Using AI to face covert attacks in IoT and softwarized scenarios: challenges and opportunities

Angelica Liguori[1,2,*], Simone Mungari[1,2], Marco Zuppelli[4], Carmela Comito[1], Enrico Cambiaso[3], Matteo Repetto[4], Massimo Guarascio[1], Luca Caviglione[4] and Giuseppe Manco[1]

[1] *Institute for High Performance Computing and Networking, via P. Bucci 8-9/C, Rende, 87036, Italy*
[2] *University of Calabria, via P. Bucci, Rende, 87036, Italy*
[3] *Institute of Electronics, Computer and Telecommunication Engineering, Via de Marini 6, Genova, 16149, Italy*
[4] *Institute for Applied Mathematics and Information Technologies, Via de Marini 6, Genova, 16149, Italy*

Abstract

Recently, the number of attacks aiming at breaching networked and softwarized environments has been growing exponentially. In particular, information hiding methods and covert attacks have been proven to be able to elude traditional detection systems and exfiltrate sensitive data without producing visible network flows or data exchanges. In this context, Artificial Intelligence techniques can play a key role in detecting these new emerging attacks, owing to their capability of quickly processing huge amounts of data without the necessity of expert intervention. In this work, we discuss the main challenges to face covert attacks in IoT and softwarized environments and we describe some preliminary results obtained by adopting Deep Learning architectures.

Keywords

Stealthy Malware, Stegomalware, Container Security, Evolving Threats

1. Introduction

The recent surge in the use of Artificial Intelligence (AI) is also supported by its adoption to face the growing number of cyber threats [1]. In fact, in recent years, attacks intensified both in terms of volume and complexity, for instance, by using techniques to elude detection or to conceal traffic flows exchanged with a remote controller [2]. Moreover, advanced persistent threats demonstrated their ability to bypass many security perimeters, also due to the use of multi-stage loading architectures or techniques to conceal attack routines. To this aim, AI demonstrated to represent an effective tool for the detection, reverse engineering, and forensics operations by offering valuable support during malware analysis operations [3].

However, an emerging aspect concerns the use of information hiding techniques to implement network covert channels. In this case, the attacker aims at eluding detection by not triggering classical defense systems based on traffic anomalies or exfiltrating sensitive data without producing visible flows. Owing to the effectiveness of the approach, threat actors are increasingly exploring new "carriers", i.e., containers able to conceal secret and malicious data. In this vein, a very recent effort is devoted to understanding the feasibility of using the AI itself as a carrier for malicious data; see, e.g., [4] for a discussion on how information can be concealed in neurons or model parameters. To make things more complex, the increasing softwarization of networks and services accounts for an almost boundless attack surface that can be exploited to make future malware difficult to tame.

From this perspective, our paper showcases the most recent advances and research questions on the use of AI to mitigate malware leveraging covert communications or implementing (hidden) leakage attempts in containerized architectures. Specifically, it discusses the use of federated learning to efficiently deploy AI-based countermeasures in ubiquitous IoT scenarios, as well as the challenges characterizing micro-service architectures built around container technologies. In addition, to offer a comprehensive discussion, we outline opportunities offered by graph generation to devise new effective solutions for detecting evolving threats.

Summing up, the contribution of this work is to shed

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy

[*] Corresponding author.

angelica.liguori@dimes.unical.it (A. Liguori); simone.mungari@unical.it (S. Mungari); marco.zuppelli@ge.imati.cnr.it (M. Zuppelli); carmela.comito@icar.cnr.it (C. Comito); enrico.cambiaso@ge.ieiit.cnr.it (E. Cambiaso); matteo.repetto@ge.imati.cnr.it (M. Repetto); massimo.guarascio@icar.cnr.it (M. Guarascio); luca.caviglione@ge.imati.cnr.it (L. Caviglione); giuseppe.manco@icar.cnr.it (G. Manco)
0000-0001-9402-7375 (A. Liguori); 0000-0002-0961-4151 (S. Mungari); 0000-0001-6932-3199 (M. Zuppelli); 0000-0001-9116-4323 (C. Comito); 0000-0002-6932-1975 (E. Cambiaso); 0000-0001-8478-2633 (M. Repetto); 0000-0001-7711-9833 (M. Guarascio); 0000-0001-6466-3354 (L. Caviglione); 0000-0001-9672-3833 (G. Manco)

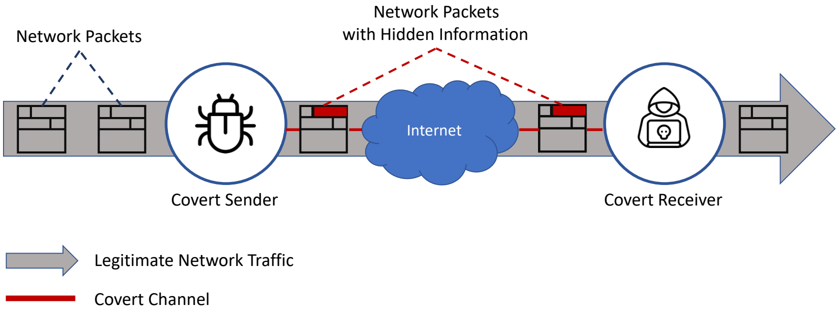

Figure 1: Reference scenario for network covert channels.

new light on the use of AI to face the emerging threat endowed with covert attacks, especially when targeting realistic scenarios based on IoT or container technologies. Another contribution concerns the investigation of "perspective" challenges given by the use of AI-based frameworks.

The rest of the paper is structured as follows. Section 2 deals with AI to mitigate threats using covert channels in IoT ecosystems, whereas Section 3 discusses opportunities for improving the security of containerized environments. Section 4 showcases opportunities arising from graph generation, and 5 presents challenges and opportunities of using AI for cybersecurity-related tasks. Lastly, Section 6 concludes the paper.

2. Covert Malware in IoT Scenarios

Information hiding techniques are increasingly used by attackers to conceal malware in different carriers [2]. For example, network covert channels, i.e., hidden communications attempts nested within network traffic, can be used to secretly exfiltrate information or to elude well-known Intrusion Detection Systems (IDSes). Figure 1 depicts a possible reference scenario. In particular, it shows the covert sender, e.g., a compromised node of an IoT deployment, exchanging secret information via network features with the covert receiver, e.g., the Command & Control facility of the attacker. To do this, the sender could directly conceal data within the header of a protocol or encode the information in the temporal evolution of network packets belonging to a specific conversation.

Unfortunately, network covert channels are often neglected by standard security tools, thus revealing their presence is mandatory to fully assess the security of a modern network. To this aim, we addressed the problem of revealing hidden network communications targeting the IPv4 protocol in an IoT ecosystem [5]. We considered the exfiltration of data hidden in the Time To Live (TTL) field of a tampered IoT node, i.e., the bit "1" and "0" are encoded by the covert endpoints with two distinguished TTL values. To reveal the presence of such covert channels, we developed a detection mechanism based on AI. In particular, we leveraged autoencoders because of their capability to also deal with attacks undocumented and unknown *a priori*, as it happens when considering the carrier used to create the covert channel. To perform the detection, we monitored "windows" of network packets to extract statistical metrics related to the TTL, e.g, the maximum, the minimum, or the average TTL values. Only legitimate traffic information has been given to the autoencoder to perform the training. By contrast, the compromised traffic information has been used to evaluate the performance of the detection model. Results showcased the effectiveness of the AI-based approach, i.e., we obtained ∼91% and ∼94% for the accuracy and the precision, respectively. Despite the promising results, we extended the work evaluating an incremental learning scheme based on an ensemble of autoencoders trained on disjointed data chunks [6]. Figure 2 depicts the proposed architecture. Compared to the results obtained by using a single autoencoder, we obtained ∼95% both for the accuracy and the precision when using an ensemble of neural models. By using the incremental learning scheme, the model can also be deployed on devices with limited computational and storage resources, for instance in home gateways or edge nodes.

Although the ensemble-based model allows for improving the detection capabilities w.r.t. a single model, it requires to set the ensemble size, i.e., the number of windows to consider in learning the model. In addition, it will be trained only against the data available on a single edge node and the owners of the data could be not inclined to share them.

To overcome all these issues, we are interested in investigating the usage of a federated learning approach to address this task. In [7], we evaluated the benefits of this paradigm in a related (information-hiding) scenario in which malicious payloads are hidden within

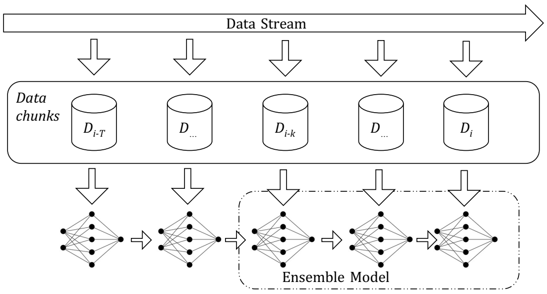

Figure 2: Incremental Deep Ensemble model approach.

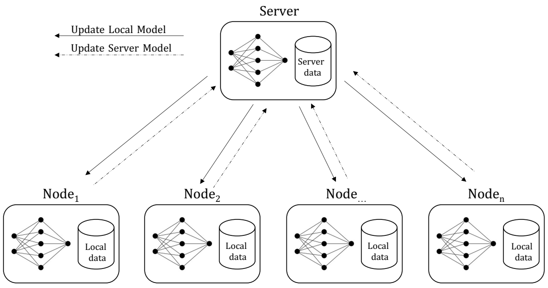

Figure 3: Federated Learning Architecture.

high-resolution icons that are commonly used in most popular mobile ecosystems, including, Android and iOS. Figure 3 depicts the proposed approach. We assume to have a centralized server acting the role of coordinator among *n* nodes. The server contains a "weak" DNN detector model trained on an initial dataset with a limited number of examples. In the early stage, this initial detector is shared across *n* end nodes, which then fine-tune their model against their own local data. To make the predictor more robust and to find a global model in a distributed manner, a subset of end nodes periodically sends updates to the server containing the weights of each layer composing their local DNN. The coordinator aggregates the information received to build an ensemble model and again shared it with the peers. This process is iterated until a certain convergence criterion is reached.

3. Container Security

Containers are now the preferred choice for the creation of scalable frameworks able to take advantage of the micro-service paradigm. Specifically, their lightweight nature offers many benefits compared to classical virtual machines, e.g., a smaller resource footprint or a reduced delay when deploying new services. Unfortunately, security requirements of containers are still not fully understood as for the case of virtual machines [8]. Among the various possible attacks, threat actors are increasingly exploiting techniques to let containers leak data beyond a well-defined execution perimeter [9]. This mechanism resembles the creation of a covert channel (see, Section 2): in this case, a shared local resource is altered to encode a secret message within its temporal evolution, which can be observed outside the single container. For instance, a sender process can manipulate the amount of memory used within its container to alter the overall (host-level) available memory. Then, in parallel, a receiving process running into another container can infer the message by inspecting such a global statistic, e.g., by reading an entry in the `/proc/` filesystem for the case of Docker [9, 10]. The security implications of such an "imperfect" isolation are several. First, containers can exchange information to orchestrate an attack and synchronize many processes to implement DDoS/slow-DoS attacks [10, 11]. Second, containers can leak information to recognize features of the underlying hardware/software infrastructure and support the reconnaissance stage at the basis of many complex attack chains [10].

To mitigate such stealthy attacks, AI is a promising approach, especially if jointly used with tools able to gather precise information on the behavior of the overall software architecture [12]. As an example, containers exchanging information in a secret manner could exhibit tight temporal correlations. This is due to the need of altering a resource and "decode" changes close in time, mainly to avoid that other competing processes will disrupt the encoded secret [13]. Hence, AI can be used to spot anomalies in the "wake-sleep" pattern of containers or to enlighten possible correlations difficult to capture with the common sense, e.g., the distillation of signatures in system calls used to access shared statistics exposed by the kernel of the guest OS. Another important application of AI concerns the creation of suitable whitelists/blacklists. In fact, containers can also exhibit tight-coupled interactions when part of the same service. The AI can be then leveraged to whitelist containers expected to have overlapped or correlated timing behaviors and prevent too aggressive detection rules. Techniques like process mining can be used to define precise traces of the system calls invoked by the software running within the various containers. This can allow for the early revealing of possible exfiltration attempts or feed an additional AI-based framework to perform runtime detection of leaking attempts.

Indeed, AI can also be employed to face other types of threats. For instance, the traffic exchanged by various containers or softwarized architectures can be used to detect exfiltration attempts, network-based attacks (e.g., DoS/DDoS) or the presence of crypto-miners sending data to a centralized master entity [14]. Besides, more classical approaches based on the analysis of logs or the (automatic) correlation of configurations and network traffic should be updated to handle the highly-mutable and dynamic nature of containerized services. Despite the considered threat, micro-service architectures implemented via containers present many research challenges and open issues, which are very stimulating. On one hand, the need of gathering, processing and storing traffic at "wire speed" without degrading the overall Quality of Experience for feeding the AI poses many technical challenges. On the other hand, the presence of sensitive data and the need of conforming to constraints like the General Data Protection Regulation (GDPR) impose to minimize the disclosure of personal or de-anonymizable bits. The GDPR also requires to consider both the physical and legal boundaries where data is stored and processed. In this case, softwarized networks and infrastructures are expected to greatly benefit from federated approaches, especially owing to the edge flavor of many future scenarios such as those based on 5G (see, e.g., [15] and the references therein).

Lastly, container security will also benefit from threat-specific hardening and configuration mechanisms. Also in this case, AI represents a stimulating asset. For instance, part of the software development life cycle of containerized applications [16] can take advantage of AI, especially to evaluate configurations, impose hardening constraints or match security issues against known CVEs or taxonomies. For the specific case of covert attacks, a promising use case for AI concerns the identification of resources that can be (ab)used to encode secret information and characterized by "loose" isolation properties. As a paradigmatic example, a suitable resource matrix [17] can be computed by inspecting containers and software (e.g., used libraries, code patterns, or execution privileges) to provide automatic configuration policies preventing leaking behaviors. This represents a major research challenge, especially to avoid that the container engine will terminate or impede the execution of "flagged" containers in a too aggressive manner, thus impairing the quality of the overall service.

4. Graph Generation for Detection

In real-world scenarios, such as social networks, biology, and recommender systems, complex relations among the entities of graphs can hardly be modeled as flat tabular data. In addition, many current AI-based solutions assume that the underlying graph is static, however real-world networks are dynamic as both their topology and dimension tend to change over time. Dynamic graphs [18] are typically adopted in several application domains, including cybersecurity with different purposes such as malware [19] and intrusion detection [20]. Dynamic evolution is difficult to model due to the dynamic nature of the underlying process, where continuous changes in the graph structure require flexible architectures able to handle such modifications. We plan to devise a deep-learning-based approach aiming at predicting graph evolution by considering stepwise changes (that can affect both, node and edge set). Moreover, to model long-term evolutions, we are interested in defining an architecture invariant to the network dimension. Learning the evolution of dynamic graphs by extracting their temporal and structural characteristics could be useful to identify polymorphic cyber attacks, such as polymorphic malware [21] or those leveraging some form of obfuscation. This can also help to assess the emerging wave of steganographic malware [22]. In essence, this class of threats leverages information hiding techniques to conceal malicious assets (e.g., configuration files or additional attack stages) in various software artifacts, such as images, executables or metadata. Unfortunately, being very threat-specific, generalizing their detection via standard mechanisms or signature-based approaches is a challenging task. In this vein, graph generation could lead to a more abstract and unified framework to early detect malicious software trying to reduce its footprint to remain unnoticed for long time frames.

5. Future and Main Challenges

To effectively deploy AI for taming covert and hidden threats, various challenges and refinements have to be addressed. First, the obtained data should model as faithfully as possible real scenarios dealing with cybersecurity. This is in general complex, but it becomes exceptionally hard when considering information-hiding-capable threats. In fact, the lack of publicly-available datasets capturing the presence of malware using covert channels or other elusive mechanisms is a well-known problem [2, 22]. Moreover, in publicly-available datasets samples that represent cyber attacks are rare. Hence, the class imbalance can dramatically affect the performances of the detection models. Therefore, it is crucial to devise approaches able to handle this skewness, e.g., by exploiting generative models able to produce realistic anomalies. Again, real data are difficult to retrieve, and, moreover, they are usually affected by noise. In this respect, unlabelled data can lead the systems to make errors if not properly handled as anomalies could be labeled as "normal", and, vice versa. In addition, due to data scarcity and skewness, the systems could erroneously classify infrequent legit behaviors (due to data shifts) as anomalous ones.

Moreover, some of the techniques briefly discussed in this work [5, 6, 7] are threat-dependent and difficult to generalize. For instance, AI-based models are able to detect hidden communication attempts only in fields of the IPv4 protocol sharing similar functionalities [6].

Therefore, a more general approach should be considered to devise flexible and robust models. A possible idea is to use more abstract metrics typical of at least a class of hiding mechanisms. Then, multiple AI pipelines can be deployed to efficiently detect a family of attacks, e.g., methods targeting IPv4 or IPv6 conversations [23]. In addition, in unsupervised settings, it is difficult to define boundaries between normal and anomalous behaviors. Again, this is especially hard when considering covert attacks, which are stealthy by-design. Similarly, challenges have to be faced when considering container security. On one hand, the overall security model still needs to be fully understood. On the other hand, the complexity of the various software layers requires proper modeling to efficiently provide data to the AI.

Lastly, as cyber-attacks can be represented via dynamic graphs, it is crucial to define strategies able to detect threats in evolving networks. Modeling and predicting the evolution of dynamic graphs is a challenging task due to their evolving nature. State-of-the-art systems lack flexibility, and, in this respect, models that guarantee invariance w.r.t the input size should be devised.

6. Conclusions

This paper discussed the use of AI to address emerging challenges, i.e., threats endowed with mechanisms able to covertly leak data in networked and softwarized environments. As discussed, each scenario requires facing different challenges. For instance, when dealing with IoT ecosystems, federated learning could be considered a main “technology enabler”, especially for distributing computation and guaranteeing that sensitive data remain confined at the border of the network. Instead, containerized architecture could benefit from AI to find correlations and patterns in the complex interplay of software components.

Unfortunately, the AI could also be exploited to hide data, thus spawning new threats. To this aim, graph generation should be carefully considered as it can help in finding a convenient representation of operations (both at the abstract level or in terms of system calls) to support the detection of advanced offensive schemes. Accordingly, part of our future research will address the aforementioned topics.

Acknowledgments

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

References

[1] M. J. H. Faruk, H. Shahriar, M. Valero, F. L. Barsha, S. Sobhan, M. A. Khan, M. Whitman, A. Cuzzocrea, D. Lo, A. Rahman, et al., Malware detection and prevention using artificial intelligence techniques, in: 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021, pp. 5369–5377.

[2] W. Mazurczyk, L. Caviglione, Information Hiding as a Challenge for Malware Detection, IEEE Security & Privacy 2 (2015) 89–93.

[3] D. Ucci, L. Aniello, R. Baldoni, Survey of machine learning techniques for malware analysis, Computers & Security 81 (2019) 123–147.

[4] T. Liu, Z. Liu, Q. Liu, W. Wen, W. Xu, M. Li, Stegonet: Turn deep neural network into a stegomalware, in: Annual Computer Security Applications Conference, 2020, pp. 928–938.

[5] M. Guarascio, M. Zuppelli, N. Cassavia, G. Manco, L. Caviglione, Detection of Network Covert Channels in IoT Ecosystems Using Machine Learning, in: Proc. of The Italian Conference on CyberSecurity, volume 3260 of *CEUR Workshop Proceedings*, 2022, pp. 102–113.

[6] N. Cassavia, L. Caviglione, M. Guarascio, A. Liguori, M. Zuppelli, Ensembling Sparse Autoencoders for Network Covert Channel Detection in IoT Ecosystems, in: Foundations of Intelligent Systems: 26th International Symposium, Springer, 2022, pp. 209–218.

[7] N. Cassavia, L. Caviglione, M. Guarascio, A. Liguori, G. Surace, M. Zuppelli, Federated learning for the efficient detection of steganographic threats hidden in image icons, in: Pervasive Knowledge and Collective Intelligence on Web and Social Media, Springer Nature Switzerland, Cham, 2023, pp. 83–95.

[8] S. Sultan, I. Ahmad, T. Dimitriou, Container security: Issues, challenges, and the road ahead, IEEE Access 7 (2019) 52976–52996.

[9] Y. Luo, W. Luo, X. Sun, Q. Shen, A. Ruan, Z. Wu, Whispers between the containers: High-capacity covert channel attacks in docker, in: Trustcom/BigDataSE/ISPA, IEEE, 2016, pp. 630–637.

[10] X. Gao, B. Steenkamer, Z. Gu, M. Kayaalp, D. Pendarakis, H. Wang, A study on the security implications of information leakages in container clouds, IEEE Transactions on Dependable and Secure Computing 18 (2018) 174–191.

[11] I. Vaccari, M. Aiello, E. Cambiaso, Slowtt: A slow denial of service against iot networks, Information 11 (2020) 452.

[12] F. Al-Doghman, N. Moustafa, I. Khalil, Z. Tari, A. Zomaya, Ai-enabled secure microservices in edge computing: Opportunities and challenges, IEEE Transactions on Services Computing (2022).

[13] C. Marforio, H. Ritzdorf, A. Francillon, S. Capkun, Analysis of the communication between colluding applications on modern smartphones, in: Proceedings of the 28th Annual Computer Security Applications Conference, 2012, pp. 51–60.

[14] J. A. Perez-Diaz, I. A. Valdovinos, K.-K. R. Choo, D. Zhu, A flexible sdn-based architecture for identifying and mitigating low-rate ddos attacks using machine learning, IEEE Access 8 (2020) 155859–155872.

[15] V. Rey, P. M. S. Sánchez, A. H. Celdrán, G. Bovet, Federated learning for malware detection in iot devices, Computer Networks 204 (2022) 108693.

[16] T. Rangnau, R. v. Buijtenen, F. Fransen, F. Turkmen, Continuous security testing: A case study on integrating dynamic security testing tools in ci/cd pipelines, in: 2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC), IEEE, 2020, pp. 145–154.

[17] R. A. Kemmerer, Shared resource matrix methodology: An approach to identifying storage and timing channels, ACM Transactions on Computer Systems 1 (1983) 256–277.

[18] S. Gupta, S. Bedathur, A survey on temporal graph representation learning and generative modeling, 2022. doi:10.48550/ARXIV.2208.12126.

[19] B. Anderson, D. Quist, J. Neil, C. Storlie, T. Lane, Graph-based malware detection using dynamic analysis, Journal in computer Virology 7 (2011) 247–258.

[20] G. Duan, H. Lv, H. Wang, G. Feng, Application of a dynamic line graph neural network for intrusion detection with semisupervised learning, IEEE Transactions on Information Forensics and Security 18 (2023) 699–714.

[21] E. Avllazagaj, Z. Zhu, L. Bilge, D. Balzarotti, T. Dumitras, When malware changed its mind: An empirical study of variable program behaviors in the real world., in: USENIX Security Symposium, 2021, pp. 3487–3504.

[22] L. Caviglione, W. Mazurczyk, Never mind the malware, here's the stegomalware, IEEE Security & Privacy 20 (2022) 101–106.

[23] W. Mazurczyk, K. Powójski, L. Caviglione, IPv6 covert channels in the wild, in: Proceedings of the third central european cybersecurity conference, 2019, pp. 1–6.