

Gathering Malware Data through High-Interaction Honeypots

(DISCUSSION PAPER)

Angelo Furfaro¹[0000-0003-2537-8918], Francesco Lupia²[0000-0003-0775-6890],
and Domenico Sacca¹[0000-0003-3584-5372]

¹ University of Calabria, Rende(CS), Italy
{angelo.furfaro, domenico.sacca}@unicl.it

² OKT srl, Rende(CS), Italy
francesco.lupia@okt-srl.com

Abstract. The widespread and ever increasing number of services and devices which expose their interfaces to the Internet make the cyberspace a fertile ground for malware activities. Hence there is a strong demand for cybersecurity solutions ensuring their safe operation. Honeypots are networked computer systems purposely designed and crafted to mimic regular services, operating systems and devices with the goal of capturing and storing information about the interactions with attacking entities and we repute them a crucial technology in the study of cyber threats and attacks. We presents the main features of EMPHAsis, a data streaming analytics system based on high-interaction honeypots, which enables the collection and analysis of relevant data about intercepted malware.

Keywords: Honeypots · Cybersecurity · Data collection · data streaming analytics.

1 Introduction

The era of Internet of Things with billions of connected devices has created an ever larger space for cyber attackers to exploit. In particular, there is a widespread amount of automated bots scanning and probing the Internet to search for vulnerable hosts. This has resulted in the need for fast and accurate detection of possible system vulnerabilities and attackers by means of the processing of the high-velocity, high-volume data from various sources to discover anomalies and/or attack patterns as fast as possible to limit the vulnerability of the systems and increase their resilience. Such data, generated in a real-time data stream, need a *big data streaming analysis* [13, 9]: the output must be generated with low-latency and any incoming data must be reflected in the newly generated output within seconds. In other words, big data streaming analytics

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). This volume is published and copyrighted by its editors. SEBD 2020, June 21-24, 2020, Villasimius, Italy.

tools must be able to identify new information, incrementally build models and access whether the new incoming data deviate from model predictions.

Even though many big data streaming analytics tools have been developed in the past few years, their usage in the field of cybersecurity is not immediate and calls for new approaches as pointed out in [5]. The main issue is how to access in real time to valuable data on possible attackers, typically log files and security alerts generated by operating systems and applications across various hosts and systems. A crucial point is to provide computer systems with software tools to identify suspicious event log activity, such as repeated failed login attempts, excessive CPU usage, large data transfers and immediate alert IT security analysts when a possible Indicator of Compromise (*IoC*) is detected.

It should be noted that the techniques and attack methods employed by malwares are typically very simple. In fact, the most used attack method is often brute-forcing login credentials of servers and, consequently, the most heavily attacked services are Telnet, FTP and SSH services. There are many tools used in computer security to catch such malicious actions and one effective solution involves the usage of honeypots. Honeypots [12] are decoy systems which aim at emulating real services on the net in order to detect and attract malicious agents. These emulated services are publicly displayed and their access is made voluntarily simple to facilitate attackers' intrusions. This is done for example by configuring accounts and services with weak credentials. After accessing the service, all the activities of the malicious agents are subject to monitoring and logging and become the object of study by specialists in the sector, in order to be able to reconstruct with extreme precision the behavior of the attacker and thus be able to prepare prevention measures to reduce the risk of future attacks. Furthermore, it is possible to obtain valuable information concerning the behavior of the attacker, the actions carried out on the target system and the types of vulnerabilities exploited to complete the malevolent activity, which can be subsequently shared with the academia and industry researchers.

Here we present the main features of EMPHAsis, a distributed system conceived to support malware detection by acting as a high-interaction honeypot [11]. The system is able to collect and disseminate information about new threats that proliferate every day on the Internet by providing researchers with fresh data that could help them to devise possible countermeasures against malicious traffic. In addition to its detection, logging and monitoring capabilities, EMPHAsis is capable to capture different malware binaries and to exploit external services for analyze them.

2 System Architecture

While malware's actions may vary depending on the targeted system, there are only few points of interest that need to be observed and analyzed: network information, the commands executed, processes created, kernel drivers loaded and the files created or modified by the malicious agent. Each attacker's action is processed by EMPHAsis according to the functional architecture depicted in Figure 1 discussed next. The system operates by constructing dynamically

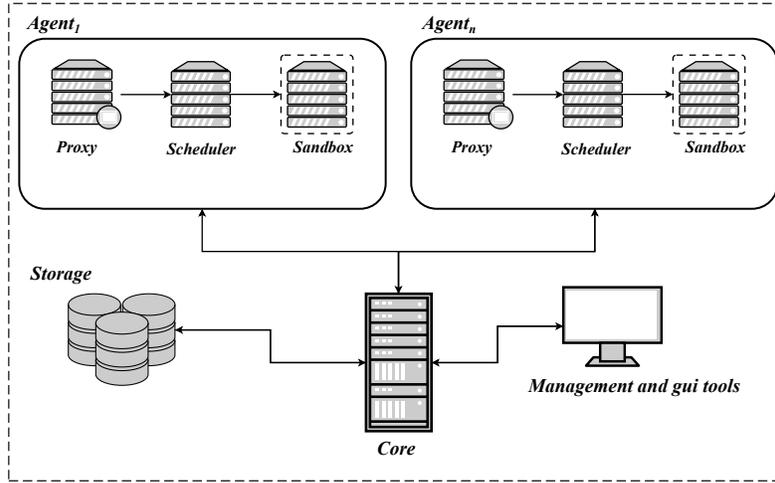


Fig. 1. EMPHAsis architecture

virtual environments (sandboxes) to which all traffic generated by the attacker is redirected out. Each environment is equipped with several specialized probes, whose goal is to monitor in a completely transparent way to the malicious entity, specific critical events that occur in the attack scenario at hand. All the data collected by the probes is eventually stored and indexed for subsequent use (e.g., in order to devise appropriate countermeasures) and possible disclosure.

In more details, the EMPHAsis architecture consists of several modules each of which is in charge of performing a specific task. The **core** is the most important module which plays the role of coordinating the operations between all the other modules of the system. Furthermore, it provides a graphical environment for managing the system, to help real-time monitoring and to visualize the collected data. When a connection attempt reaches the system, the **agent** module checks whether there is a configured service to handle the request. If a suitable service exists, a virtual trap environment is built and executed on the fly and all the attacker's traffic is then redirected out to it. The upper part of Figure 2 shows the main steps performed by the **agent** module in response to an attack. In more detail, as a preliminary step the *proxy* component notifies a new connection arrival to the **core** module while collecting also information about the geographical origin of the attack. Then, it continues by performing payload detection and if a suitable service is found, the request is handled and the *scheduler* component is alerted. The *scheduler* is the main responsible for the virtual trap environment setup. After receiving an alert from the *proxy*, it prepares and starts the *sandbox* and then injects one or more probes directly into the isolated environment. These probes will monitor all the actions performed by the attacker (or malware) inside the sandbox. In particular, in the lower part of Figure 2 is shown an example of a specific case where four probes are in place and used for monitoring networking events, commands executed in the system

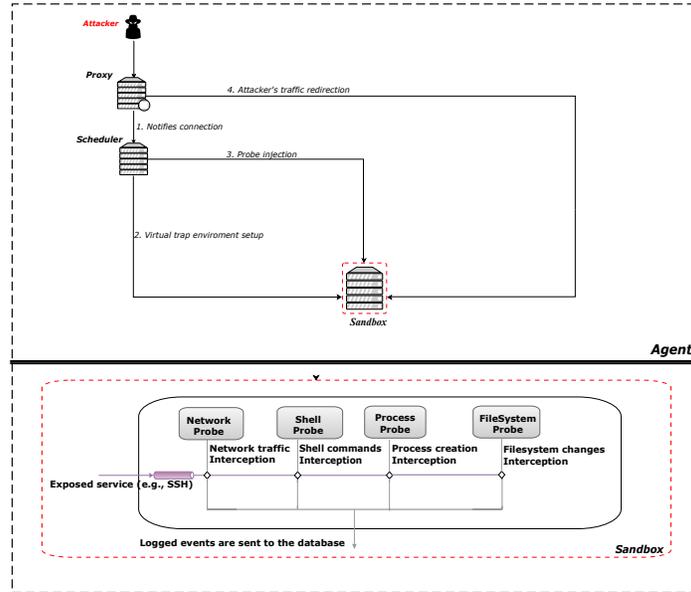


Fig. 2. Agent module

shell, process creations, and changes to the filesystem. It should be noted that the concrete implementation of these probes poses technical challenges. Indeed, one needs to use specific tools and instruments according to the type of operating system used by the emulated guest machine. Our probes support a custom 32-bit Linux kernel and we rely on QEMU-KVM to emulate this platform. Further implementation details can be found in [6]. It is interesting to notice that our implementation of the **agent** exploits a fake interactive session where attacker's traffic is man-in-the-middle proxied. This is achieved by using a custom shell implementation.

The **storage** module comprehends one or more nodes in a redundant configuration whose role is to offer all standard operations for storage and subsequent consultation of the collected data. An attack can produce a significant amount of information, which therefore needs to be properly cataloged and organized for subsequent consultation. To this end, our system exploits appropriate components for the storage and processing of such information, making it suitable for the production of detailed reports and in characterizing the threats detected. Additionally, the above processing can be customized by the user for the specific resources she is most interested in. In particular, an individual storage node uses both a MongoDB NoSQL database [1] and an Elasticsearch database [3]. The MongoDB holds the users' credentials required to access the control and the data analytic dashboard, stores the configuration (including custom kernel images and the probes) which will be downloaded by the **agent** module during the initialization phase of the sandboxes, and stores the captured malware sam-



Fig. 3. Main panel of the EMPHAsis dashboard

ples. The system will then send over all the security events described above to the Elasticsearch database. Such events can be visualized by the built-in dashboard (see section 3) which exploits a Kibana instance for advanced filtering mechanisms.

A key aspect in order to achieve effective results (in terms of number of malicious actions logged and malware instances intercepted) concerns the correct positioning of the honeypots in the infrastructure to be protected. Thus, there is significant motivation for studying the best possible locations where agent modules (and honeypots in general) can be deployed in a network. Interestingly, EMPHAsis agents can be deployed both behind or outside the perimeter of the network, therefore allowing to detect also potential insider threats. In any case, deploying honeypots in a production environment can be problematic since it could expose the network to far greater risks than the threats from which it is intended to be protected. Towards this end, it is essential to study and analyze the way honeypots interact with the rest of the system before the actual deployment. This could be done by using virtual simulation environments, such as the one described in [7] which allow to reproduce multiple operating systems as well as networks in a realistic and controlled way.

3 Collected Data

All of the security events collected through the probes are sent to the storage module and stored in the Elasticsearch database allowing for fast real-time access through the dashboard. More precisely, we construct a single index which is shared by every probe of any agents running in the system. Figure 3 shows the main panel of the dashboard which displays the information about the active sandboxes, under the control of the **core** module, and the number of intercepted attacks. The lower part of the panel reports some statistics about the last five days of operation: the percentage of sandboxes launched, grouped by their type,

#	Timestamp	Protocol	Source IP	Destination IP	Source port	Destination port	Net layer	Event
1	02/21/2020 (00:00:00)	TCP	222.186.175.150	10.0.2.2	22	22	IPv4	2020-02-20 23:59:50 +0100 CET fw-entry auth.info sshd[4279]; Username: root Password: 261979
2	02/21/2020 (00:00:00)	TCP	222.186.175.150	10.0.2.2	60808	22	IPv4	2020-02-20 23:59:42 +0100 CET fw-entry auth.info sshd[4270]; Failed password for root from 222.186.175.150 port 60808 ssh2
3	02/21/2020 (00:00:00)	TCP	218.92.0.211	10.0.2.2	60886	22	IPv4	2020-02-20 23:59:53 +0100 CET fw-entry auth.info sshd[4282]; Username: root Password: sdfkj
4	02/21/2020 (00:00:01)	TCP	222.186.175.150	10.0.2.2	60876	22	IPv4	2020-02-20 23:59:50 +0100 CET fw-entry auth.info sshd[4270]; Failed

Fig. 5. Network log tab



Fig. 6. Filesystem log tab

sorting to the malware identification and classification services offered by Virus-total [14] and those developed by Cythereal [2]. In particular the Cythereal MAGIC API [2] exploits a state-of-art machine-learning based malware analysis system which is able to identify the belonging family of a malware even in presence of sophisticated code obfuscation techniques.

4 Conclusions and Future works

We presented the main functionalities of EMPHAsis and its practical applicability for malware capturing, analysis and prevention. EMPHAsis exploits high-interaction honeypots and has a modular and extensible architecture making it effective and suitable for various practical deployment scenarios. Currently EMPHAsis features the emulation of Linux based systems on top of which the great

majority of Internet exposed services are based on. Moreover, it can be easily extended to support other unix-based systems. As a future work, we plan to support also computer systems based on Microsoft Windows or on Apple macOS. Another important direction of improvement is the integration of EMPHAsis with more advanced data analysis tools specifically tailored to cybersecurity events, i.e. the so called *security information and event management* systems (SIEM), like IBM QRadar [8] and ElasticSIEM [10], which has been recently developed on top of Elasticsearch. We also plan to extend the interoperability of EMPHAsis with other malware analysis tools, like for example cuckoo [4] sandbox, and to integrate in it virtual environment technologies for building more realistic and complex decoy environments [7], e.g. honeynets, and thus increasing the effectiveness in capturing more sophisticated malware.

Acknowledgments This research has been supported by the ISCOM - Italian Ministry of Economic Development under agreement EMPHAsis (CUP B51G17000300006).

References

1. Cuckoo. <https://cuckoosandbox.org/>.
2. Cythereal, changing the rules of cyber engagement. <http://www.cythereal.com>.
3. Elasticsearch. <https://www.elastic.co/products/elasticsearch>.
4. MongoDB. <https://www.mongodb.com>.
5. Pelin Angin, Bharat K. Bhargava, and Rohit Ranchal. Big data analytics for cyber security. *Security and Communication Networks*, 2019:4109836:1–4109836:2, 2019.
6. Michele Bombardieri, Salvatore Castanò, Fabrizio Curcio, Angelo Furfaro, and Helen D. Karatza. Honeypot-powered malware reverse engineering. In *2016 IEEE International Conference on Cloud Engineering Workshop, IC2E Workshops, Berlin, Germany, April 4-8, 2016*, pages 65–69, 2016.
7. Angelo Furfaro, Luciano Argento, Andrea Parise, and Antonio Piccolo. Using virtual environments for the assessment of cybersecurity issues in iot scenarios. *Simulation Modelling Practice and Theory*, 73:43–54, 2017.
8. IBM. QRadar. <https://www.ibm.com/security/security-intelligence/qradar>.
9. Taiwo Kolajo, Olawande Daramola, and Ayodele Adebiyi. Big data stream analysis: a systematic literature review. *J. Big Data*, 6:47, 2019.
10. Mike Paquette. Introducing Elastic SIEM. <https://www.elastic.co/blog/introducing-elastic-siem>, 2019.
11. Lance Spitzner. The honeynet project: Trapping the hackers. *IEEE Security & Privacy*, 1(2):15–23, 2003.
12. Lance Spitzner. Honeypots: Catching the insider threat. In *19th Annual Computer Security Applications Conference (ACSAC 2003), 8-12 December 2003, Las Vegas, NV, USA*, pages 170–179, 2003.
13. Nicoleta Tantalaki, Stavros Souravlas, Manos Roumeliotis, and Stefanos Katsavounis. Linear scheduling of big data streams on multiprocessor sets in the cloud. In Payam M. Barnaghi, Georg Gottlob, Yannis Manolopoulos, Theodoros Tzouramanis, and Athena Vakali, editors, *2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019, Thessaloniki, Greece, October 14-17, 2019*, pages 107–115. ACM, 2019.
14. Virustotal. How it works. <https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works>.