POSTER: Towards Identifying Early Indicators of a Malware Infection

Sareena K P*
sareena@cse.iitm.ac.in
Indian Institute of Technology Madras
India

Chester Rebeiro
chester@cse.iitm.ac.in
Indian Institute of Technology Madras
India

Unnati Parekh
unnati.parekh502@gmail.com
India

Kamakoti V
kama@cse.iitm.ac.in
Indian Institute of Technology Madras
India

ABSTRACT

A malware goes through multiple stages in its life-cycle at the target machine before mounting its expected attack. The entire life-cycle can span anywhere from a few weeks to several months. The network communications during the initial phase could be the earliest indicators of a malware infection. While prior works have leveraged network traffic, none have focused on the temporal analysis of how early can the malware be detected. The main challenges here are the difficulty in differentiating benign-looking malware communications in the early stages of the malware life-cycle. In our quest to build an early warning system, we analyze malware communications to identify such early indicators.

CCS CONCEPTS

• **Security and privacy → Malware and its mitigation**; **Network security**; **Firewalls**.

ACM Reference Format:
Sareena K P, Chester Rebeiro, Unnati Parekh, and Kamakoti V. 2019. POSTER: Towards Identifying Early Indicators of a Malware Infection. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3321705.3331006

1 INTRODUCTION

The increasing scale and sophistication of cyber-attacks against critical infrastructures has become a major cause of concern to the industry. These attacks are triggered using malware such as worms, virus, and trojans. Securing these infrastructures is inevitable as malware can trigger sophisticated attacks that can potentially bring down infrastructures, ex-filtrate sensitive data, create a Distributed Denial of Service (DDoS) attacks among many others. The complexity of these attacks clearly demonstrate the need for a quick detection and isolation of malicious behaviours in these networks.

Malware detection is primarily based on two approaches: *signatures* or *heuristics*. Signature-based methods rely on previously observed unique identification features either in the malware binary or when it is executed. While this has high precision, they can be easily evaded by modern day polymorphic and metamorphic malware [4]. The second approach relies on heuristics to determine rules that can differentiate malicious behavior from benign ones. While these rules facilitate detection of zero-day malware, they suffer from high false positives. Additionally, these methods detect the malware only after execution of the infected program.

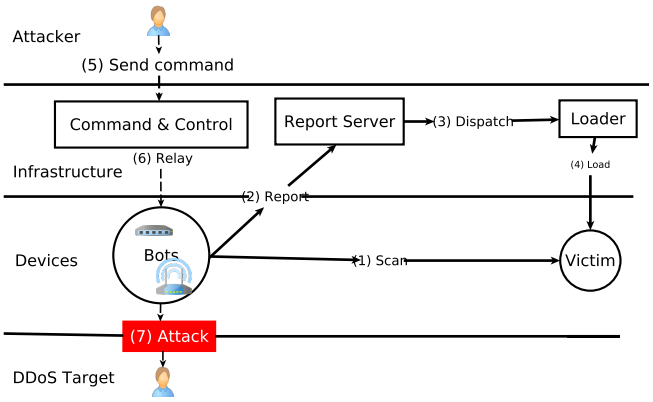


Figure 1: The life-cycle of Mirai [3]. The malware is active long before the actual attack in stage 7.

Analysis of the life-cycle of most malware reveals that a malware goes through multiple stages before the actual attack. At every stage, the malware relies on network communication with its external command-and-control (C&C) servers, to receive updated code and instructions for activity [3, 7]. Such an arrangement facilitate polymorphic behaviors in malware as well to stage the actual attack after a thorough reconnaissance of the target [5]. For instance, Fig.1 depicts the life-cycle of the Mirai malware [3], which triggers a DDoS attack in step 7. However, it is known that the malware is active in the system (and network) much before the actual attack. During this time it looks for new targets (step 1), reports an infection to its master (step 2), and, receives code and instructions (steps 3-6) for subsequent actions. A recent study on malware communications reveal the potential of network traffic as *early warning signals* of a malware infection *several weeks and often months* before the malware sample is discovered by a detection mechanism [7]. Early identification of a compromise facilitates quick and effective attack mitigation, by enabling network defenders to restrict the scope of infection and reduce the damage due to the attack.

In this work, we analyze malware network communications to identify *early indicators of malware infection*. Such indicators facilitate automated malware analysis for building early warning and detection system. However, the main challenge is to differentiate the early indicators from behaviours of a benign communication. At the same time, such indicators need to be invariant to the polymorphic behaviours of the malware. To this end, we analyze 30

*Corresponding Author

AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand

ACM ISBN 978-1-4503-6752-3/19/07.
https://doi.org/10.1145/3321705.3331006

Table 1: Polymorphic behaviour of the malware in the CTU Dataset [1]

Malware	Scenarios	# Flows	# 1	# 2	# 3
Neris	IRC, SPAM, CF	5942	40+	40	7212
RBot	IRC, PS, US	30016	10	4	11
Virut	SPAM, PS, HTTP	193	17	5	33
DonBot	PS	1557	8	3	6
Sogou	HTTP	17	6	4	17

#1: Number of Patterns in Packet Length Difference; **#2**: Number of Patterns in Periodicity; **#3**: Number of DNS queries; **IRC**: Internet Relay Chat; **CF**: Click Fraud; **PS**: Port Scan; **US**:Compiled and controlled by Dataset author.

features from malware network communication traces to determine their suitability to assist in early detection. We also train an ML model using these features and investigate if it can detect malware even when the test data has few initial flows of the network traffic.

2 RELATED WORKS

Multiple prior works [4, 6–8] have analyzed malware communications. Among them, [7] is notable for its extensive study that highlights the potential of malware communications as early indicators of malware infection. The others employ machine learning to analyze network-traffic for malware detection. The work [4] proposes a feature representation for HTTP traffic that is invariant to certain polymorphic behaviors of malware (like payload, URL path, intensity, timing, etc.). On the other hand, [6, 8] address HTTPS traffic as well and detect malware by identifying the protocols, applications running on the host, or the servers infected by the malware. While these prior solutions have addressed encrypted traffic and polymorphism, they do not deal with the temporal behaviors and life-cycle of the malware to build early warning systems.

3 EARLY INDICATORS

In this section we discuss the challenges that drive the choice of features, and the features we consider in our preliminary study.

3.1 Challenges

Early identification of malware behavior faces multiple challenges. First, it is crucial that these mechanisms detect early behaviors (features) of a malware, before the execution of the target attack of the malware. While this would need extensive long-term data capture of malware life-cycle, there is a lack of such data sets. To this end, we perform our initial studies for early indicators on an available malware data set [1], and consider features that would pertain to early stages of any malware life-cycle. At the same time, we emphasize on the need to collect such long-term data for our future work.

Second, the indicators identified should be *invariant* to the dynamic threat landscape that involves polymorphic and metamorphic malware. Most modern-day malware are equipped to make their behaviors dynamic using techniques like domain generation algorithms (DGA) and server-side polymorphism among others. Table 1 highlights the polymorphic behaviour in the malware in the data set we used. We use k-means clustering to evaluate the number of patterns observed in packet features. For instance, for the Neris malware in a 6 hours capture involving 5942 flows, 40+ patterns of packet length difference and 7212 DNS queries were observed. While the former indicates polymorphism using packet features, the latter indicates polymorphism using DGAs. The malware may exhibit polymorphism across multiple dimensions. While we analyze 30 features in the current study, we consider invariance to DGA and server-side polymorphism alone in this poster. An extensive analysis of the impact of polymorphism on the selected features is left for future work. Finally, a growing proportion of malware use encrypted HTTPS protocol to evade detection by network traffic analysis. Hence, the features should also include Secure Socket Layer (SSL) and Transport Layer Security (TLS) features.

Table 2: Features

Cat.	#	Feature
I	1	Number of DNS Failures
	2	Number of NxDomain Failures
	3	Number of SERVFAIL failures
II	1	Validity of Certificate
	2	Average of the public key length used
	3-5	Number, Standard Deviation and Average of Certificate Paths
	6	Number of communications using the latest TLS version
	7	Ratio of self signed certificates in all certificates
	8	Number of Certificates
	9	Number of Certificates per SSL handshake
	10-12	Mean Number of Domains belonging to the certificates, Is Server Name Indication specified?, Is Common Name used?
	13-14	Ratio of SSL records having Server Name Indication (SNI), Is IP address used in SNI?
	15	Average flow Duration
	16-17	Periodicity - Average and Standard Deviation
	18	State of Established Connection
	19	Ratio of size of outbound packets to inbound packets
	20	Ratio of connections that use HTTPS and HTTP protocol
	21-22	Flow Duration - Average and Standard Deviation
	23	Total size of inbound payload
	24	Number of inbound packets
	25	Total size of Flows
	26	Number of outbound packets
	27	Number of Flows

Cat.: **Category**; **I**: **Source-based Features**; **II**: **Connection-based Features**;

3.2 Features

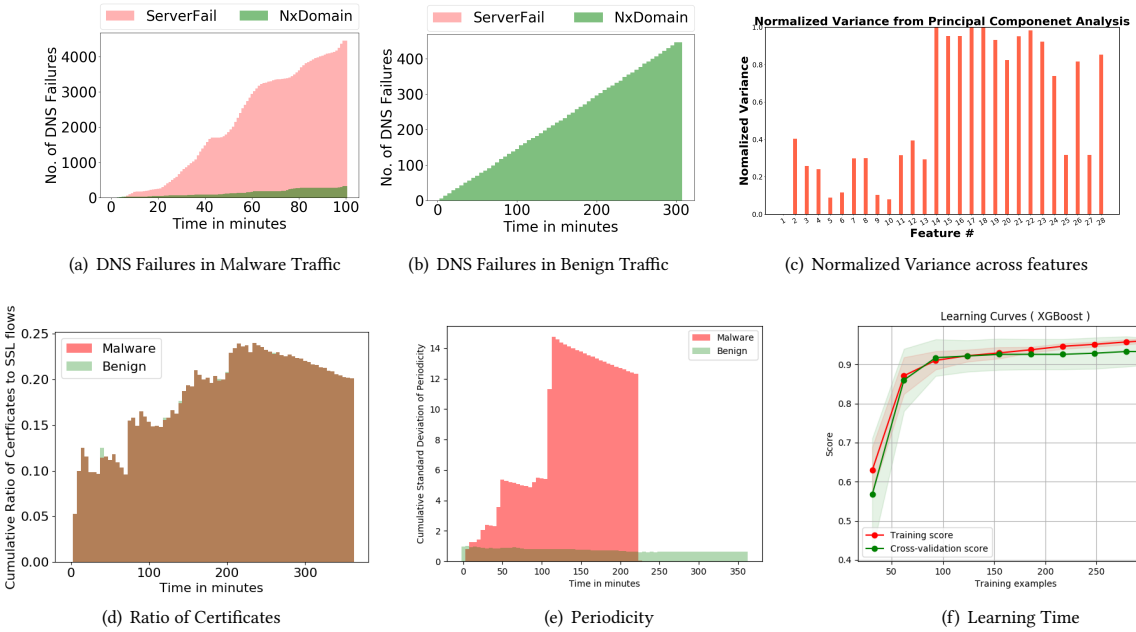
Table 2 lists the features we evaluated for the analysis. We classify the features into two:

Source-based features capture behaviour specific to each host in the network. Recall that most malware employ DGA to exhibit polymorphism. When running DGA, the malware makes a large number of DNS requests, many of which fail. We leverage this by choosing features related to the number of DNS failures and error codes.

Connection-based features are specific to each connection in a host. We define a connection as the tuple ⟨*source IP, destination IP, source port, destination port*⟩. These include the features related to SSL handshakes, duration, number of established connection states, and, the number and size of packets (outbound and inbound) in the connection. We also determine the periodicity of communications in the benign and malicious connections (between a source IP and destination IP) in the data set. This is because, malware is known to have a heartbeat, which are periodic messages to communicate its presence and activity in the target network to the corresponding C&C servers. To assess periodicity, we take the average of second order time difference of periodic messages in a connection. It is important to note here that, a malware may maintain a periodic connection to the C&C using different IP address, such that the IP flows are not periodic, whereas the HTTPS flows are periodic [5].

4 METHODOLOGY AND RESULTS

We extract the DNS, connection and SSL features using Zeek [2] scripts on the PCAPs as well as from the ARGUS logs from the labeled dataset [1]. We run principal component analysis (PCA) and Kullback Leibler (KL) Divergence to identify the features of higher variance. We next plot the cumulative frequency distribution



(a) DNS Failures in Malware Traffic (b) DNS Failures in Benign Traffic (c) Normalized Variance across features

(d) Ratio of Certificates (e) Periodicity (f) Learning Time

Figure 2: Results

(CDF) of important features for both malware and benign distributions against time to quantify how early they can contribute to the detection of malware. Finally we implement ML models to detect malware, and analyze if the trained model can detect malware even when presented with a least sequence of packets in the traffic.

DNS Failures. Figs. 2(a) and 2(b) plot the number of DNS failures observed from a host running Neris malware and a benign host in the dataset respectively against time. As evident, the number of DNS failures is very high (in 1000s) in the malware traffic as compared to the benign traffic. Additionally, for the malware, the error code SERVFAIL[1] was predominant as compared to NXDOMAIN[2]. The same was observed for other bots in the dataset as well. The results indicate that DNS SERVFAIL error codes can be used as potential early indicators in malware detection.

Connection-based features. Fig. 2(c) plots the variance across all connection-based features derived from PCA. From the plot, among the features evaluated, malware and benign traffic differ considerably in features related to periodicity and flow duration as compared to other features. Figs. 2(d) plots the cumulative ratio of number of flows containing certificates to the total number of SSL flows in the connection against time. The plot indicates that almost all malware flows use certificates similar to benign flows and hence is not an indicator of compromise. Fig. 2(e) plots the cumulative standard deviation of periodicity observed in the malware and benign traffic against time. The plot indicate that periodicity can be a potential indicator of a malware communication.

Early Detection. Fig. 2(f) plots the training and cross validation scores of the ML model trained using the first 50% of flows in the PCAP file. Though the accuracy of detection is low initially, the model is able to detect with $> 90\%$ accuracy when trained with first < 100 flows in the PCAP file.

[1]Server failed to complete the DNS request.
[2]Domain name does not exist.

5 CONCLUSION

Malware exhibits characteristic patterns in its network communications. In this poster, we highlighted a few such potential indicators and proposed to leverage them to build an early warning system. The evaluated ML model showed its potential to detect malware early. However, an extensive evaluation of long term malware communications is necessary to differentiate the benign-looking behaviors of the malware with higher accuracy. To make the system adapt to the changing threat landscape, we propose to build an automated feature extraction system to extract the malware specific indicators of compromise in the future.

ACKNOWLEDGMENTS

This work is supported by DST-FIST program Grant 2016, Department of Science and Technology, India.

REFERENCES

[1] [n. d.]. The CTU-13 Dataset. https://www.stratosphereips.org/datasets-ctu13 Accessed: 2019-02-17.
[2] [n. d.]. The Zeek Network Security Monitor. https://www.zeek.org/ Accessed: 2019-04-17.
[3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. 2017. Understanding the Mirai botnet. In *USENIX Security Symposium*. 1092–1110.
[4] Karel Bartos, Michal Sofka, and Vojtech Franc. 2016. Optimized Invariant Representation of Network Traffic for Detecting Unseen Malware Variants.. In *USENIX security symposium*. 807–822.
[5] Yehonatan Cohen and Danny Hendler. 2018. Scalable Detection of Server-Side Polymorphic Malware. *Knowledge-Based Systems* 156 (2018), 113–128.
[6] Jan Kohout and Tomáš Pevnỳ. 2015. Automatic discovery of web servers hosting similar applications. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 1310–1315.
[7] Chaz Lever, Platon Kotzias, Davide Balzarotti, Juan Caballero, and Manos Antonakakis. 2017. A Lustrum of malware network communication: Evolution and insights. In *IEEE Symposium on Security and Privacy (SP), 2017*. IEEE, 788–804.
[8] Jakub Lokoč, Jan Kohout, Přemysl Čech, Tomáš Skopal, and Tomáš Pevnỳ. 2016. k-NN classification of malware in HTTPS traffic using the metric space approach. In *Pacific-Asia Workshop on Intelligence and Security Informatics*. Springer, 131–145.