ADAM: Automated Detection and Attribution of Malicious Webpages

Ahmed E. Kosba*, Aziz Mohaisen†, Andrew G. West†, and Trevor Tonn‡
*Dept. of Computer Science, University of Maryland, College Park, akosba@cs.umd.edu
†Verisign Labs, {amohaisen, awest}@verisign.com
‡iDefense, Verisign, ttonn@verisign.com

I. Introduction

Malicious webpages are a prevalent and severe threat in the Internet security landscape. This fact has motivated numerous static and dynamic techniques for their accurate and efficient detection. Building on this existing literature, this work introduces ADAM, a system that uses machine-learning over network metadata derived from the sandboxed execution of webpage content. Machine-trained models are not novel in this problem space. Instead, it is the dynamic network artifacts (and their subsequent feature representations) collected during rendering that are the greatest contribution of this work.

There were two primary motivations in exploring this line of research. First, *iDetermine*, VeriSign's status quo system for detecting malicious webpages is a computationally expensive one. While that system is the basis for our ground-truth and network metadata herein, it also does a great quantity of other analysis to arrive at accurate labels (*e.g.,* packet inspection, system calls). We envision our efforts could well integrate as a tiered classifier that enables greater scalability with minimal performance impact. Second, existing literature on webpage classification [1]–[5] were able to provide promising accuracy. Because these approaches rely primarily on static features, we hypothesized that metadata from network dynamics might assist in the task.

This exploration is not without challenges. First, webpages face a host of vulnerabilities: exploit kits, defacement, malicious redirections, code injections, and server-side backdoors – all with different signatures. This malice may not even be the fault of the webpage owner (*e.g.*, advertisement networks). Moreover, the distribution of behavior is highly imbalanced, with our dataset having 40× more benign objects then malicious ones. Despite these challenges, our approach is currently broadly capable of 96% accuracy, with injection attacks and server-side backdoors being identified as areas for performance improvement and future attention.

In the following sections, we present the system description followed by an outline of our approach. Then, we provide our preliminary results and discuss future research directions.

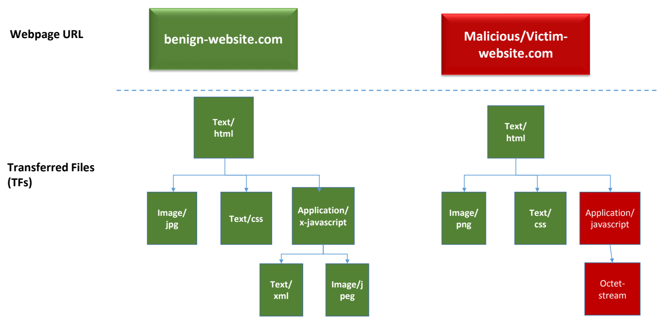

Figure 1. Two examples of transferred file trees

II. *iDetermine* and Dataset Description

iDetermine is Verisign's system for classifying URL behavior. It crawls websites using an orchestrated and virtualized web browser. For each analyzed URL the system maintains records of each HTTP request-response made while rendering that page. This results in a tree-like structure (see Fig. 1) where each node stores basic file attributes and network information (*e.g.*, HTTP response code, IP address, and Autonomous System (AS) number). We term the nodes in these graphs *transferred files* (TFs). These nodes also contain classification data from iDetermine's deep analysis and we use this as ground-truth in training/evaluating our approach. A URL is considered malicious if any of its child TFs are malicious.

Our dataset consists of 20k webpages, 10k each of "malicious" and "benign" types. These URLs were randomly selected from iDetermine's operational history of Internet-scale crawling. Analyzing these URLs yields 800k benign TFs and 20k malicious TFs. The distribution of vulnerabilities is visualized in Fig. 2.

III. ADAM: The Approach

The end goal of ADAM is to classify webpages based on the basic metadata stored in *iDetermine*, but not any complex and expensive features. The approach we developed was to build a classifier that learns about the properties of each transferred file retrieved while visiting each class of the URLs. Then, whenever a URL is analyzed during operation, the basic properties of each TF are used to predict whether it is malicious or not. A URL is labeled as benign if all of its TFs were classified as benign by the classification algorithm.

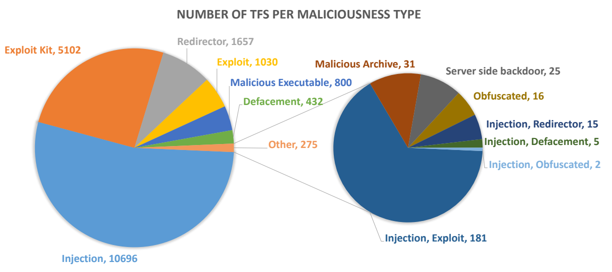

Figure 2. Distribution of malice among the TFs

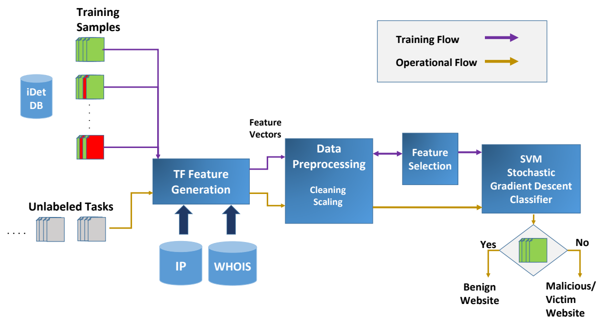

Figure 3. The workflow for classifying URLs based on TFs

Nearly 40 features are used in the the classification process, falling in the following broad categories:

- BASIC NETWORK DATA: HTTP header; AS number; IP address prefix; IP geolocation; WHOIS info.
- LEXICAL: URI length; subdomain count; path depth; bag-of-words representation for URI tokens.
- TREE-BASED: Parent-child host/IP diversity; node depth; number of children.

Broadly, we consider the above to be *network metadata* as each feature speaks to transport properties, addresses, or linking structures. These are trained into an SVM classifier based on stochastic gradient descent, with Fig. 3 illustrating the overall workflow of ADAM.

The iDetermine system does process and store additional data that could be useful in classification. For example, payload/content-based features derived from Javascript as in [1], [3], or flow information features as in [1] can be extracted and utilized. We do not integrate these features in order to maintain a content-agnostic and scalable classifier.

IV. PRELIMINARY RESULTS

Evaluation was performed using 10-fold cross-validation. Tab. I enumerates performance metrics at both TF and webpage granularity, showing an overall result of 7.6% FN rate and 6.3% FP rate. The reason for having a 14.7% FN rate on the TF-level is that simple metadata may not be indicative for all types of TF malice behavior. Additionally, compared to previous literature, the TF results are consistent with respect to the fact that our TF records dataset is highly imbalanced. Literature studies showed that as the data gets highly imbalanced, the accuracy degrades, e.g. 25% FN rate at a ratio of 100:1 of benign to malicious URLs [5].

Table I
EVALUATION RESULTS

	Acc.	Prec.	Recall	F-score	FP-rate	FN-rate
TF-level	0.965	0.390	0.850	0.530	0.0314	0.147
page-level	**0.930**	**0.935**	**0.924**	**0.930**	**0.063**	**0.076**

Table II
EVALUATION RESULTS W/O "INJECTION" SAMPLES

	Acc.	Prec.	Recall	F-score	FP rate	FN rate
TF-level	0.982	0.527	0.873	0.657	0.0153	0.126
page-level	**0.966**	**0.948**	**0.951**	**0.949**	**0.0257**	**0.048**

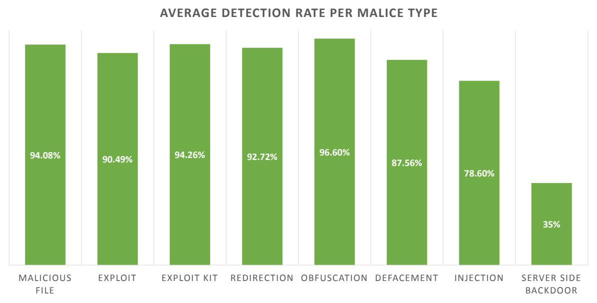

Figure 4. Detection rate per TF vulnerability type

To better understand how well the detection mechanism performed, Fig. 4 shows the detection rate per each vulnerability/attack type at the TF-level. Note that the "injection" and "server side backdoor cases" were most detrimental to overall performance. This is made clear in Tab. II which provides overall performance absent those problematic instances, resulting in 2.5% FP rate and 4.8% FN rate overall.

V. CONCLUSION AND FUTURE WORK

This poster has presented results from our ongoing work, ADAM, which utilizes machine learning over network metadata collected during dynamic webpage execution. Absent "injection" and "server-side backdoor" vulnerabilities our system has proven effective at tolerable false-positive rates. At this point we are working to develop features to correct this weaknesses. We also hope to explore the global TF graph to gain broader perspective of malice on the Internet and infrastructure re-use by miscreants. Finally, we hope to install our classifier in a live fashion to investigate operational challenges in large-scale contexts.

REFERENCES

[1] L. Xu, Z. Zhan, S. Xu, and K. Ye, "Cross-layer detection of malicious websites," in *CODASPY*, 2013.
[2] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in *ACM KDD*, 2009.
[3] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: a fast filter for the large-scale detection of malicious web pages," in *WWW*, 2011.
[4] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *IEEE Security and Privacy*, 2011.
[5] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious urls," *ACM Trans. Intell. Syst. Technol.*, 2011.