A Strings-Based Similarity Analysis Approach for Characterizing IoT Malware and Inferring Their Underlying Relationships

Sadegh Torabi*, Mirabelle Dib*, Elias Bou-Harb†, Chadi Assi*, and Mourad Debbabi*
*The Security Research Centre, Concordia Institute for Information Systems Engineering Concordia University, Montreal, Canada
†The Cyber Center For Security and Analytics, University of Texas at San Antonio, San Antonio, United States

***Abstract*—Mitigating threats associated with the rise of Internet-of-Things (IoT) malware requires creating a better understanding about the characteristics and inter-relations of IoT malware. In this letter, we perform a large-scale characterization of IoT malware. The analysis of 70,000 recently detected malware executables indicate that they belong to a few known families. Additionally, we highlight the lack of sophisticated IoT malware binary obfuscation. Thus, enabling reverse-engineering and static malware analysis, while performing a multi-level strings-based analysis to uncover groups of correlated IoT malware with common characteristics/features (e.g., adversarial IP addresses and malware-specific strings). Moreover, while our findings indicate malicious implementation reuse, we illustrate the rapid IoT malware evolution by identifying covid-related malware samples. Finally, this work provides a basis for developing AI-based malware detection/mitigation models, which benefit from the simplicity and reliability of the extracted strings-based characteristics/features for effective IoT malware classification and family attribution.**

***Index Terms*—IoT malware, strings-based similarity analysis, clustering, covid-related malware, adversarial IP address.**

I. Introduction

Inferring and mitigating threats associated with the Internet-of-Things (IoT) paradigm requires developing a better understanding about the behavioral characteristic of the rising number of IoT malware and their underlying relationships. This is a challenging task due to the lack of information about deployed IoT devices in the user space and the insecurity of such devices at scale. To address these challenges, various IoT malware/data collection initiatives have been introduced over the past years [1]–[3]. Such fundamental knowledge and data about IoT malware can be utilized to perform static malware analysis [4] and extract information on the structural and behavioral characteristics of the analyzed executable binaries, while creating a better understanding about the state of IoT malware and their underlying implementations [5], [6].

The analysis of the rising number of IoT malware/botnets [7]–[9] indicates that a considerable number of them have two main objectives: (i) malware propagation and botnet expansion by identifying and exploiting vulnerable IoT devices, and (ii) orchestrating large-scale DDoS attacks by leveraging compromised devices as attack enablers. To fulfill their objectives, IoT malware need to communicate with adversarial hosts (e.g., C&C servers) to obtain malicious command/payload and upload gathered information. This is typically achieved by embedding a series of commands and IP addresses to ensure successful post-infection communication for operating further malicious activities. Indeed, inferring such information from the malware binaries is key to understanding IoT malware dynamics and interrelationships. Nevertheless, this is challenging due to the common use of code obfuscation techniques by adversaries, which aim specifically at preventing automated static and signature-based malware analysis.

To this end, we leverage a specialized IoT Honeypot (IoTPoT [3]) along with malware-related information obtained from `VirusTotal` to analyze more than 70,000 IoT malware binaries/executables that were detected over 20 months (Sept. 2018 to May 2020). Motivated by our preliminary analysis, which highlights the lack of sophisticated code obfuscation within the analyzed IoT malware binaries, we propose a strings-based IoT malware analysis as a reliable and lightweight approach for performing static malware analysis while uncovering unique characteristics and underlying interrelationships among the analyzed malware binaries. Specifically, we utilize reverse-engineering techniques to extract meaningful strings from the analyzed binaries including adversarial IP addresses (e.g., C&C servers) and meaningful commands/strings. We perform a multi-level similarity analysis to uncover underlying relationships among the analyzed malware samples within and between different families.

Note that various static malware analysis techniques have been previously utilized to analyze malware binaries and extract malware-specific features, which can leverage AI-based malware detection and analysis [6], [10], [11]. For instance, Alasmary et al. [12] utilized features related to the malware control flow graphs (CFGs) to perform malware detection using deep learning methods. Gibert et al. [13] utilized convolutional neural networks (CNNs) along with image-based features for malware classification. Additionally, ensemble methods using a combination of features have been introduced for malware detection and classification [6], [10], [14]. Nevertheless, unlike these previous works, we are not attempting to build a detection/classification model for IoT malware. Instead, we explore the feasibility of utilizing strings-based features as a lightweight and yet reliable approach for characterizing IoT malware and investigating their similarities. Consequently, such features can drive future work towards implementing scalable IoT malware detection, classification, and attribution using deep/machine learning algorithms.

To this end, we summarize the main results/contributions of this work in the following:

- We analyze a large corpus of real IoT malware binaries/executables that were detected over a course of 2 years. We leverage a publicly available threat repository (`VirusTotal`) along with information about the detected IoT malware binaries to characterize various known IoT

malware families, while highlighting new, possibly undetected malware samples.
- Motivated by the lack of malware obfuscation in the context of IoT, or the use of common obfuscating techniques that are easily reverse-engineered, we design and execute a multi-level strings-based malware similarity analysis approach to cluster IoT malware executable binaries and investigate their underlying correlations by extracting adversarial IP addresses and/or embedded commands/payloads. We explored within and between malware family correlations and show that the detected IoT malware families can be in fact associated with a small number of implementations, which shed light on common practices used by adversaries to create and operate IoT malware in the wild.
- We shed light on the evolution of IoT malware by identifying covid-related IoT malware, which reflect the intentions of adversaries to abuse ongoing global events such as the covid-19 pandemic to distribute malware. Moreover, we demonstrate the effectiveness of our proposed strings-based similarity analysis by uncovering correlated clusters of covid-related IoT malware, which share similar command strings and/or adversarial IP addresses.

II. Methodology

In this letter, we shed light on the characteristics of IoT malware, while exploring the feasibility of strings-based analysis/features for implementing IoT malware detection and family attribution techniques. Specifically, we aim at answering the following main research questions (RQs):

1) *How can we leverage available information on IoT malware along with static malware analysis techniques to build a better understanding about the current state of IoT malware and the various IoT malware families?*
2) *How can we leverage strings-based information extracted from IoT malware binaries to explore malware characteristics and hidden interrelationships among malware samples? How can we leverage the proposed approach to infer IoT malware evolution?*

To answer the research questions, we propose a multi-level approach, which consists of the following main components: (i) static malware analysis through reverse-engineering and extraction of meaningful strings from the executable binaries to build a better understanding about the IoT malware and infer network-related characteristics and features, while inferring hidden interrelationships among different samples; and (ii) clustering analysis to explore structural and behavioral similarities among the analyzed IoT malware samples while inferring groups/communities of correlated samples.

A. IoT Malware Collection and Labeling

We leverage a known IoT-based honeypot (`IoTPOT` [3]) to obtain over 70,000 detected IoT-tailored malware samples between September 2018 and May 2020. Among these samples, we performed pre-processing steps to filter out corrupted files and/or samples with no executable data (e.g., HTML/-text), ending up with a large corpus of 49,272 IoT malware samples. Furthermore, to have a consistent malware attribution and labeling procedure, we leveraged `VirusTotal` and `AVClass` to obtain malware information such as family name, whenever available (Table I). `AVClass` is an open-source tool that uses a ranking/voting system to select the most likely family name for a given malware sample based on reported information/labels (e.g., `VirusTotal`) by multiple antivirus vendors [15]. Note that `AVClass` cannot assign malware family names when no family name/labels are associated to them by antivirus vendors, or when they are labeled with generic (e.g., linux) malware names. We label those samples as "Generic" for further analysis.

B. Static Analysis (Strings-Based Analysis)

An effective static malware analysis approach is to explore embedded indicators such as commands, payloads, and other identifiable information by extracting meaningful strings from the executable binaries [6]. In this letter, we analyzed IoT malware by utilized reverse-engineering techniques to extract meaningful strings from the binary code. More specifically, we use regular expressions and text-based analysis techniques to obtain IP addresses associated with possibly malicious hosts controlled by adversaries (e.g., C&C servers). For instance, as shown in Listing 1, the IoT malware is trying to use an HTTP get request to download malicious payload (bins.sh) from the specified host (http://103.*.*.*/). Furthermore, it is clearly observed that the malware is using different techniques to download malicious scripts/payloads, as presented by the consequent instructions/commands using the TFTP protocol (e.g., tftp 103.*.*.* -c get tftp1.sh). Moreover, it is interesting to see that 786 IoT malware samples contained masked IP addresses (Table I), which are associated with targeted destination IP addresses and subnets (e.g., 36.248.%d.%d). While such behavior is not common among IoT devices, it can provide a clear indication of targeted scanning behaviors by the analyzed malware.

```
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /;
wget http://103.*.*.*/bins.sh; chmod 777 bins.sh;
sh bins.sh; tftp 103.*.*.* -c get tftp1.sh;
chmod 777 tftp1.sh; sh tftp1.sh; tftp -r
tftp2.sh -g 103.*.*.*; chmod 777 tftp2.sh;
sh tftp2.sh; ftpget -v -u anonymized -p anonymized
-P 21 103.*.*.* ftp1.sh ftp1.sh; sh ftp1.sh;
rm -rf bins.sh tftp1.sh tftp2.sh ftp1.sh; rm -rf *
```

Listing 1. Extracted strings with adversary IP address.

Packed/Obfuscated Malware Binaries. Malware packing/obfuscation is a common practice, which aims at scrambling the actual code of the malware to evade detection and prevent automatic detection and analysis using conventional methods. As a result, we were unable to extract useful strings and IP addresses from about 21,037 IoT malware binaries, representing about 42.7% of the analyzed samples (Table I). To investigate malware packing/obfuscation, we looked common indicators of packing and obfuscation methods. Interestingly, we identified about 52% of the obfuscated samples (10,938 samples) to be packed using `UPX`,[1] which is an open source program for compressing executable files. We leveraged a combination of manual and automatic reverse-engineering

[1]The Ultimate Packer for eXecutables https://github.com/upx/upx

methods using tools such as UPX [16] and IDA Pro [17] to unpack the UPX-packed binaries for further analysis. In fact, we were able to extract useful strings and IP addresses from about 84.6% of the upx-packed samples (9,145 out of 10,938 samples). Finally, we were unable to unpack/decrypt about 20% of the analyzed malware binaries and thus, no useful strings/IPs were extracted from them for further analysis. One reason could be due to the fact that the adversaries leverage unique techniques for obfuscating/packing their code. It is also possible that the collected malware binaries were corrupted and therefore, contained no useful information. We consider the implementation of further malware de-obfuscation techniques for analyzing those samples for future work.

C. Similarity Analysis

We perform similarity analysis in terms the identified network information within the obtained strings from the malware executables. For each analyzed malware binary, we leverage Natural Language Processing (NLP) techniques such as word tokenization to process the identified strings and extract meaningful words (e.g., commands and IPs). We use a combination of the Jaccard ($J(A,B) = \frac{|A \cap B|}{|A \cup B|}$) and overlap ($O(A,B) = \frac{|A \cap B|}{min(|A|,|B|)}$) similarity coefficients to compare the identified words and obtain correlated malware samples in terms of the shared adversarial IP addresses and/or analyzed strings. Moreover, we leverage ClusterONE [18] algorithm to investigate correlated malware samples and identify groups of similar IoT malware implementation. ClusterONE tries to discover densely connected and possibly overlapping regions (high cohesiveness) by starting from a single seed vertex, and adopting a greedy strategy to extend the group with new vertices so that the newly added vertex to increases the cohesiveness of a group. The clustering outcomes can be in fact used to highlight granular similarities among a group of malware samples, which reflect the unique characteristics of the IoT malware and their underlying implementations.

D. Limitations

The generalization of our results might be hampered by the fact that we rely on a single source, namely IoTPoT [3], to obtain real samples of IoT malware binaries. Furthermore, IoTPoT is deployed on a limited number of IP addresses, which mainly interact with Telnet-specific requests. Despite these limitations, it is worthy to mention that the analysis of Internet-scale scanning activities generated by compromised IoT devices showed that Telnet ports (e.g., TCP 23/2323) are indeed among the most predominantly targeted ports/services by IoT malware [7], [8], [19]. Furthermore, the deployed IoT honeypot (IoTPoT) have been shown to be more robust towards capturing various IoT-tailored attacks as compared to other honeypots (e.g., Honeyd)[2]. Thus, addressing the generalizability of our findings through the analysis of a large and representative sample of real IoT malware executables, which covers a variety of detected attacks by different IoT malware variants/families over the past two years.

[2]http://www.honeyd.org/

TABLE I
A SUMMARY OF THE ANALYZED IOT MALWARE SAMPLES. MALWARE FAMILY NAMES ARE OBTAINED USING AVCLASS AND VIRUSTOTAL. THE LAST TWO COLUMNS REPRESENT THE NUMBER OF SAMPLES WITH ADVERSARIAL (ADVER.) AND/OR TARGET IP ADDRESSES.

Malware Family	Count	%	Packed Samples	Packed UPX	Adver. IP	Target IP
Mirai	**42,537**	**86.33**	18,552	10,409	33,146	–
Gafgyt	1,024	2.08	593	185	605	8
Tsunami	73	0.15	19	19	73	–
Ircbot	39	0.08	39	8	–	–
Silex	6	0.01	–	–	6	–
Bricker	4	0.01	–	–	4	–
Other	4	0.01	2	–	1	–
Unknown	**5,327**	**10.81**	1,649	300	3,988	778
Generic	258	0.53	183	17	81	–
Total	49,272	100	21,037	10,938	37,904	786

III. EXPERIMENTAL RESULTS

A. Identified IoT Malware Families

As summarized in Table I, the analyzed samples belong to a handful of IoT malware families, with majority of the detected IoT malware samples (about 86%) to be labeled as Mirai, followed by a significantly fewer detected samples as Gafgyt. While the prevalence of Mirai malware samples comes in line with prior studies that analyzed the behaviors of infected IoT devices in the wild [8], [19], it can indicate the fact that IoT malware authors tend to reuse the existing Mirai implementation, especially when they have been effectively used to exploit IoT dvices with weak/default credentials. In addition, about 11% of the analyzed malware samples were not found in VirusTotal reports, meaning that they have not been detected by the major antivirus vendors.

Moreover, we extracted useful strings from the analyzed malware samples. In general, about 77% of the analyzed malware binaries (37,904 out of 49,272) contained one or more IP addresses associated with adversaries (Table I). These addresses are likely to be associated with downloading instructions or payloads from hosts that are controlled by the adversary (e.g., Listing 1). More specifically, the strings analysis uncovered 37,904 malware executables that contained adversarial IP addresses. Interestingly, these adversarial IP addresses correspond to 7,340 unique IP addresses, which are distributed across 55 countries, with about half of them (50.46%) located in the U.S., as illustrated in Figure 1.

Furthermore, about 2% of the identified malware samples (786 samples) contained masked IP addresses associated to possible scanning targets. Indeed, the analysis revealed a total of 134,901 target IP addresses, which correspond to 2,083 unique IP address and/or subnet masks (e.g., 123.123.%d.%d). As summarized in Table I, only 8 Gafgyt malware samples contained both adversarial and target IP addresses, while the remaining 778 malware samples were not associated to known malware families according to VirusTotal reports, which may indicate new malware variants/families that are yet to be discovered by antivirus vendors.

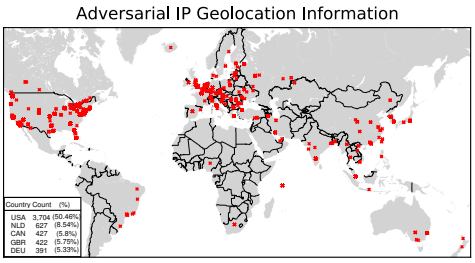

Fig. 1. The distribution of the adversarial IP addresses across 54 countries.

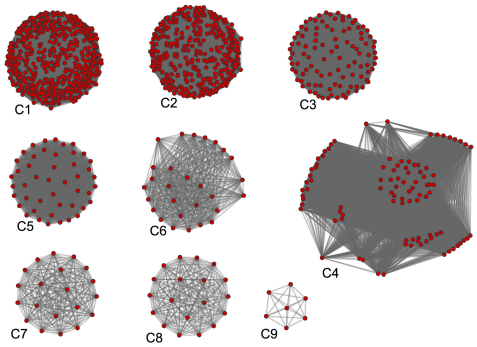

Fig. 2. Covid-related malware clusters.

B. Similarity Analysis: Covid-Related IoT Malware

This research was done during the covid-19 pandemic, which resulted in a surge in the overall Internet usage. More importantly, adversaries were found to abuse the situation to spread covid-related malicious content and malware. Interestingly, we identified 1,535 IoT malware samples that contained covid-related strings (i.e., "covid", "covid-19" or "corona"). The majority of these samples belong to `Mirai` (1,388 out of 1,418), with a significantly fewer samples to be associated with `Gafgyt` (68). Interestingly, 79 covid-related malware samples were not associated with any known IoT malware, indicating new IoT malware that has not been yet discovered by antivirus vendors, as seen from `VirusTotal`'s report. To investigate malware interrelationships, we select the covid-related IoT malware samples to perform the proposed similarity analysis considering two aspects: (i) the strings-based similarities, and (ii) the identified adversarial IP addresses.

1) Strings-Based Similarity: We performed strings-based analysis on the obtained covid-related strings within the analyzed malware samples. Our analysis revealed 27 unique string/command combinations that were shared across all covid-related samples. Moreover, our manual investigation of those 27 unique string/command indicated further similarities, which could be leveraged to reduce the number of groups by combining similar covid-related strings. To do so, we leveraged our strings-based similarity analysis technique to identify further correlated strings. We used a lower bound of 30% ($threshold \geq 0.3$) in our similarity analysis to ensure the quality of the obtained groups while eliminating loosely correlated samples from the analysis. We obtained an adjacency matrix representing the pairwise strings similarity measure ([0.3,1]), with the vast majority of the correlated malware samples showing very high similarity measures ($\geq$ 75%). Consequently, we performed clustering analysis (`ClusterONE` algorithm) by initially converting the obtained adjacency matrix into an undirected network with malware samples as vertices and edges representing the pairwise strings similarity coefficient. Indeed, the results highlight 9 clusters of mutually exclusive and densely-connected IoT malware samples (C1–C9), as depicted in Figure 2.

We summarize the clustering results in Table II. For instance, the command "`/bin/busybox CORONA`", which uses busybox to execute possibly malicious script named "CORONA", was found across 728 samples, representing about 52% of total covid-related samples.

TABLE II
UNIQUE COVID-RELATED STINGS WITHIN THE ANALYZED STRINGS.

CID	Members	%	Density	Covid String Example
C1	798	51.99	1.00	/bin/busybox CORONA
C2	420	27.36	1.00	Corona.
C3	103	6.71	1.00	DESC='corona'
C4	94	6.12	0.82	wget -g ping.covid-19.casa
C5	46	3.00	1.00	corona
C6	28	1.82	0.95	cat shto.sh>Corona3.sh
C7	20	1.30	1.00	Corona_64 Coronav5l
C8	19	1.24	1.00	Coronavirus.
C9	7	0.46	1.00	jupiter.covid-19.casa

In addition, the command "`/bin/busybox wget -g http://**.covid-19.casa`" was used in 86 malware samples to access a set of covid-related domain names for downloading/uploading various malicious payload and scripts. Additionally, all clusters resulted in high cluster density ($density = \frac{\sum \text{Edge weights}}{|Edges|}$), which indicates that the clustered malware samples share almost the exact covid-related string combinations (Table II). Moreover, it is clearly observed that the detected groups have no inter-relationships in terms of the identified strings similarities, thus, resulting in a statistically significant difference in the number of in-weights as compared to out-weights (Mann-Whitney U test with $p < 0.001$).

2) Adversarial IP-Based Similarity: Additionally, about 86.6% of the identified covid-related IoT malware samples (1,330 out of 1535) contained adversarial IP addresses. Given a set of IP addresses associate with each IoT malware binary, we performed similarity analysis by identifying the pair-wise similarity coefficient using Jaccard index. We set the similarity threshold to zero to capture all common IP addresses, while excluding isolated samples along with samples that did not contain any IP addresses from further analysis.

As illustrated in Figure 3, our IP-based similarity analysis uncovered 40 groups of correlated malware samples (with $max = 515$ and $min = 2$ adversarial IP addresses), which indicate the underlying relationship in terms of common malware operators/authors. Furthermore, as highlighted by the red circles in Figure 3, the majority of the identified groups consist of samples that belong entirely to `Mirai` family, while only two identified groups (C2–C3) belong to `Gafgyt`, respectively. In addition, the largest group of correlated IP addresses (C1) contained a small fraction of IP addresses

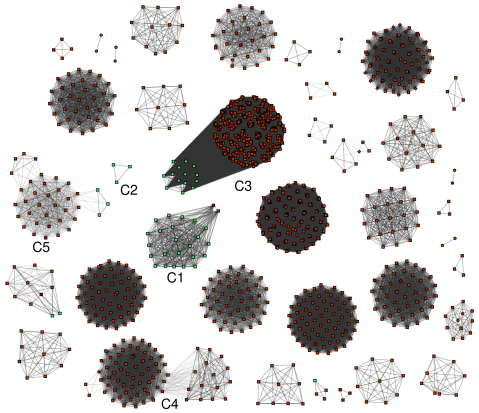

Fig. 3. Covid-related adversarial IP clusters.

that belong to `Gafgyt` (greed nodes), which indicates the possible descriptiveness in the provided family information by `VirusTotal` due to the overlapping characteristics of the involved IoT malware. This, highlights the challenging tasks of malware labeling and the limitations of existing IoT malware labeling techniques employed by antivirus vendors.

From a different perspectives, the majority of the identified groups in Figure 3 represent fully connected covid-related malware samples, which reflects a common adversarial behavior in terms of reusing existing resources to recreate various copies of the malware. We also note that some clusters such as C4 and C5, consist of multiple fully connected components, which are connected through few peripheral nodes. These connected sub-components highlight the use of different sets of IP addresses within various groups of IoT malare. Nevertheless, this behavior might be justified by the fact that some adversarial IP addresses might be blacklisted over time, and thus, forcing malware operators to leverage new sets of IP addresses along with the ones that are not yet blacklisted.

IV. Conclusion

In this letter, we leveraged a specialized IoT honeypot to obtain a representative sample of IoT malware and analyze their executable binaries. Our findings indicate that the majority of the detected malicious executables fall under a handful of known families (e.g., Mirai and Gafgyt). Moreover, motivated by the lack of sophisticated malware obfuscation within the IoT context, we introduced strings-based similarity analysis as a reliable and lightweight approach for extracting useful information from IoT malware binaries while uncovering unique characteristics and interrelationships among the analyzed samples. More specifically, our results shed light on the emergence of covid-related IoT malware, which indicate the rapid IoT malware evolution while highlighting the aggressive behaviors of adversaries towards abusing global events for maximizing malware propagation and distribution. Additionally, our strings-based similarity analysis uncovered mutually exclusive groups of correlated IoT malware samples with common adversarial IP addresses and/or extracted strings. Our findings demonstrate the fact that IoT malware authors heavily rely on reusing previous code/malware executables to generate new instances of malicious executables. This is due to a number of main factors such as the resource constraints on the deployed IoT devices, which makes it difficult to create sophisticated malware executables. Moreover, previous studies showed that compromised IoT devices are treated as disposable attack enablers, which might not have a significant value by themselves. Therefore, adversaries might not invest much time/effort towards building novel attack techniques given that they can reuse/tweak previous code and implementation such as the released Mirai source code to create and propagate slightly customized malware.

Finally, while our analysis helped in better understanding the IoT malware threat landscape, we shed light on the effectiveness of the strings-based analysis and features for uncovering correlated IoT malware. This can be indeed utilized for developing IoT malware detection and mitigation through the implementation of deep/machine learning models that utilize strings-based features to perform malware classification and family attribution.

References

[1] B. Wang *et al.*, "IoTCMal: Towards A Hybrid IoT Honeypot for Capturing and Analyzing Malware," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.
[2] "Shodan," Retrieved from https://www.shodan.io/, 2019.
[3] Y. M. P. Pa *et al.*, "IoTPOT: Analysing the Rise of IoT Compromises," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, Washington, D.C., 2015.
[4] Q.-D. Ngo *et al.*, "A survey of iot malware and detection methods based on static features," *ICT Express*, 2020.
[5] E. Cozzi *et al.*, "The Tangled Genealogy of IoT Malware," in *The Annual Computer Security Applications Conference (ACSAC)*, 2020.
[6] M. Alhanahnah *et al.*, "Efficient Signature Generation for Classifying Cross-Architecture IoT Malware," in *IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–9.
[7] P.-A. Vervier and Y. Shen, "Before Toasters Rise Up: A View into the Emerging IoT Threat Landscape," in *Int. Symp. on Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 556–576.
[8] M. S. Pour *et al.*, "On Data-driven Curation, Learning, and Analysis for Inferring Evolving Internet-of-Things (IoT) Botnets in the Wild," *Computers & Security*, p. 101707, 2019.
[9] M. Antonakakis *et al.*, "Understanding the Mirai Botnet," in *Proc. of the 26th USENIX Security Symp.*, Vancouver, BC, 2017, pp. 1093–1110.
[10] D. Gibert, C. Mateu, and J. Planes, "HYDRA: A Multimodal Deep Learning Framework for Malware Classification," *Computers & Security*, p. 101873, 2020.
[11] I. U. Haq and J. Caballero, "A Survey of Binary Code Similarity," *arXiv preprint arXiv:1909.11424*, 2019.
[12] H. Alasmary *et al.*, "Analyzing and detecting emerging internet of things malware: A graph-based approach," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8977–8988, 2019.
[13] D. Gibert *et al.*, "Using convolutional neural networks for classification of malware represented as images," *J. of Computer Virology and Hacking Techniques*, vol. 15, no. 1, pp. 15–28, 2019.
[14] R. Islam *et al.*, "Classification of malware based on integrated static and dynamic features," *J. of Network and Computer Applications*, vol. 36, no. 2, pp. 646–656, 2013.
[15] M. Sebastián *et al.*, "AVClass: A Tool for Massive Malware Labeling," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2016, pp. 230–253.
[16] UPX Team, "UPX." [Online]. Available: https://upx.github.io/
[17] Hex-Rays, "IDA Pro." [Online]. Available: https://www.hex-rays.com/products/ida/
[18] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting overlapping protein complexes in protein-protein interaction networks," *Nature methods*, vol. 9, no. 5, p. 471, 2012.
[19] S. Torabi *et al.*, "Investigating Internet-Scale Reconnaissance Activities by Compromised IoT Devices Through The Lens of A Large-Scale Network Telescope," *IEEE Transactions on Dependable and Secure Computing TDSC (DOI: 10.1109/TDSC.2020.2979183)*, 2020.

View publication stats