (Ghz) or better dual core recommended
- Hard Disk: 500 gigabyte (GB) hard disks 7200 RPM Serial ATA (SATA)
- RAM: 16 GB or higher
- Network Adapter
- Monitor: Dual Super VGA (SVGA) monitors, 17 inches or larger supporting 1440 x 900 minimum resolution

The Xen hypervisor installed on two machines was used to create a virtual environment. The virtual environment consists of the following main modules and submodules:

- Xen server farm:
  - Enterprise network:
    - DHCP Server
    - DNS Server
    - LDAP Server
    - Clients
  - Network devices:
    - Switch
    - Router
  - Monitoring Center (SOC):
    - IDPS (Intrusion Detection and Prevention System)
    - Log Management System
  - Firewall
- Enterprise cloud (Google Cloud Platform):
  - Web Server
- Attacker emulation script

The Figure 1 shows the solution architecture and interaction between different parts of the system.

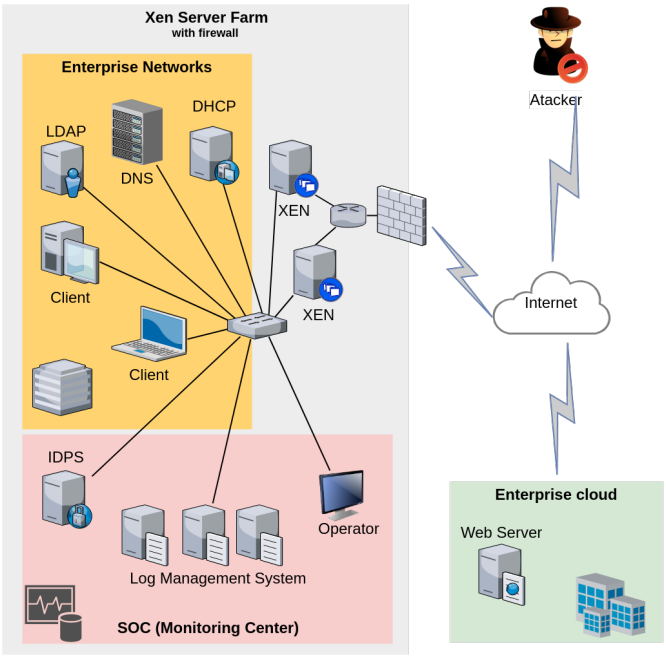

Figure 1. Architecture of the solution.

The conceptual model of the Enterprise Network was designed in the likeness of a typical branch of any enterprise.

There are three different focus areas in which a SOC may be active, and which can be combined in any combination:

- *Control* - focusing on the state of the security with compliance testing, penetration testing, vulnerability testing;
- *Monitoring* - focusing on events and the response with log monitoring, SIEM administration, and incident response;
- *Operational* - focusing on the operational security administration such as identity and access management, key management, firewall administration;

Our Security Operation Center is designed to perform monitoring functions. The focus is on events and responses from monitoring system logs. The next section gives a detailed description of this center.

C. SOC (Monitoring Center)

There are different logging systems with different database models and different approaches for storing and searching for data in collected logs. Elastic stack was selected as a part of our project because it is the only one we found that can receive, store, analyze and represent collected data by itself. This product was analyzed to select the best solution.

The Elastic Stack consists of three modules:

- Logstash - an application that receives, filters, modifies and redirects an input message
- Elastic search - a search and analytics engine based on Apache Lucena

- Kibana - web interface for representation of data from elastic search

Elasticsearch can be run in any OS with Java VM. Elasticsearch is a data schema-free, that means the administrator can create their own types in the system. It creates secondary indexes for each field automatically, supports server-side scripts, triggers. Data can be split into shards and storing on multiple nodes. Elastic search has a configurable eventual consistency(all, quorum, one).

Scheme of the SOC work implementation is shown in the Figure 2

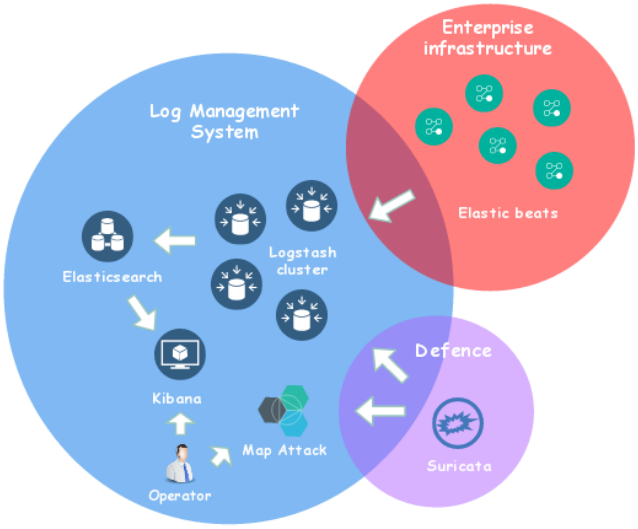

Figure 2. SOC Architecture

In our implementation, the multiple logstash instances obtain the logs from all services of our infrastructure (It has been done to mitigate load) and deliver it to the elasticsearch. We don't use elasticsearch cluster to store logs by reason of resource limitation. The elasticbeats was used as the log-pushers.

Suricata was chosen as intrusion detection system (IDS) in real-time. Suricata inspects the network traffic using extensive rules, sign language and has powerful Lua scripting support for detection of complex threats. With usage standard input and output formats like JSON integrations with tools like Elasticsearch allows us to achieve pretty good interaction between the monitoring system and Suricata. Finally, all of IDS logs come to elasticsearch.

Suricata also can be compiled to be used as intrusion prevention system (IPS). Despite the fact that we use only IDS in our project, we view Suricata as a potential defense of our infrastructure by adding IPS opportunities in the future.

In the SOC presented to visualize the attacks on the organization in real-time the GeoIP attack map visualizer was connected. The data server of the visualizer follows log files and parses our source IP, destination IP, source port, and destination port displaying it on the screen. As the program relies entirely on syslogs and our infrastructure is based on different physical/virtual machines it was necessary to correct its behavior by accepting the logs from Suricata remotely in JSON-format. One of the possible solution how to route the Suricata logs traffic to the GeoIP attack map by modifying to work with Suricata's eve.json file was described in the found article [17].

IV. Proof of concept

According to research conducted in section III-C the logs of all services including Suricata are delivered to a monitoring center where the operator can react to attacks and any other anomaly of the information system in a company. The visualization of the attacks on the organization with usage GeoIP attack map allows operators to observe it in real-time.

V. Discussion

Despite that a monitoring center was designed and implemented, this is not enough for a real Security Operational Center. In this architectural solution, there is no automatic system for preventing cyber attacks. This model guarantees prompt notification of the operator as a result of various incidents but does not guarantee an automated timely reaction to the incident. This is a significant drawback in the real combat situation.

VI. Ethical issues

This project corresponds to all ethical norms of morality and in its research does not violate any laws related to personal data and statistical information about user activity.

VII. Conclusions and suggestion for further research

When building corporate systems, a significant role should be given to security. That in practice is very rare. Basically, companies do not want to spend money on security until an incident occurs. In our work, we offer a free solution that could help companies improve their level of security. This research allows companies to carry out free monitoring system solutions and is the first step in improving this practice in the industry.

In addition, there are some potential directions for improving the existing methodology, which is valuable for further research. These include the automation of the incident response system, the development and implementation of an intrusion prevention system, and the analysis and refinement of the existing model.

For a complete SOC solution to work buy-in is needed from all aspects of the organisation in which the intended system is to be deployed. Top-level management needs to view a SOC center as a necessity instead of an expense and the SOC center needs to help business achieve its goals whilst it(SOC) continues to provide the services for which it was implemented. SOC is an organisation wide system that if implemented properly and with cooperation from all in the organisation it can help an organisation thrive in today's complex cybersecurity environment.

REFERENCES

[1] Logging best practices http://dev.splunk.com/view/logging

[2] Logging Cheat Sheet. https://www.owasp.org/index.php/Logging_Cheat_Sheet

[3] Java Best Practices for Smarter Application Logging & Exception Handling. https://stackify.com/java-logging-best-practices/

[4] Security Operations Centers helping you get ahead of cybercrime http://www.ey.com/Publication/vwLUAssets/EY-security-operations-centers-helping-you-get-ahead-of-cybercrime/$FILE/EY-security-operations-centers-helping-you-get-ahead-of-cybercrime.pdf

[5] What is a SOC (Security Operations Center)?Security Affairs http://securityaffairs.co/wordpress/47631/breaking-news/soc-security-operations-center.html

[6] Building a World-Class Security Operations Center: A Roadmap https://www.sans.org/reading-room/whitepapers/analyst/building-world-class-security-operations-center-roadmap-35907

[7] Discover Logging Best Practices. Part 1: Collecting Logs. https://logmatic.io/blog/beyond-application-monitoring-discover-logging-best-practices/

[8] Kibana – make sense of a mountain of logs. http://kibana.org

[9] The OpenSOC project, Cisco. https://opensoc.github.io/

[10] Cisco Github OpenSOC page. https://github.com/OpenSOC/opensoc

[11] Elasticsearch, Logstash, Kibana with Curator and Beats support. https://github.com/sqshq/ELK-docker

[12] Github quick-elk project. https://github.com/kurtado/quick-elk

[13] Suricata-Elasticsearch-Logstash-Kibana. https://github.com/bulju/suricata-elk

[14] Build Open Source Security Operations Center (SOC). http://www.nmesh.co.uk/mediawiki/index.php/Build_Open_Source_Security_Operations_Center_(SOC)

[15] GeoIP https://github.com/MatthewClarkMay/geoip-attack-map

[16] Github geoip-attack-map page. https://github.com/MatthewClarkMay/geoip-attack-map

[17] HOWTO: Traffic and Attack Map for Suricata. https://samiux.blogspot.de/2016/12/traffic-and-attack-map-for-suricata.html