DOI: 10.26314/GARR-Conf21-proceedings-07

An Efficient and Privacy-Aware Method for Revealing Network Covert Channels

Marco Zuppelli, Luca Caviglione, Corrado Pizzi, Matteo Repetto

IMATI - Institute for Applied Mathematics and Information Technologies

Abstract. The ability of creating covert channels within network traffic is increasingly exploited by malware to elude detection and remain unnoticed. Unfortunately, spotting such hidden communication attempts often requires to evaluate composite and huge volumes of data, which may lead to scalability and privacy issues. We propose an efficient method for computing suitable indicators to reveal the presence of covert channels within the bulk of traffic. To meet performance criteria, we exploit code augmentation features of the Linux kernel. Privacy is guaranteed by using a counter-based mechanism not requiring to store information of the header. To prove its effectiveness, we tested our idea with a covert channel targeting IPv6 traffic.

Keywords. Covert channels, code augmentation, stegomalware, detection.

Introduction

Attacks like ransomware, cryptominers, and advanced persistent threats daily endanger individuals and large-scale organizations, highlighting the limits of common security tools and causing relevant economical losses. Among the various offensive techniques, an emerging trend concerns the adoption of information hiding or steganography to create attacks difficult to spot or detect. Accordingly, such a group of threats has been named stegomalware. In general, stegomalware implements methodologies to hide malicious routines or configuration files within innocent-looking pictures, bypass firewalls, implement stealthy multi-stage loading architectures, and force execution enclaves provided by sandboxing and virtualization (Caviglione et al. 2020).

A key component for the success of stegomalware is rooted in its ability to create network covert channels. In essence, a network covert channel allows two remote endpoints to secretly exchange data via the injection of arbitrary information within legitimate traffic flows (Zander et al. 2007). Unfortunately, detecting or mitigating such abusive communications pose several challenges. First, the feature exploited by the attacker is not known a priori, thus requiring to inspect various traffic entities (e.g., header fields, payloads or behaviors like the throughput or the inter-packet delay) at the same time. Second, adopting deep inspection techniques may cause scalability and performance issues. Third, gathering traffic information introduces additional fragilities in the privacy of users. In fact, even in the presence of suitable anonymization techniques, large volumes of data could make statistical guessing attacks a real concern (Burkhart et al. 2010).

Therefore, this chapter introduces a framework exploiting code augmentation features

Online, 7-16 giugno 2021 - ISBN: 978-88-946629-0-0

of the Linux kernel to efficiently inspect traffic and compute privacy-preserving metrics that can be used to reveal the presence of covert communications. The contributions of this chapter are:

- knowledge transfer: the diffusion of information-hiding-capable threats requires their understanding to completely assess the security of modern digital infrastructures ;
- support the development of digital infrastructures: enforcing security while guaranteeing suitable performance metrics is relevant to develop next-generation services. Yet, the use of lightweight mechanisms can mitigate hardware costs and the complexity of the software needed to gather information;
- data usage: the Internet contains data that can be used to profile or track users. Thus, being able to detect exfiltration attempts is of prime importance, while the development of privacy-by-design detection mechanisms or countermeasures can encourage their deployment.

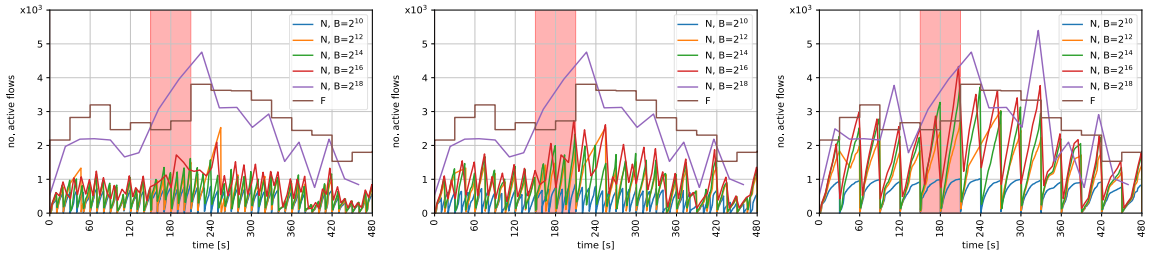
The rest of the chapter is structured as follows. Section 2 showcases how network covert channels can be detected by using indicators not leaking or collecting sensitive information, while Section 3 concludes the chapter and portraits possible future research directions.

2. Covert Channel Detection via Counters

To prove the effectiveness of our idea, in this chapter we consider attacks exploiting covert channels targeting the header of the IPv6 protocol, which have been observed in real threats or are expected to become a concern in the near future (Zander et al. 2007, Caviglione et al. 2021). As a paradigmatic example, in the following we discuss network covert channels built via the embedding of secret data in the Flow Label field.

The typical approach for detecting covert communications requires to trace network traffic with a per-flow granularity. For each stream, parameters like the number of packets, data transmitted on the wire and the average inter-packet delay are recorded. Values of fields suspected to contain secrets have to be stored, too. As a result, the memory footprint could be prohibitive and recorded information can disclose personal details or the user identity. To cope with such drawbacks, we propose a technique able to scale independently of the number of flows. Rather than keeping the "state" for each flow, we only count the number of occurrences for the different values that a given field assumes. To enforce scalability, multiple values may be grouped together into a "bin", and a single counter is used for the whole group. For instance, for the case of the Flow Label, bins can be a partition of its 20-bit space leading to $B \leq 2^{20}$ equally-sized containers. This approach, named as counters, guarantees privacy and anonymity, since no values of the field under investigation are stored. To pursue efficiency, the data collection phase has been implemented via the extended Berkeley Packet Filter (eBPF), which introduces a lightweight overhead and allows to have a general framework to gain visibility over network and software.

To prove the effectiveness of the approach, we present results partially borrowed from our ongoing research (Caviglione et al. 2021). Since each IPv6 conversation has a unique Flow Label, the number of bins N observed in a time window T can provide a rough estimation of the number of active flows. Discrepancies between a known behavior or measurements of an external monitoring tool (denoted as F in the following) can spot an



(a) $T = 10$ (b) $T = 15$ (c) $T = 30$

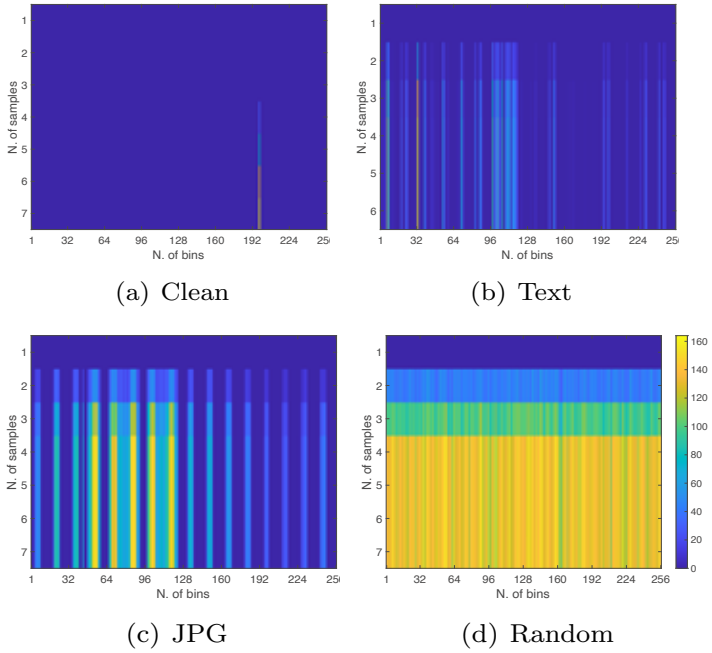
Fig. 1
Number of changing bins of the inspected traffic for different observation windows of T seconds. Red areas denote when a covert communication is present

attacker maliciously altering the field to exfiltrate data or to orchestrate an attack. Figure 1 shows the idea.

The number of changing bins can be also organized in a heatmap to produce a synthetic image capturing the evolution of the various values assumed by the Flow Label. Specifically, Figure 2(a) represents a "clean" network conversation, i.e., only the bin related to the licit Flow Label value increases, while Figures 2(b), 2(c), and 2(d) show a flow containing a covert channel exfiltrating text, a JPG image or random data, respectively. This "pictorial" representation of the channel does not contain any sensitive information and can feed AI-based frameworks to detect threats and classify the exfiltrated data (e.g., a .dll or a shellcode).

3. Conclusions and Future Works

This chapter presented a technique leveraging code augmentation for capturing the behavior of traffic in an efficient and privacy-preserving manner. The use of eBPF allows to easily extend the collection phase to other fields, protocols or traffic behaviors, thus improving the extensibility of the approach. Future works aim at refining the idea, especially



(a) Clean (b) Text

(c) JPG (d) Random

Fig. 2
Heatmaps for various covert transmissions, with B=28. Heatmaps have been computed only considering packets belonging to the covert channel

to identify the exfiltrated content or the class of malware using the covert channel. Part of the ongoing research is aimed at integrating the eBPF-based detection mechanism within cloud-based architectures or novel security toolkits.

References

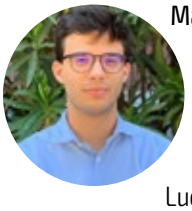
Burkhart M., Schatzmann D., Trammell B., Boschi E., Plattner, B. (2010), The role of network trace anonymization under attack, ACM SIGCOMM Computer Communication Review, 40(1), pp. 5-11.

Caviglione L., Mazurczyk W., Repetto M., Schaffhauser A., Zuppelli M. (2021). Kernel-level tracing for detecting stegomalware and covert channels in Linux environments, Computer Networks, (191), pp. 1-12.

Caviglione L., Choraś M., Corona I., Janicki A., Mazurczyk W., Pawlicki M., Wasielewska K. (2020), Tight arms race: overview of current malware threats and trends in their detection, IEEE Access, (9), pp. 5371-5396.

Zander S., Armitage G., Branch P. (2007), A survey of covert channels and countermeasures in computer network protocols, IEEE Communications Surveys & Tutorials, 9(3), pp. 44-57.

Authors



Marco Zuppelli marco.zuppelli@ge.imati.cnr.it

Marco Zuppelli is a research fellow at IMATI-CNR. He investigates novel detection methods for steganographic malware exploiting both network and local covert channels.

Luca Caviglione luca.caviglione@ge.imati.cnr.it

Luca Caviglione is a Senior Research Scientist at IMATI-CNR. Its prime research topics are information hiding, network covert channels and networking.

Corrado Pizzi corrado.pizzi@ge.imati.cnr.it

Corrado Pizzi is a technologist at IMATI-CNR and he graduated in Mathematics at Genoa University. He studied techniques of geometric reasoning for the extraction of shape features from discrete surface models and methods for data set simplification. He is responsible of the IMATI 3D scanning and printing laboratory and of the IMATI information systems and network services. His interests are also in network security and in system management.

Matteo Repetto matteo.repetto@ge.imati.cnr.it

Matteo Repetto is a Research Scientist at IMATI-CNR and he is the Scientific and Technical Coordinator of the EU projects ASTRID and GUARD. His research interests include network functions virtualization and service functions chaining, network and service security.