# Protocol Deployment for Employing Honeypot-as-a-Service

Alexandros Kostopoulos[1](✉), Ioannis P. Chochliouros[1],
Constantinos Patsakis[2], Miltos Anastasiadis[3],
and Alessandro Guarino[4]

[1] Hellenic Telecommunications Organization (OTE) S.A., 99 Kifissias Avenue,
15124 Maroussi, Athens, Greece
{alexkosto,ichochliouros}@oteresearch.gr
[2] University of Piraeus, Piraeus, Greece
kpatsak@gmail.com
[3] Motivian Eood, Sofia, Bulgaria
manastasiadis@motivian.com
[4] StagCyber, Cavazzale, Italy
a.guarino@stagcyber.eu

**Abstract.** The YAKSHA project aims at reinforcing EU-ASEAN cooperation and building partnerships in cybersecurity domain by developing a solution tailored to specific national needs leveraging EU know-how and local knowledge. YAKSHA enhances cybersecurity readiness levels for its end-users, helps better prevent cyber-attacks, reduces cyber-risks and better governs the whole cybersecurity process. The EU-ASEAN cooperation also helps to mitigate some of the weaknesses identified in the cybersecurity ecosystem. In this paper, we consider the protocol deployment process for employing honeypot-as-a-service, with focus on the Internet of Things (IoT) use case.

**Keywords:** ASEAN-EU cooperation · Cybersecurity · Data analytics · Honeypots-as-a-service · ICT · Malware · Protocol deployment

## 1 Introduction

Informatisation is widely recognized as a “key” enabling factor for developing economies and society. ASEAN (Association of Southeast Asian Nations), and in particular low- and middle-income countries in this region, have long been subject to several cybersecurity issues and are exposed to specific risks, ranging from data breaches to intentional intrusions by adversaries. Security is a common element buttressing all other ICT technologies [1, 2], and without which ICT can be considered as much of a risk than an opportunity for organisations, businesses and governments [3].

YAKSHA [4] aims at reinforcing EU (European Union) - ASEAN cooperation and building partnerships in the cybersecurity domain by developing a solution tailored to specific user and national needs, leveraging EU know-how and local expertise. YAKSHA develops and introduces the innovative concept of “honeypot-as-a-service” which greatly enhances the process of gathering threat intelligence [5, 6]. Many

Published by Springer Nature Switzerland AG 2020
I. Maglogiannis et al. (Eds.): AIAI 2020 Workshops, IFIP AICT 585, pp. 105–115, 2020.
https://doi.org/10.1007/978-3-030-49190-1_10

companies and organisations desire to test the systems they deploy in terms of security, however they are not always able to do it “appropriately”. Moreover, since the developments in this field are continuous, many end-users would like to be informed about zero-day exploits for their systems. YAKSHA enables organisations to handle this challenge in an automated way, allowing different modalities of processing and features, whether this is access to sample of other nodes or more advanced algorithms.

YAKSHA develops innovative methods for malware detection, collection and analysis [7, 8], as well as designs a specialized ontology to be used for long-term storage and analysis of the information, and deploys standard information formats and interfaces to facilitate interoperability. The YAKSHA software solution is validated in real-world pilot projects in both regions, focusing on Vietnam, Malaysia and Greece. The test cases are represented by complex end-users with articulated cybersecurity risks and allows the consortium to gather feedback to be used later to bring the solution to the market. YAKSHA develops a comprehensive business plan for the transition of the solution and complementary services to Technology Readiness Level 9 (TRL9), leveraging the very strong ASEAN partners that are members of the consortium - including governmental organisations and leading end-users communities and associations with regional scope. As part of this effort, YAKSHA builds a whole ecosystem of partners around its solutions. This contributes to enhancing cybersecurity skills in Europe and creating new positions for cybersecurity specialists in ASEAN. Moreover, the direct access to the all-important ASEAN market provided to partners will positively impact the competitiveness of European security industry.

The paper is organised as follows: Sect. 2 presents the overall architecture of the YAKSHA platform. Section 3 provides the pilots’ deployment protocol for the YAKSHA data flow compliance. Section 4 focuses on the protocol deployment for the Internet of Things (IoT) use case. In Sect. 5 we conclude our remarks and present our future work.

## 2 Architecture

YAKSHA is a distributed system which allows the automated deployment of honeypots, data collection and analysis as well as reporting and information sharing with affiliated YAKSHA installations. Honeypots despite the fact that they provide a very good insight on the actual attacks that can be launched against a system [9, 10], they are rather difficult to be properly deployed and in many occasions, require a lot of effort and dedicated personnel to integrate them [11]. Moreover, most honeypots are focusing specific systems, with Windows and Linux dominating the field. As a result, many companies and organisations are discouraged to deploy their custom honeypots to monitor possible attacks on their systems [12]. Furthermore, it is quite often that these systems are not properly configured to monitor the attacks properly, or perform maintenance procedures for the continuation of these efforts. YAKSHA aims to advance current state-of-the-art and practice, by providing an easy mechanism for the automated deployment and management of honeypots, so that organisations and companies can easily create custom honeypots with the integrated sensors properly configured and sending all the collected information to a central repository that they manage [13].

Currently, most honeypots and sandboxes might be able to collect a lot of valuable information about an attack [14, 15], however, this information is reported in the form of logs, and require a dedicated analyst to go through them and understand the attack pattern and impact [16]. The latter is something that cannot be expected in many "actors" (such as companies and organisations), as in many occasions they would need a simple report that states that their system is vulnerable to, for example, a denial of service (DoS) or privilege escalation attack, where the attacker was able to perform a list of actions, using the collected piece of code.

In this regard, YAKSHA advances current state-of-the-art by extracting actual knowledge from the log files in a human readable format, so that the attack analysis can be simplified and partially automated. In addition, YAKSHA makes honeypots more stealth, and collects even more important information, when this is possible. Finally, YAKSHA advances existing knowledge by providing machine learning tools and Artificial Intelligence (AI) algorithms able to detect malware more accurately, correlates the information with other samples, and extracts attack vectors and patterns.

The modular and distributed nature of YAKSHA allows it to cater for both opportunistic and continuous sample collection, and selective information sharing with other entities when deemed necessary. YAKSHA, therefore, enables organisations, companies and government agencies to upload custom honeypots that "meet" their own specifications, monitor attacks in real time and analyse them. However, since some of these honeypots may expose corporate or organisation specific vulnerabilities, each YAKSHA node may specify policies for information sharing per honeypot, attack pattern, affiliated nodes or even user roles of users in affiliated nodes. To this end, initially each YAKSHA installation is an independent instantiation of the system which has its own users (e.g.: admins, auditors, analyzers, backup managers, integrators, etc.), its own honeypots, and performs its processing locally. Clearly, a YAKSHA node, due to processing requirements would consist of more than one computer, but in what follows is considered as a single system.

The conceptual architecture of a YAKSHA node is illustrated in Fig. 1. On top, the installed honeypots which are exposed to the Internet so that attackers will try to penetrate them. YAKSHA apart from typical Linux and Windows honeypots aims to provide hooks for IoT devices, as well as for Android and for SCADA (Supervisory Control and Data Acquisition) systems [17]. YAKSHA takes into consideration tools to provide a sandbox for automating the analysis of the collected samples [18], nonetheless, YAKSHA strives to extend them, not only in terms of integrating them in a honeypot, but hooking other operating systems, extracting more information, but more importantly, processing the extracted information.

In the following, we briefly discuss the various distinct Engines/Modules included in the original YAKSHA architecture.

Firstly, the *Maintenance and Integration Engine* which allows the configuration of a new honeypot, uploading and exposing it to the Internet and data wipe. Therefore, this engine enables the node admin to deploy a honeypot from scratch, share it and, if deemed necessary, drop back to initial settings to collect more data. The critical point of this module is to automate the procedure of creating hooks to the system for a range of operating systems, so that end-users can assess the risk to which several of their systems are exposed to. Apart from making the procedure seamless and automatic, this module must make the procedure transparent, so that the hooks cannot be traced by an adversary who penetrated the system. The integration for systems beyond desktop and server environments is not easy nor straightforward, as core and undocumented changes have to be made to the underlying operating system. The architecture of a YAKSHA node is illustrated in Fig. 2, as appearing below.

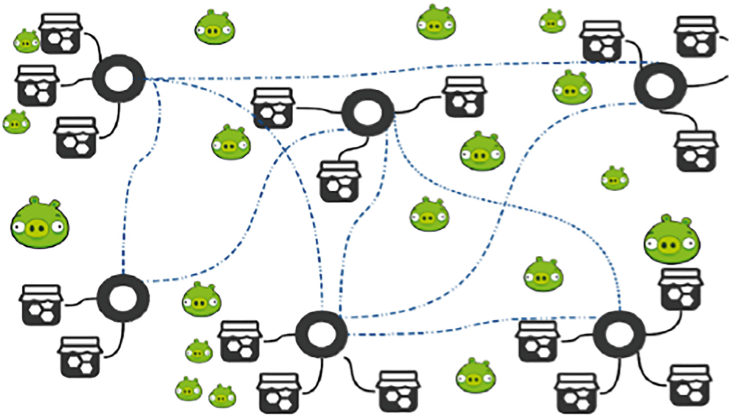

**Fig. 1.** YAKSHA overall architecture.

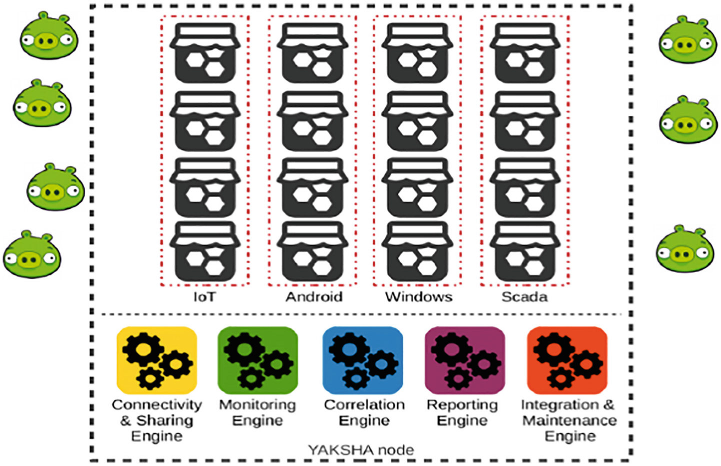

**Fig. 2.** The architecture of a YAKSHA node.

The *Monitoring Engine* performs sanity checks to determine whether the honeypot is properly working and records all changes in memory, processes and filesystem as well as network connections to detect anomalies that an adversary performs during an attack. The core goal of this module is to monitor everything that happens in the system so that:

- Its presence cannot be traced by the attacker. If the attacker understands that s/he is being monitored, then the honeypot loses its value, as the attacker is expected to quit.
- Collect all the information so that the actions can be replicated.
- Maintain all the actions in a sandboxed environment so that no malicious actions can escape the honeypot and infect the rest of the system.

The *Correlation Engine*, on receiving the data from the monitoring engine, tries to find how significant is the penetration and propagation of the sample, that is: what privileges did the malware manage to gain, did it manage to add users, did it manage to write to protected folders or shut down the system. Then it tries to correlate the attack patterns with input from older samples. For instance, it finds whether changes in the registry, filesystem, system calls etc., have already been made by another sample. The Correlation Engine is the most crucial part of YAKSHA as it is the selling point of the platform since it automates the evaluation of the risks a system is exposed to. Many systems may enable the deployment of a honeypot, however, without taking into consideration the amount of effort for installation and customisation, at the end of the day, one has to go through each attack independently and evaluate it. YAKSHA does not make this analysis obsolete by making it completely automated; instead, it provides a set of filters which significantly reduces the amount of manual work and ranks it according to the impact on the system. In addition, attacks are clustered to extract further attack patterns in terms of tools, methods and results.

The *Reporting Engine* is the module charged with presenting the information in a readable form to the users. On one hand, it is in charge of issuing alerts and aggregating information for specific documents targeted to technical personnel; on the other hand it has the task of providing management with meaningful input on the organisation's cybersecurity current posture and risk levels. For the non-technical audience inside the organisation, the Reporting Engine provides dashboard presenting in real-time the status of the node, the level of risk for each asset, type of attacks, threat vectors, as well as an estimation of the possible impacts. The latter is presented both in financial and operational terms. The dashboard also presents in real-time the areas where the risk of attack is higher and propose controls to apply to mitigate them; for example, patches to apply, firewall and IDS (Intrusion Detection System) configurations [19] to be updated, etc. In short, YAKSHA's Reporting Engine supplies the decision-makers with a continuous risk assessment tool, as well as the IT personnel with detailed technical reports while acting as a recommender system, when this is possible.

Finally, the role of *Connectivity and Sharing Engine* is to allow the exchange of information with other YAKSHA nodes. Each node has its own users, with their respective roles and can connect with other nodes to exchange malware samples. Each node can select which samples to share and with which nodes, for instance node A has some SCADA honeypots [20] that wants to share only with node B and not with node C. These policies can be further tuned so that, *for example*, only auditors from node B can access this information.

## 3 Pilots' Deployment Protocol for YAKSHA Data Flow Compliance

This section describes the identified baseline of activities that form part of the pilots' deployment protocol. These activities ensure that the YAKSHA platform deployed in the pilots' premises is compliant with the dataflow principles and legal grounds for data processing defined for proper operation and results of the platform [21]. The protocol activities for data flow compliance are essential and required to take place for all pilots.

The protocol's activities for organisational data flow compliance include:

- *Node installation and administrative access flow:* A pilot organisation appoints YAKSHA administrators (persons), assisted by the YAKSHA technical team to install and administrate a YAKSHA node for the given organisation. A YAKSHA node instantiation with all administrative data access/data flow is properly set up and configured for the given organisation.
- *End-users access flow:* Relevant employees of the pilot organisation (security officers, managers, executive positions) are given access to the YAKSHA node to receive cybersecurity repots and alerts on the status of the cybersecurity analyses. All YAKSHA end-users' access is properly set up according to roles established.

The protocol's activities for technical data flow compliance include:

- *Honeypot-to-Node Affiliation:* A pilot organisation technically affiliates (sets, configures and deploys) all needed honeypots to an organisation's own and instantiated (operational) YAKSHA node. The pilot organisation ensures each honeypot is affiliated to only one operational YAKSHA node of that organisation. For the sake of separation of administration and data flow management, each honeypot reports results to one affiliated node only while the node policy may allow sharing reports from honeypots with other affiliated nodes (within or outside the organisation).
- *Honeypot-to-Node Data Flow:* The pilot organisation ensures adequate level of the Quality of Service (QoS) for the network connection between a honeypot and its affiliated node. The data flow between a honeypot and its node has to be reliable and timely (e.g., soft real-time restriction is desirable) to ensure adequate analysis and alerts triggering when an attack is taking place. This protocol activity must entail pilot organisation guarantee reliable honeypot-to-node data flow. Any misconfiguration or connection unavailability (e.g., due to low QoS) may degrade the results of YAKSHA analytics and the overall results/evaluation of the pilot. This protocol activity defines reliable data flow not only between a honeypot and the node, but also between a honeypot and its associated (external) tools for malware analysis, such as the Cuckoo Sandbox[1], and also between those tools and the affiliated YAKSHA node.
- *Node-to-Node Data Flow:* A pilot organisation affiliates with other pilot organisations to automatically exchange collected malware data. To do so, the pilot organisation, upon organisation-level decision making, sets and configures a data

[1] More details can be found at: https://cuckoosandbox.org/.

sharing policy of the YAKSHA node of that organisation to enable such sharing with other organisations' nodes. Importantly, the pilot organisation must ensure no business-sensitive data is shared with other organisations.

- *Data protection compliance:* A pilot organisation guarantees all technical means of data protection are properly enabled -or set- as part of the YAKHSA platform solution, both for the data flows from the honeypots to the affiliated nodes and for the data flows from a YAKSHA node to other YAKSHA nodes affiliated for such data sharing. Any improper security settings -or no security means for those data flows- imply incorrect or distrusted analysis or reports by the YAKSHA node. For instance, data from honeypots to an affiliated node may be tampered in transit so that an attacker may influence on the analysis of the platform. The pilot organisation ensures all personally identifiable data from honeypot collection, such as IP addresses, must be treated according to the regulations or laws of each country the pilot takes place. Particularly, legal ground compliance must be ensured for data collection, retention and sharing as discussed in [22].

## 4 Deployment Protocol for the IoT Use Case

The goal of the smart home IoT use case is to use a YAKSHA node within a pre-commercial environment (infrastructure and settings) provided by a telecom operator (i.e., the Hellenic Telecommunications Organization S.A. (OTE)) to collect real data of potential attacks against the smart home IoT platform (pre-commercial) product. YAKSHA analytics capability is used to raise awareness and provide decision support in strengthening the cybersecurity posture of the product.

Using YAKSHA in a pre-commercial environment makes OTE aware of potential attacks in the wild against OTE's products and services. OTE has developed, completely "in-house", an end-to-end (E2E) platform/solution for building energy monitoring and management. The solution is based entirely on open source technologies and it is modular by design. As such, it is vendor- and technology-agnostic, capable of integrating numerous and heterogeneous sensors, modules and devices to a common ecosystem.

The IoT testbed layout is depicted in Fig. 3 and includes the following parts:

- A wide range of end-devices/sensors, such as air-quality, temperature, humidity, pressure, activity, luminance, fire and for power/energy communicate with the backend (cloud) infrastructure over a wide range of short/long range technologies (i.e.: Ethernet, WiFi, z-wave[2] BLE (Bluetooth Low-Energy), LoRaWAN (Long-Range Wide Area Network), NB-IoT (Narrow-Band IoT)).
- IoT hubs/gateways (local and remote-based on LoRaWAN) for facility automation and energy management/control (based on events/rules) supporting multiple HAN (Home Area Network)/BAN (Body-Area Network)/LAN (Local Area Network)/

[2] More details can be found at: https://www.z-wave.com/.

WAN (Wide Area Network) technologies/interfaces; over 150 Techs/protocols are currently supported.

- A (common) backend infrastructure (including storage, monitoring/data visualization, command exchange, etc.).

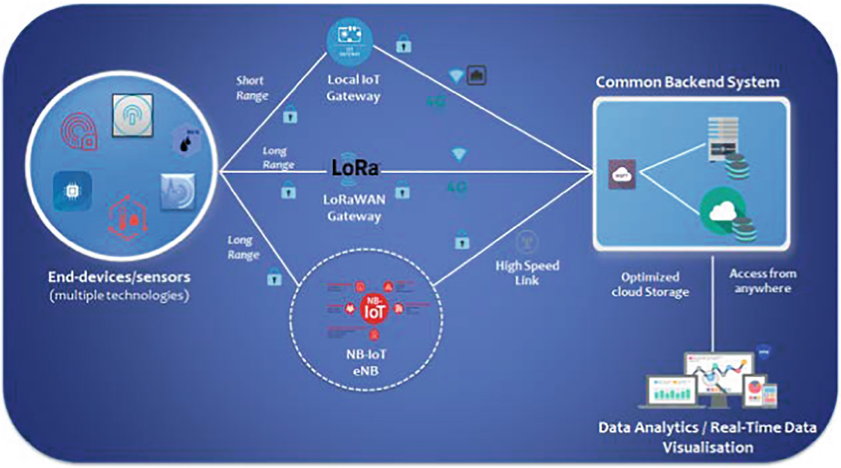

**Fig. 3.** OTE's IoT testbed layout.

Monitoring data are sent to a common backend system with optimised cloud storage via a gateway. The gateway could be either a local IoT (e.g.: UP Board[3] Raspberry Pi), a LoRaWAN, or an NB (NodeB)/eNB (evolved NodeB) IoT gateway.

The IoT service is accessible via any end-device (e.g.: phone, tablet, laptop), from any network. Two user interfaces are available. The first interface is via a mobile application installed in end-users' devices, which enables users to monitor the measurements of the sensors, or manage their devices. The second interface communicates with web server via http requests, which also enables users to define specific automations for their devices.

Next, the identified baseline of activities that form part of the OTE's pilot deployment protocol are described. The protocol's activities for technical data flow compliance related to the *development of the testing environment* include:

- *Storage and computational resources' definition:* The required storage and computational resources were defined in order to develop a testing environment to host the IoT service.
- *Virtual Machine Creation for hosting MQTT in the YAKSHA testing environment:* A virtual machine is created within OTE's cloud environment in order to host the Message Queuing Telemetry Transport (MQTT) broker. MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol, which is designed for

[3] More details can be found at: https://up-board.org/.

connections with remote locations where a "small code footprint" is required or the network bandwidth is limited.

- *Virtual Machine Creation for hosting database in the YAKSHA testing environment:* A virtual machine (VM) is created within OTE's cloud environment in order to host the database, where all the data sent by the sensors are to be stored.
- *Virtual Machine Creation for hosting data visualization services in the YAKSHA testing environment:* A VM is created within OTE's cloud environment in order to host services related to data analytics and real-time data visualisation, as well as enable users to create customised figures.
- *Network configuration support for the YAKSHA testing environment:* A separate VLAN is configured for testing purposes. An IP (Internet Protocol) address is assigned to each virtual machine, gateway and sensor device in order to enable communication among each IoT platform component.

The protocol's activities for technical data flow compliance related to *data transmission* include:

- *Gateway configuration to enable connectivity with the back-end system:* The gateway is configured in order to support a plethora of WAN wireless/wired connectivity options (e.g.: Ethernet, direct VGW-to-VGW (virtual private gateway) communications, 2G/3G/4G/4G+, RF@ 433/868/2400 MHz).
- *Gateway configuration to enable connectivity with the sensors/IoT devices:* The gateway is configured in order to be hardware agnostic (vendor independent) and protocol/technology agnostic. It also supports a plethora of HAN/BAN/LAN communication technologies/protocols (e.g.: modbus, z-wave, KNX[4], BACnet/IP[5], BLE, WiFi, unlicensed bands such as RF@433/868 MHz, etc.).
- *Sensors/IoT devices:* A subset of the available sensors/IoT devices is selected in order to be used for the YAKSHA pilot (e.g.: air-quality, temperature, humidity, pressure, soil moisture, PIR/activity, luminance, distance, fire, rain, sound, GPS, cameras, gyroscopes, accelerometers, etc.).
- *Data collection and transmission:* Data from a wide range of energy and non-energy related measurements (such as weather data either from a local weather station or from "external" weather operators, temperature, humidity and barometer data from sensors, fire/CO2/water/activity related data from sensors, etc.) are collected, processed, controlled, stored, and pushed.

The protocol's activities for technical data flow compliance related to the *end-users* include:

- *End-user interaction with the IoT platform:* Users are enabled to connect to the IoT service, as well as receive push notifications (and/or e-mail) to their device (iOS and/or Android, wearables) on specific user/system related events (e.g.: "dryer is done", turn device on upon activity detection, alarm), and enable voice commands.

[4] https://www.knx.org/knx-en/for-professionals/What-is-KNX/A-brief-introduction/index.php.

[5] More details can be found at: http://www.bacnet.org/Tutorial/BACnetIP/index.html.

## 5 Conclusion

Information systems, regardless of whether they are mobile or not, have penetrated our everyday lives, automating many procedures but also creating huge dependencies. Moreover, modern information systems are highly heterogeneous and due to networking, especially connection to the Internet, they are exposed to far too many risks [23]. Corporations and organisations need to know these risks and assess them, as in many occasions they may constitute the core part of their lifecycle or because they need these systems to operate. Apart from several measures, like auditing, penetration tests etc., security analysts need to know how adversaries try to hack their systems and study their behavior, knowledge, etc. Cybersecurity capabilities are essential for achieving individual organisations' and societies' goals, even more so in the globalized and connected world.

This vital knowledge demands a lot of manual work to deploy such a system, but most importantly, a lot of knowledge and analysis to extract the attack patterns and determine their impact [24–26]. YAKSHA aims to automate a big part of this procedure and distribute this information to peers. For many reasons, most companies and organisations are not able to create such analytics for their systems, so YAKSHA can become the first such system to automatically provide such features. Therefore, the technological experience and knowhow of the EU is "matched" with the wide deployments of ASEAN countries to develop and field test the solution.

In addition, YAKSHA intends to provide an automated framework for deploying honeypots and correlating the collected information. The essential goal is to enable end-users, whether they are governments, organisations or companies, to easily setup customised honeypots, which will allow them to understand how their systems are being attacked on the wild, in an autonomous way.

**Acknowledgments.** The paper has been based on the context of the *"YAKSHA" ("Cybersecurity Awareness and Knowledge Systemic High-level Application")* Project, funded by the EC under the Grant Agreement (GA) No. 780498.

## References

1. Symantec: 2019 Internet Security Threat Report (ISTR), vol. 24. Symantec Corporation (2019)
2. Trustwave: 2019 Trustwave Global Security Report. Trustwave (2019)
3. European Union Agency for Network and Information Security: Threat Landscape Report 2018: 15 Top Cyberthreats and Trends. ENISA (2018). https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2018
4. YAKSHA (Cybersecurity Awareness and Knowledge Systemic High-level Application) H2020 Project, GA No. 780498. http://project-yaksha.eu
5. Spitzner, L.: Honeypots: Tracking Hackers. Addison-Wesley Reading, Boston (2003)
6. Balamurugan, M., Poornima, S.C.B.: Honeypot as a service in cloud. In: Proceedings of the 9th International Conference on Web Services Computing (ICWSC) 2011, pp. 39–43. IEEE (2011)

7. Mokube, I., Adams, M.: Honeypots: concepts, approaches, and challenges. In: Proceedings of the 45th ACM Southeast Conference (ACM-SE), pp. 321–326. ACM (2007)
8. Gandotra, E., Bansal, D., Sofat, S.: Malware analysis and classification: a survey. J. Inf. Secur. **5**(02), 56–64 (2014)
9. Kumar Jain, Y., Singh, S.: Honeypot based secure network system. Int. J. Comput. Sci. Eng. **3**(2), 612–620 (2011)
10. McGrew, R.: Experiences with honeypot systems: development, deployment, and analysis. In: Proceedings of the 39th Hawaii International Conference on Systems Science (HICSS 2006), vol. 9, p. 220a. IEEE (2006)
11. Zhang, F., Zhou, S., Qin, Z., Liu, J.: Honeypot: a supplemented active defense system for network security. In: Parallel & Distributed Computing Applications and Technologies, pp. 231–235 (2003)
12. Parker, M.: Why honeypot technology is no longer effective, CSO (2015). https://www.cso.com.au/article/576966/why-honeypot-technology-no-longer-effective/
13. Chin, W.Y., Markatos, E.P., Antonatos, S., Ioannidis, S., et al.: HoneyLab: large-scale honeypot deployment and resource sharing. In: Proceedings of the 3rd International Conference on Network and System Security (NSS), pp. 381–388. IEEE (2009)
14. Bringer, M.L., et al.: A survey: recent advances and future trends in honeypot research. Int. J. Comput. Netw. Inf. Secur. **10**, 63–75 (2012)
15. Mitchell, A.: An intelligent honeypot. Thesis, Cork Institute of Technology, May 2018
16. Holz, T.: Learning more about attack patterns with honeypots. In: Proceedings of Sicherheit 2006, pp. 30–41. Informatik e.v. (GI) (2006)
17. Kovaliuk, D.O., Huza, K.M., Kovaliuk, O.O.: Development of SCADA system based on web technologies. Int. J. Inf. Eng. Electron. Bus. **10**(2), 25–32 (2018)
18. Sun, B., Fujino, A., Mori, T., Takahashi, T., Inoue, D.: Automatically generating malware analysis reports using sandbox logs. IEICE Trans. Inf. Syst. **E101D**(11), 2622–2632 (2018)
19. Liao, H.-J., et al.: Intrusion detection system: a comprehensive review. J. Netw. Comput. Appl. **36**(1), 16–24 (2013)
20. Jicha, A., Patton, M., Chen, H.: SCADA honeypots: an in-depth analysis of Conpot. In: Proceedings of the IEE 19th International Conference on Intelligent and Security Informatics (ISI), pp. 196–198. IEEE (2016)
21. YAKSHA project: Deliverable 2.1: Data Collection Methodology, June 2018
22. Commission of the European Communities: Horizon 2020 Work Programme 2016–2017. http://ec.europa.eu/research/participants/data/ref/h2020/wp/2016_2017/main/h2020-wp1617-intro_en.pdf
23. European Union Agency for Network and Information Security: Threat Taxonomy. ENISA (2016). https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/threat-taxonomy/at_download/file
24. Mairh, A., Barik, D., Verma, K., Jena, D.: Honeypot in network security: a survey. In: Proceedings of the 2011 International Conference on Communication, Computing & Security (ICCCS), pp. 600–605. Elsevier (2011)
25. Shamsi, J.A., Zeadally, S., Sheikh, F., Flowers, A.: Attribution in cyberspace: techniques and legal implications. Secur. Commun. Netw. **9**(15), 2886–2900 (2016)
26. Egele, M., Scholte, T., Kirda, E., Kruegel, C.: A survey on automated dynamic malware-analysis techniques and tools. ACM Comput. Surv. (CSUR) **44**(2), 6, 1–42 (2012)