Callegari, C., Cantelli Forti, A., D' Amore, G., Hoz, E. de la, Echarri, D., García-Ferreira, I., López-Civera, G., 2016, "An architecture for securing communications in critical infrastructure" , ICTE 2016, Proceedings of the 13th International Conference on e-Business and Telecommunications, V. 1, p. 111-120.

Available at http://dx.doi.org/10.5220/0006016801110120

An Architecture for Securing Communications in Critical Infrastructure

Christian Callegari[1], Alessandro Cantelli Forti[1], Giuseppe D'Amore[2], Enrique de la Hoz[3], David Echarri[4], Iván García-Ferreira[4], Germán López-Civera[3]

[1]*RaSS National Laboratory, CNIT, Pisa, Italy*

[2]*Vitrociset S.p.A., Rome, Italy*

[3]*Computer Engineering Department, University of Alcala, Alcala de Henares, Spain*

[4]*Oesia Network, Madrid, Spain*

{*christian.callegari,alessandro.cantelli.forti*}*@cnit.it; g.damore@vitrociset.it,* {*g.lopez,enrique.delahoz*}*@uah.es,* {*igarcia,decharri*}*@oesia.com*

Keywords: Critical Infrastructure, Intrusion Detection System, Intrusion Prevention System, Honeynet, Firewall

Abstract: The disruption of communications in critical infrastructures could have a serious impact on the health, safety, security or economic well-being of citizens or even prevent the effective functioning of governments or other agencies. For this reason, in this paper we present a distributed architecture, named CYBERSENS, aimed at preventing, early detecting, and mitigating cyber attacks to critical infrastructure networks. CYBERSENS is an advanced IDS/IPS system specially tailored for securing communications in critical infrastructures. It's federated architecture, the combination of misuse detection techniques and novel anomaly detection approaches, and the inclusion of mechanisms for self-obfuscation and self-protection, makes our proposal specially suitable for these scenarios.

1 INTRODUCTION

Since the mid-1990's, dramatic experiences caused by natural or man-made disasters made urgent to understand the dependency of our society from those infrastructures that, if disrupted or destroyed would seriously compromise our quality of life and/or overall functioning of the society. Therefore, Critical Infrastructure protection has become a general label for a range of activities undertaken jointly by government and operators of key location, facilities and system to ensure an adequate management risk.

Critical infrastructures (EUCommission, 2004) consist of those physical and information technology facilities, networks, services and assets which unavailability or malfunction could have a serious impact on the health, safety, security or economic well-being of citizens or prevent the effective functioning of governments. Critical infrastructures extend across many sectors of the economy, including banking and finance, transport and distribution, energy, utilities, health, food supply and communications, as well as key government services.

Infrastructure systems are characterised by a high degree of interconnection. Many physical, virtual and logical dependencies are not apparent until a crisis occurs and the connection breaks down. The high level of interdependence can lead to cascading shut-downs. At the same time, smaller and smaller disruptions are enough to cause dramatic consequences in complex systems.

In this paper we present a security system, named CYBERSENS, for the protection of the critical infrastructures from cyber attacks. In a nutshell, CYBERSENS should be able to prevent, early detect, and mitigate cyber attacks, while simultaneously keeping the communications alive and preserving the privacy of the critical infrastructure users.

In more detail, the functionalities of the CYBERSENS system are

- Intrusion prevention/detection: it represents the core activity of the system. The early detection and localisation of cyber attacks is carried out by applying signature-based techniques, effective in detecting well-known attacks, and anomaly based techniques, effective in also detecting novel (e.g., zero-day) attacks
- Privacy preserving data aggregation and export: given its distributed architecture, the CYBERSENS system makes use of probabilistic data structure and algorithms that aim to aggregate, ex-

change, and export private data among the different probes and between the probes and the Main Control Unit (MCU), without disclosing any private information to any external entity

- Traffic masking by encryption and traffic generation: data links between nodes are encrypted using known standard best encryption techniques known to be effective. Traffic padding is provided as an option by producing or injecting fake traffic into data links to prevent a passive traffic sniffing to figure out usage statistics or behaviours
- Attack deviation and redirection – Honeynet: fake working hosts are decoyed at the purpose of being a natural target for incoming malicious traffic behind external level of defences. Analysis of traffic coming from honeynet is used to increase knowledge about malicious techniques in use
- Self-protection: systems aimed at detecting cyber attacks often become the target of attacks aimed at stopping the correct functioning of the system itself, e.g. by increasing consumption of computational resources. Hence, static secure techniques, such as firewalls, are deployed for protecting the system itself from external attacks

It is important to highlight that in our proposal, not only does our system take into account the single critical infrastructure, but it can also protect the interconnection among several infrastructures if needed.

The remainder of the paper is structured in the following way: in Section 2 we present the general architecture of the proposed system, describing all the system components and their functionalities. Then Section 5 details the objectives and the specification of the system, while in Section 3 and 4 we detail the Anomaly Detection Subsystem and the Misuse-based Detection Subsystem, respectively. Finally Section 6 concludes the paper with some final remarks.

2 General Architecture

As stated in the introduction, the main goal of CYBERSENS module is to be able to prevent cyberattacks against the telecommunication networks that enable critical infrastructure normal operation, by performing an early detection of any sign indicator of an attack. This early detection allows to prevent a possible compromise of the IT assets under protection of the system.

To accomplish this objective, an advanced Intrusion Prevention System (IPS) will be deployed, following a distributed approach that allows to meet performance and privacy requirements. To this aim the system must take into account the sensitivity of the data that are exchanged between critical infrastructures and the impact that an attack could have on them. Moreover the developed detection system must not introduce data leaks or new vulnerabilities in the system it is trying to protect. Therefore special measures related with data privacy and the protection of the whole system have to be taken.

From the functional point of view, the different CYBERSENS elements perform a task that is within three main functional blocks: information gathering module, analysis engine, and alert system (Bace, 2000). These three submodules create a sequential flow that allows to cover each phase during a typical cyber incident: monitoring, detection, and mitigation.

In Figure 1 we can observe the overall architecture of CYBERSENS. The main elements of the system are:

- **Local Central Unit:** CYBERSENS central unit is entitled with the actual IPS functionalities. The objective is to combine misuse and anomaly detection techniques to cover a wide variety of attacks and be able to discover and react to non-standard types of attacks and new vulnerabilities.

  Misuse detection techniques are based on the process of data samples looking for known malicious behaviour. They usually work using some kind of signature that describes what to look for in the data to classify it as malicious. Despite the fact it is a really widely used approach they can only detect already disclosed vulnerabilities, so novel attacks that cannot be directly detected with these signatures or inferred from them will not be caught by the detection system. Even so, they perform really well at detecting known attacks and are therefore needed to protect the system from the millions of malicious samples already described by signatures. In contrast to misuse detection methods, anomaly detection aims to detect deviations from the normal behaviour. These deviations or anomalies may be produced by intruders or may be due to hardware or software malfunction. The idea is that this kind of anomalies can be used as early detection indicators of attacks. More details about the detection strategies will be provided in Sections 3 and 4

  Privacy issues are also a concern, since the information collected at each CYBERSENS central unit is very sensible, as it represents the current status of the overall architecture and its elements. Consequently extensive use of cryptographic algorithms and secure message exchange is made in the communication between each CYBERSENS central unit and its probes and also between each

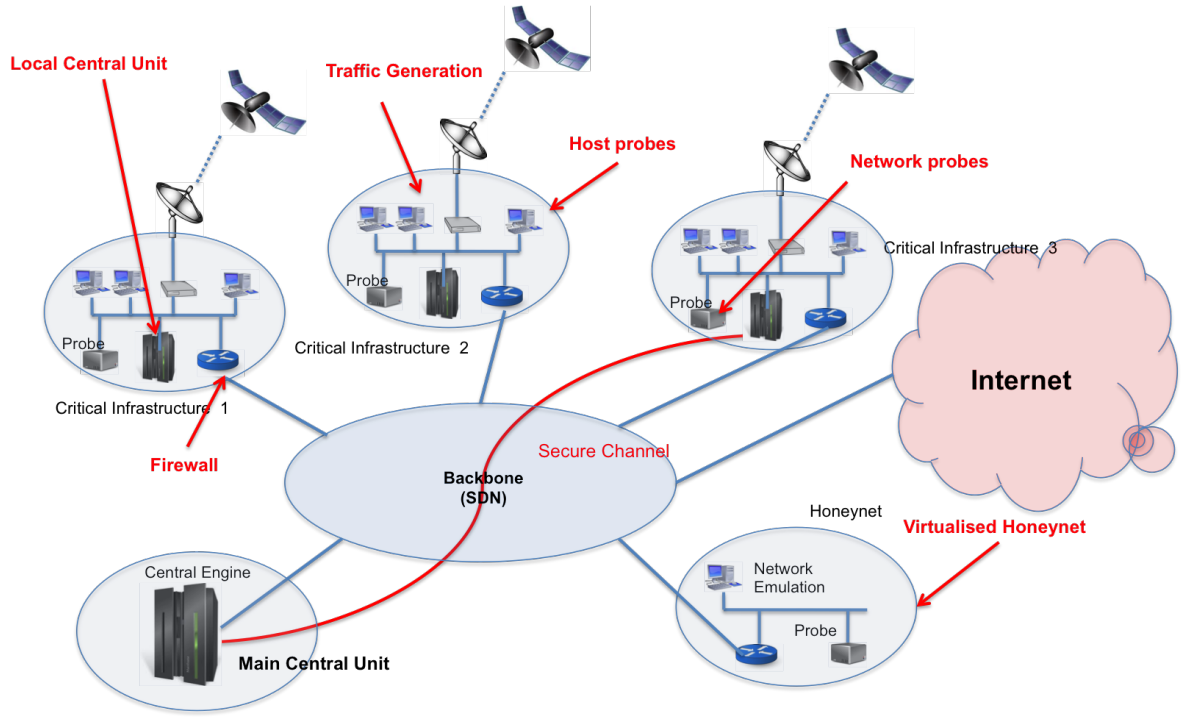

Figure 1: System Architecture

CYBERSENS central unit if needed.

- **Main Central Unit (MCU):** this elements is responsible to collect and correlate the data sent by the local central unit, so as to generate alarms and put the proper reaction strategies in place. For this reason, each local central unit sends to the main central unit information about the detected attacks(e.g., location of the attack and the category it can be classified in). The Common Attack Pattern Enumeration and Classification (CAPEC) (Enumeration, 2013) initiative is used as a common language to describe the attacks included in the alerts. It is composed by a hierarchical structure that easily allows to correlate and cluster related alerts to make it easier to analyse the situation in real time.
- **Network probes:** The purpose of these probes is to collect all input and output traffic that travels through the critical infrastructure network. This function is usually performed by dedicated hardware systems that allow to forward all the traffic that goes through them to another node entitled with monitoring tasks. There exists a wide variety of hardware (mainly industry level switches and routers) that already implements these techniques like SPAN port or port mirroring (Woodring, 2001). Virtualised solutions also have this functionality so even in the case of a virtualised environment inside a critical infrastructure, traffic monitoring can also be performed without acquiring new hardware. Also open source projects like OpenVSwitch or VyOS (virtual switch and router solution respectively) include mirror port capabilities, so they can be easily integrated into a virtualised architecture if needed.

  The second aspect that has to be addressed when deploying a network probe monitoring infrastructure is the aggregation and reduction level made over the collected traffic before exporting it to monitoring or analysis tools. Netflow (Claise, 2004) is a standard proposed by CISCO that includes a format that describes the flows and a protocol to generate and transmit these flows and the statistics related to them to other devices.

  Considering the multi-operator scenario, like the one depicted in Figure 1, each network probe, apart from simply collecting/aggregating the traffic, must also perform some additional operations depending on the required output. In more detail, as it will be clear in the following sections, the probes must either perform some processing (e.g., misuse based intrusion detection) or send aggregated information to Central Unit (one per critical infrastructure), where the traffic is analysed by the anomaly detection algorithms.

  In the latter case, considering that direct analysis

of raw network traffic is not feasible due to computational cost and bandwidth limitations some kind of "additional" aggregation and data reduction is needed prior to detection phase. Both techniques aim to reduce the size and noise in collected data, retrieving meaningful statistics and summaries that allow to analyse it without losing any relevant information or at least being able to configure how much information we are able to loose during the process. Moreover, given the "multi-operator" scenario depicted in Figure 1, where multiple critical infrastructures are interconnected, some privacy constraints can arise that make "standard" aggregation techniques, such as NetFlow, unusable.

For this reason the CYBERSENS network probes have the ability of performing random aggregation (e.g., by using random data structures like sketches (Cormode and Muthukrishnan, 2005)). This kind of data aggregation techniques allow to dramatically reduce the size of the data while retaining enough information about its trend or the number of occurrences of each data instance. Moreover this data structures may be configured to achieve a balance between the data stored in memory and the amount of information they are able to ignore.

Therefore each probe can be configured to aggregate the data as needed depending on the size of the network, the number of elements monitored on it, and the kind of analysis to be performed.

- **Host probes:** The purpose of this agents is to collect information about the applications that are running inside the host. This information includes system calls, software executed, application specific or operating system logs, etc. This data is useful to detect threats that are already inside the host, representing a second barrier of protection against attacks that could bypass network detection. This kind of probes may have an impact in the performance of the monitored hosts, so reduction techniques are applied. The reduction level applied to the data will be configurable from the central unit assigned to it. This reduction can be simply based on the sampling rate or the actual format which the logs are transformed to. Application malfunctionings can also be detected monitoring this kind of logs, this way end-of-life hardware can also be addressed and prevent service outage due to downtime.

  The software category where these probes are included is Host-based intrusion detection system. An an example, in our system, we have used OSSEC (Bray et al., 2008), an open-source project that represents the de facto standard in the industry and have versions for both UNIX and Windows operating systems.

- **Virtualised Honeynet:** One of the main objectives of the CYBERSENS module is to build a resilient system that allows to perform early detection of new threats and attacking techniques. This means that the system itself has to be able to learn from novel attacks to detect 0-day type vulnerabilities, especially in the Anomaly Detection phase. One of the best ways to obtain up-to-date types of attacks is to get them directly from the source, which means to collect information about real attacks. To do so without damaging any real infrastructure the usual practice is to employ honeypots. Honeypots can be described as computer system expressly configured to attract attackers (Zhang et al., 2003). To this aim they mimic the software, configuration and behaviour of a legitimate system similar to a typical server exposed to the Internet. Once an attacker tries to penetrate this kind of system all the actions she makes are logged and monitored for further analysis. This process includes the collection of all traffic, commands executed, file changes, etc. With this information an expert can create attack signatures to feed misuse detection tools. Anomaly detection systems can benefit too from the information gathered through honeypots as they can feed the algorithms with anomalous traffic that can be used to train them. The combination of multiple honeypots that tries to create a whole network that resembles a typical company network, is what is called a honeynet(Spitzner, 2003).

  In the context of CYBERSENS, the honeypots employ virtualization techniques to deploy the vulnerable system. Virtualization allows to isolate the honeypot and to be able to gather information of it from the host system. They also allow to deploy different kinds of systems in a single host, making it easier to imitate a network where multiple kinds of host and services are deployed.

  In general, many different types of honeypots are available, usually classified as low or high interaction. Low interaction honeypots tries to mimic a service or only a few services of a system. They are intended to be used as bait and collect information about the level of exposure (i.e. how many attacks are we receiving?) and the source of the attacks (i.e. IP addresses, location, attack attempts). As they only imitate a set of services they are easier to detect by an intruder, who therefore stop attacking it or even tries to attack the honeypot itself. On the other hand high interaction honeypots

are focused on simulating a real system including all typical services. Consequently, the information collected from them is much more detailed and they can also create a timeline of the attack and the techniques employed to exploit the honeypot.

CYBERSENS makes use of a hybrid approach in which bait-like systems are deployed in each critical infrastructure and when an attacker is attracted to it and tries to connect, he is redirected to the actual honeynet, realised by a virtualised dedicated VPN either "internal" to a given critical infrastructure or "external" to all of them. It is important to highlight that all the data collected from the honeynets may also need some kind of aggregation before exporting it to its central unit for analysis, so similar techniques to the ones employed in the probes should be applied.

It is worth noting that the decision of redirecting a particular flow to the honeynet is made by a Main Central Unit. CYBERSENS will only alert that a malicious activity is undergoing in a particular segment of the network.

- **Traffic generating system:** even though all sensible traffic between different critical infrastructures must be encrypted, a passive sniffing attack could be possible if the network is compromised at any point of the communication. Despite the fact that an attacker would not be able to eavesdrop the content of the communication, statistical analysis may disclose some meaningful information. This type of analysis involves the study of the timestamp, size, source and destination of the messages exchanged. This type of data is not usually encrypted and an attacker could link and correlate the instant when a message is sent with side-channel information. For example the message that is sent when an alert is launched may not vary too much from one alert to another, so it will take up similar size during the time and will always be sent to the same IP address. An attacker can induce when an alert is launched if she intentionally generates an alert and correlate the message she sees on the network despite the fact that it is encrypted.

  The purpose of this submodule is to avoid this kind of analysis by generating and injecting fake traffic into the data links that supports each critical infrastructure. This means that an attacker could not distinguish between real and fake traffic when observing it from outside. To accomplish this objective Traffic Flow Confidentiality (TFC) techniques could be employed. These methods increase the entropy level of the traffic patterns that can emerge from the typical behaviour of the users and system that use it to connect to other networks. They do so by modifying the length, frequency or origin-destination pattern. One direct consequence of this techniques is the impact they can have on the performance of the network as they may involve delaying or rerouting packets (Carlen, 2013).

  Therefore this module must be correctly installed in each critical infrastructure, so that it provides protection from eavesdropping inside the critical infrastructure and also in the communications made through the backbone network.

- **Backbone Software Defined Network (SDN):** despite it is not strictly part of the CYBERSENS system, the design of the backbone network that supports the communication between all CYBERSENS central units and the MCU is a critical issue. This network must be flexible enough to be able to meet changing requirements and different performance demands according to the activity performed at each critical infrastructure. The employment of Software Defined Networks (SDN) represents the main approach to build a resilient and flexible network.

  Even though availability issues have been taken into account during architecture design and the CYBERSENS module can act autonomously at each critical infrastructure if needed, a network outage will impact the performance of the detection and reaction process. To address this issue, a resilient backbone network is needed and the SDN approach allows to deploy multiple data links between each node and make use of backup links if needed. Moreover it provides different alternative paths to reach each critical infrastructure even in the case of an attack as the reaction strategies put in place by the MCU may alter the topology of the network to protect some assets.

  Obviously, the backbone network have to meet all the requirements imposed by the rest of the CYBERSENS architecture elements. That means that encryption mechanisms and secure communication tunnelling methods like IPSEC and VPN must be provided by the network and properly configured. SDN may help to develop robust configuration that can be replicated at each critical infrastructure via a centralized management point (SDN controller).

The result of combining all these elements is a solid workflow that enables a robust intrusion detection system. The development of a hybrid approach that combines misuse and anomaly detection

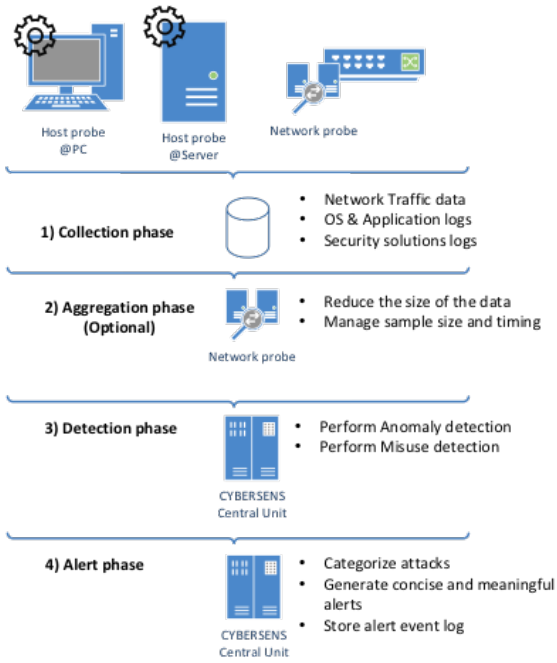

Figure 2: System Workflow

techniques provides a wide protection against known and novel threats. All the processes involved in CYBERSENS module can be summarized in four main categories (as sketched in Figure 2).

1. **Collection phase:** this phase includes the deployment of network and host probes that collect all the data that will be fed later to the detection engine

2. **Aggregation phase:** raw data collected from probes is not feasible to process, therefore dimensionality reduction and aggregation techniques are applied to it before performing actual detection on the data

3. **Detection phase:** misuse and anomaly detection methods will be applied to detect malicious behaviours in the system. This process includes privacy preservation techniques that allow to detect anomalies while protecting sensitive data collected from each probe

4. **Alert phase:** every time an anomaly is detected appropriate measures have to be taken. To do so, CYBERSENS must communicate with the MCU sending alerts and information about the detected incident

3 Anomaly Detection Subsystem

In distributed anomaly detection algorithms, multiple detection probes – distributed in the backbone network – monitor a given portion of the network separately and report the collected information to a single location (namely the Local Central Unit) that analyzes the data and generates the alerts. A sketch of this architecture in provided in Figure 3.

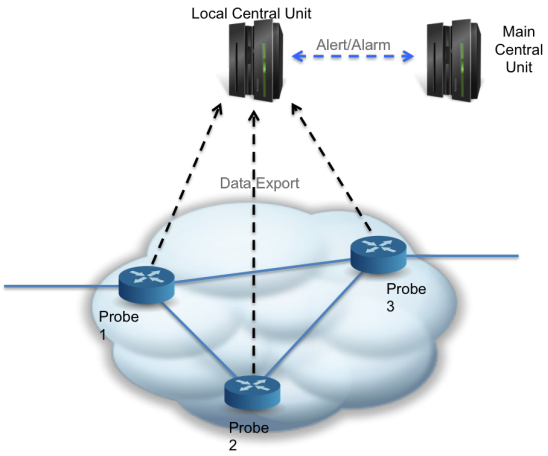

Figure 3: Anomaly Detection system: Architecture

As a first step, the probes have to collect and preprocess the network traffic, so as to extract the needed information to be sent to the central unit, which then performs the actual anomaly detection phase. As an example, by limiting the scope to the simplest case of anomaly detection algorithms that analyzes traffic volumes only, the data collected by the probes can be simply represented by the estimation of the number of traffic flows observed in a given time window. Hence, the first problem to be solved is to provide a reliable estimate of such quantities. Note that this task, that is not trivial when performed over the multi-gigabits links of a backbone network, has been discussed in several previous works, and the use of probabilistic data structure has emerged as a standard approach (Flajolet and Martin, 1985)(Callegari et al., 2010). It is worth noticing here that, when aggregating these structures at the CYBERSENS Central Unit level, several problems (such as not counting duplicated flows – i.e. the flows observed by more than a single probe) must be solved.

CYBERSENS mainly makes use of reversible sketches (Schweller et al., 2004) in this phase to aggregate the traffic and of LogLog Counter (Durand and Flajolet, 2003) for estimating cardinalities, combined together in the LogLog Counting Reversible Sketch data structure (Callegari et al., 2012). Nonetheless, the architecture has been designed to be

general enough to allow to run multiple aggregation and estimation techniques.

In general, the information produced by the probes represent sensitive users information and must not be openly disclosed. From this point of view, the use of probabilistic data structures,as the ones previously discussed, also provides an ideal container to keep all such data in an aggregate and not directly accessible way. However, whenever traffic anomalies are detected, the use of LLCRS provides us a way of identifying the flows (in particular, IP addresses) responsible for that supposed misbehavior.

From the anomaly detection perspective, the CYBERSENS system can run several anomaly detection algorithms, based on different statistical approaches. In more detail, we have studied two distinct approaches:

- **Distributed CUSUM:** in such an approach the probes mainly export simple flow counters towards the CYBERSENS central unit, which detect anomalous behaviors by means of a change-point detection algorithm
- **Distributed PCA:** in this case, the probes perform some preliminary processing over the data (distributed PC computation) and the CYBERSENS aggregates the pre-processed data to perform the actual anomaly detection phase (Callegari et al., 2015)

As a final step, if an anomaly is detected the central unit sends an alert to the MCU, containing:

- timestamp: date and time of acquisition
- spatial ID: ID conveying spatial information associated to the source that has provided the considered data
- cyber threat classification (class): the type of cyber threat revealed, if available
- attacker(s) and victim(s) IP addresses

4 Misuse-based Detection Subsystem

As depicted in Figure 4, where the general architecture of the Misuse-based detection subsystem is shown, such a subsystem is composed of several components, namely a set of distributed sensors and a centralized server.

The sensor represents the software that is installed in the network and is composed of several open-source tools. They provide three main capabilities: IDS, misuse detection and real time monitoring.

CYBERSENS can include different IDS to detect threats in the network and in the system, being

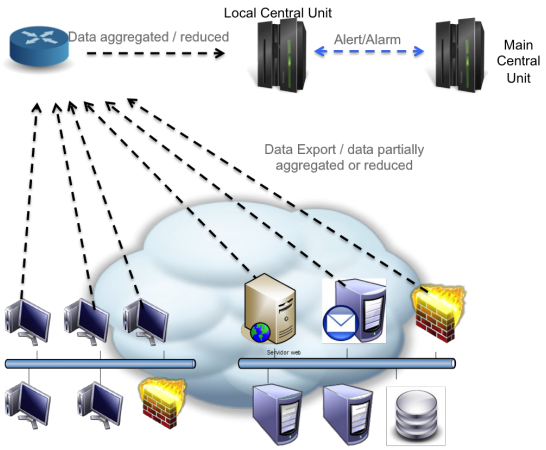

Figure 4: Misuse-based Detection system: Architecture

three top possibilities Snort (*www.snort.org*), Suricata (*suricata-ids.org*), and Bro (*www.bro.org*), because of their wide adoption among security industry and being considered best-of-breed in security. Moreover, apart from these "basic" configuration, a sensor can include other tools to detect specific attacks or gain additional information: e.g., OSSEC (Bray et al., 2008), Ntop (*www.ntop.org*), Nagios (*www.nagios.org*).

All these tools are linked together so as to provide a user with a single integrated environment. Moreover, to combine and process the logs produced by the different tools, as well as the system logs (syslog in Linux) that are continually monitored so as to get all the thread in the system, CYBERSENS makes use of a set of regular expressions, which filter the system logs to a new CYBERSENS log, which is stored in database.

The described processes are achieved by a Cybersens central agent that is located in the sensor system. This agent mainly controls each tool to assure that everything is working fine in the sensor. If some vital process to the system fails the agent sends an alarm to the server.

On the other hand, the server's main responsibilities are: to receive event information from sensors, to correlate events into alerts, and to perform online inventory and sensors configuration. It is worth noting that the communication between the server and the different sensors through the network is performed through a VPN tunnel, which assures a secure communication.

Apart from these "specific" misuse-based detection subsystem elements, also the Local Central Unit is involved, mainly working as a a correlation engine. In more detail, its main goal is to analyze the received events, and determine which ones are valu-

able enough to be reported to the MCU as an alert. To do so, its logic uses several heuristics, for example:

- Events are of the same type, and are directed to the same target
- An event can be associated to known information of the target. For example, if we know the attack affect an SSH server on Linux, and target is a Linux, an alert can be raised. Otherwise, if it is a Windows machine, it won't. In another case, we could raise an alert if the target is known to be vulnerable to the vulnerability actually being exploited by the attack

Moreover, the subsystem is also equipped with an active and passive scanner that continuously looks for any change in the network (e.g., new application, topological changes). The active scanners are performed by tool which scan the network periodically, like nmap (*nmap.org*), while the passive scanners "listens" the network looking for changes, like p0f (*lcamtuf.coredump.cx/p0f.shtml*).

Finally, the subsystem also makes use of several vulnerability analysis tools like Openvas (*www.openvas.org*), to detect any problem in the network, and of monitoring tool that allow to profile the several "entities" in the network.

As in the anomaly detection case, also this subsystem, in case an attack is detected, sends an alert to the MCU.

5 CYBERSENS system evaluation benchmarks

To measure the level of accomplishment of CYBERSENS module, first we have to identify and enumerate its goals and then find performance indicators that allow us to get metrics about the behaviour of the module. Following the functionalities outlined in Section 1, the CYBERSENS goals can be summarised in five main categories:

- Detection and prevention of malicious events: it represents the main objective of the module and its performance must be carefully measured. The detection phase can be seen as a classification problem in which the events generated by the captured network traffic and logs captured must be divided into normal and malicious categories. To measure the quality of the decisions these algorithms take, two of the main performance indicators are the false positive rate and the detection rate. False positive rate indicates how much the algorithm classifies a benign sample as malicious, while the detection rate tells how many malicious samples are properly detected as bad ones. The combination of different samples of these metrics using different configurations allows to draw a Receiver Operating Characteristic (ROC) curve. This kind of graphical plot shows the behaviour of classification algorithms when their discrimination parameters are modified. It will provide a good indicator of how well and how flexible are the detection algorithms developed in the CYBERSENS module. Additional tests can also be performed to analyse how the algorithms behave at guessing the specific category of attack seen in the malicious events
- Privacy preservation and data aggregation: the second goal of the CYBERSENS module is to be able to manage the complexity and size of the data collected from the network and hosts. The combination of aggregation and reduction techniques has been studied. Privacy is a complex parameter to measure as there are not direct tests that allows to see how much private data is leaked during the normal work process of the system. Since encryption mechanisms are going to be enforced during the transport of the data between each element of the architecture, a coverage test is performed. In this type of tests we can evaluate how many nodes are configured to use encryption mechanisms and if they are properly configured (e.g., encryption algorithms, key size, key storage method)
- Traffic generation: as previously mentioned, the platform communications and network traffic must be robust against passive sniffing attacks. To address this issue the platform fake flows are injected at each network so as to make it harder for an attacker to deduce any meaningful information doing statistical analysis of the traffic. The level of protection that introduces this feature is measured doing actual statistical analysis of the traffic and running pattern searching algorithms against captured traces with and without injections inside. Another key issue is how these injections may influence the detection performance as they may introduce non-standard behaviour inside the network that could be detected as anomalous and increase the false positive rate. To evaluate such an impact samples of traffic data with and without injections inside are collected and the same measures are made to judge the detection performance
- Attack redirection to honeynets: when an attack is detected by CYBERSENS module it should be redirected to a honeynet. This objective is focused on the collection of attacker data while interacting inside the honeynet, producing precious information that can feed the detection algorithms with

new malicious behaviour. It is also expected that very few of the normal traffic is redirected to the honeynet as it will produce service outage for legitimate users of the network. This requirement is directly linked with the detection performance indicators. A high false positive rate would probably produce more redirections of benign traffic to a honeynet. With this in mind the type of attacks or the specific threshold that is used to choose a flow as malicious (and therefore a candidate for redirection) is configurable, making it easy to adjust the rate in case of detecting legitimate traffic ending at the honeynet

- Self-protection: as the system itself is expected to be the target of attacks it has been built to resist them and react in the case of a platform compromise. To this aim, CYBERSENS system deploys probes at the actual elements of the platform. Therefore proper responses are intended to protect the platform itself (e.g.,isolating partial segments that may be under attack, containing the spread of the intrusion). On the other side resource overconsumption may also be the source of targeted attacks aimed to disable security measures to bypass monitoring services. Moreover it can lead to a DoS attack if the resource consumption lead to a service outage in the actual services of the critical infrastructure. To measure how this requirement is met, simulated alerts are sent to the MCU to evaluate how the system react to attacks inside the platform and if they are properly isolated

Given these functionalities, the main requirements that the CYBERSENS system should fulfil, to properly protect a critical infrastructure, are:

- Detection rate in the range 70% to 90%, with a false alarm rate lower than 20%
- Ability of processing, almost in real-time, network traffic with a granularity either at the flow level, if direct access to the network devices is guaranteed, or at the aggregate level, if data are exported from the network devices to the monitoring probe
- Ability of redirecting the 70% of anomalous traffic to the honeynet, with only a maximum of 1% of normal traffic also redirected to the honeynet
- Ability of processing more than 1000 events in the case of usage of log records for forensic activities, corresponding to maximum time to perform searches in data recently archived of 15' and of 30' in data archived in 6 months
- Ability of disclosing not more than 1% of private data during aggregation and export

6 Conclusions

One of the main concerns for today's critical infrastructures is protection against cyber attacks. For effective protection, advanced Intrusion Prevention/Detection Systems (IDS/IPS) are paramount. This paper presents CYBERSENS, a novel advanced IDS/IPS system intended for critical infrastructures. Particularly, the general architecture and the main functionalities of the CYBERSENS system, as well as the interactions with other subsystems within the critical infrastructure network are described. Finally, an outline of the evaluation metrics we will use to assess CYBERSENS performance is presented.

ACKNOWLEDGEMENTS

This work was partially supported by SCOUT, a research project supported by the European Commission under its 7th Framework Program (contract-no. 607019). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the SCOUT project or the European Commission.

REFERENCES

Bace, R. G. (2000). *Intrusion detection*. Sams Publishing.

Bray, R., Cid, D., and Hay, A. (2008). *OSSEC host-based intrusion detection guide*. Syngress.

Callegari, C., Di Pietro, A., Giordano, S., Pepe, T., and Procissi, G. (2012). The loglog counting reversible sketch: A distributed architecture for detecting anomalies in backbone networks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 1287–1291.

Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., and Pepe, T. (2010). When randomness improves the anomaly detection performance. In *Proceedings of the International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*.

Callegari, C., Giordano, S., and Pagano, M. (2015). *Network and System Security: 9th International Conference, NSS 2015, New York, NY, USA, November 3-5, 2015, Proceedings*, chapter Enforcing Privacy in Distributed Multi-Domain Network Anomaly Detection, pages 439–446. Springer International Publishing, Cham.

Carlen, P. L. (2013). Traffic flow confidentiality mechanisms and their impact on traffic. In *Military Communications and Information Systems Conference (MCC), 2013*, pages 1–6. IEEE.

Claise, B. (2004). Cisco systems netflow services export version 9.

Cormode, G. and Muthukrishnan, S. (2005). An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58 – 75.

Durand, M. and Flajolet, P. (2003). Loglog counting of large cardinalities. In *In ESA*, pages 605–617.

Enumeration, C. A. P. (2013). Classification (capec). *URL https://capec. mitre. org*.

EUCommission (2004). Critical Infrastructure Protection in the Fight against Terrorism.

Flajolet, P. and Martin, G. N. (1985). Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209.

Schweller, R., Gupta, A., Parsons, E., and Chen, Y. (2004). Reversible sketches for efficient and accurate change detection over network data streams. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 207–212, New York, NY, USA. ACM.

Spitzner, L. (2003). The honeynet project: Trapping the hackers. *IEEE Security & Privacy*, (2):15–23.

Woodring, S. (2001). Port mirroring in channel directors and switches. US Patent App. 10/026,706.

Zhang, F., Zhou, S., Qin, Z., and Liu, J. (2003). Honeypot: a supplemented active defense system for network security. In *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*, pages 231–235. IEEE.