Early Detection of Malicious Webmail Attachments Based on Propagation Patterns

Yehonatan Cohen
Ben-Gurion University
yehonatc@cs.bgu.ac.il

Danny Hendler
Ben-Gurion University
hendlerd@cs.bgu.ac.il

Amir Rubin
Ben-Gurion University
Microsoft Israel R&D Center
t-amirub@microsoft.com

ABSTRACT

Email remains one of the key mediums used by cybercriminals for distributing malware. Based on a large data set consisting of antivirus telemetry reports, we conduct the first comprehensive study of the properties of malicious webmail attachments. We show that they are distinct among the general web-borne malware population in terms of the distributions of file-types, malware *reach* (the number of machines to which the malware is downloaded), and malware type and family. Furthermore, we show that malicious webmail attachments are unique in the manner in which they propagate through the network.

We leverage these findings for defining novel features of malware propagation patterns. These features are derived from time-series representation of malware download rates and from the community structure of graphs that model the network paths through which malware propagates. Based on these features, we implement a detector that provides high-quality early detection of malicious webmail attachments.

CCS Concepts

•Security and privacy → Malware and its mitigation; Browser security; Network security;

1. INTRODUCTION

Traditional antivirus software relies on signatures to uniquely identify malicious files. Once a file is determined to be malicious, its signature is computed and added to a signatures database. Malware writers, on the other hand, have responded by developing obfuscation techniques with the goal of evading such content-based filters. Polymorphic and metamorphic malware utilize dead-code insertion, subroutine reordering, encryption, and additional techniques, in order to alter a file's content and make signature-based detection more difficult [24, 34]. A consequence of this arms race is that numerous new malware instances are generated every day, thus limiting the effectiveness of static detection approaches. For effective and timely malware detection, signature-based mechanisms must be augmented with detection approaches that are harder to evade.

Email remains one of the key mediums used by cybercriminals. With a daily rate of messages sent and received exceeding 200 billion in 2015, it is probably still the most popular Internet-based application [19]. It is anticipated that over one third of the worldwide population will be using email by the end of 2019. Unfortunately, email has become a fertile ground for the distribution of spam mail, consisting of unsolicited messages, often sent in bulk. According to recent statistics, in spite of a gradual decline in spam rates over the last few years, more than half of inbound business email traffic during 2015 was spam [11].

Spam mail is more than just a nuisance: it can be dangerous, because it is often used for phishing attacks and for spreading malware. Despite spam and malware filters, email-borne malware in general, and malicious attachments in particular, are still very much alive and kicking. Recent examples include the Win32/Gamarue worm, which was the most commonly encountered threat in 2H15 [9], the DRIDEX online banking malware [10], and many more.

The majority of PC users today use webmail rather than desktop email clients. Indeed, over the course of 2015, webmail opens represented almost 58% of all non-mobile email opens, whereas desktop email opens represented approximately 42% [30].

1.1 Our Contributions

Based on a large data set that we received from Microsoft, consisting of antivirus telemetry reports triggered by file downloads from the web, we conduct in this paper the first comprehensive study of webmail attachments and their unique characteristics.

First, we find that there is a clear dichotomy between webmail attachments and other web-downloaded files: the vast majority of files that are downloaded at least once from a webmail server are downloaded exclusively from webmail servers. Having established that webmail attachments are a distinct sub-population of the larger web-downloaded files population, we set out to identify their characteristics. In particular, we investigate what features of the webmail malware population distinguish it from the general population of web-downloaded files.

We find that the distributions of the two populations in terms of file types, *reach* (number of machines to which a file is downloaded), malware classes and malware families are very different. We then show that, even more interestingly, *webmail attachments are unique in the manner in which they*

WWW '17 Perth, Australia

DOI: 10.475/123_4

propagate through the web in terms of both space and time.

We model time propagation using *download rate vectors* (DRVs, see Section 3.1) — time-series that represent the rate in which files spread through the network over time. Our analysis of DRVs reveals that the propagation process of webmail-attached malware is characterized by short bursts, separated by relatively extended periods of quiescence. We use *dynamic time warping* [4, 23] for deriving from DRVs features that are able to identify this behavior.

We represent the manner in which webmail-attached files propagate in space using *file-location graphs* (see Section 3.4), composed of nodes representing files on the one hand, and nodes representing their origins (domains, URLs) or destination machines, on the other hand. By applying a community detection algorithm to these graphs, we are able to derive reputation-related features locally within each community.

We implement a detector for malicious webmail attachments, based on the collection of features we identified. Our detector employs a supervised classification algorithm, trained using ground truth labels that we received from Microsoft and data we collected from VirusTotal (VT). Our performance evaluation shows very good results in terms of accuracy. Moreover, it establishes that our detector is able to provide high quality early detection of new malicious webmail attachments, significantly earlier than their first submission to VirusTotal.

The rest of this paper is organized as follows. In Section 2, we describe our data set, show empirically that webmail attachments are a distinct sub-population within the wider population of web-downloaded files, and reveal several important differences between the two populations. In Section 3, we model the propagation of web-downloaded files and derive features that capture the unique propagation patterns of webmail-malware in space and time. In Section 4, we describe our detector, evaluate its performance and discuss deployment considerations. We survey related work in Section 5 and conclude with a short discussion in Section 6.

2. WEBMAIL ATTACHMENT PROPERTIES

This work is based on a large data set of telemetry reports that we received from Microsoft. The data set includes a subset of the telemetry reports that were generated by Microsoft Windows antivirus software (such as Windows Defender, Security Essentials, and Windows Live OneCare) on client machines running Windows 7 or 8 and sent to a backend logging server, in the course of 14 days during 01/06/15-14/06/15. Each telemetry report in the data set was generated when a file that was downloaded from the web to a Windows-based machine triggered an antivirus scan.

Each report contains a SHA-1 hash that uniquely identifies the contents of the file whose download triggered the report generation. For privacy reasons, file names were not provided to us. Instead, a hash value of the name was provided, which is a unique identifier of the file name but not a unique identifier of the file, since the same file may appear on different machines (and/or in different times) under different names. Each report also contains the report generation time and a unique anonymized identifier of the client machine. There are also attributes specifying the IP address and URL from which the file was downloaded.

The *classification* attribute stores the result of the antivirus scan. If the file was found to be malicious, it specifies the high-level malware class (e.g. trojan or worm). In this case, the *malware family* field stores an ID of the malware family to which the file belongs. These two fields were not made available to us as part of the reports. Instead, we received from Microsoft ground truth data that contains the classification of data set files and the families to which those of them that are malicious belong. We label data set files as malicious or benign using this data. See Section 4.1 for more details.

Table 1. Our data set's dimensions.

	Total	Webmail attachments
Number of files	3,880,957	154,410
Malware files	187,834	19,864
Number of machines	9,052,518	3,172,844
Number of server IPs	302,045	112,893
Number of domains	141,718	42,474
Number of URLs	3,130,339	468,733

Table 1 presents the dimensions of our data set. The left column presents data pertaining to all the reports in our data set and the right column presents data pertaining only to email attachments downloaded from webmail servers. It can be seen that our data set encompasses almost 3.9 million different files, approximately 150,000 of which were downloaded from webmail servers. The total number of malicious files is slightly more than 200,000. Files were downloaded from approximately 3.6 million different URLs to over 9.3 million different client machines.

Although there is no report field that indicates whether or not a file was downloaded from a webmail server, this can be inferred by searching the domain name (derived from the URL field) in publicly available lists of known webmail domains.[1]

Are files either downloaded from the web *only* as webmail attachments or *never* downloaded as webmail attachments? In other words, is there a clear dichotomy between files that are downloaded from webmail servers and files that are not? Our evaluation shows that the answer to this question is mostly positive.

Fig. 1a presents the distribution (and cumulative distribution, represented by the yellow curve) of *webmail exclusiveness* for files that have been downloaded from a webmail server at least once. Exclusiveness is defined as the ratio between the number of times a file was downloaded from a webmail server and the total number of its downloads. Approximately 90% of these files are downloaded exclusively from webmail servers.

Having determined that files downloaded from the web can mostly be categorized as webmail attachments or not, we now set out to check whether there are significant differences between the characteristics of webmail attachments and those of other web-downloaded files in our data set. In what follows, we consider a file that was downloaded from a webmail server at least once as a webmail attachment.

File Types

Table 2 shows the distribution of file types for those types that constitute at least 1% of the respective population.[2]

[1] We use the lists in [2, 3].

[2] File names are hashed for privacy reasons, but file name extensions were provided to us.

Table 2. File type distributions of webmail files and other files.

File type	*Benign files*		*Malware files*	
	Webmail	Other	Webmail	Other
.exe	29.68%	75.92%	3.82%	65.64%
.zip	33.59%	17.77%	81.42%	15.08%
.rar	4.04%	1.56%	6.04%	13.89%
.docx	3.79%	0.93%	2.15%	0.21%
.xls	10.3%	0.62%	0.9%	0.13%
.bat	0.56%	1.16%	0.02%	0.13%
.js	9.22%	1.24%	0.78%	0.05%

It can be seen that the distributions of the general web-downloaded files population and the webmail files population are very different from one another. As shown by the left-hand side of Table 2, while the most frequently downloaded benign file type in the general population are executable files (approximately 76% of all files), zip files are more common within webmail attachments and constitute about one third of all attachments. Excel files, constituting approximately 10% of benign webmail attachments, represent only approximately 0.6% of the general web-downloaded benign files population.

Focusing on malicious files (the right-hand part of Table 2), we find significant differences as well. Zip files account for more than 81% of webmail malware, but represent only approximately 15% of general web-downloaded malware. While zip is the format of choice for distributing malware via webmail attachments, the most "popular" format in the general malware population are executable files, accounting for almost two thirds of all web-downloaded malware.

It is noteworthy that common file extensions, such as pdf and jpg, are very rare in our data set (and are consequently not shown in Table 2). For instance, pdf files account for only 0.04% and 0.15% of the general malware and webmail malware data set populations, respectively. This is most probably much lower than the rate of pdf webmail attachments downloaded in general. The reason for this discrepancy is that, as noted before, *web-downloaded files appear in our data set only if they triggered antivirus scans.*

Malware Classes

Malicious files are classified according to the ways in which they infect their targets, propagate, and behave inside the compromised system. Standard key malware classes include viruses, worms, trojan downloaders, backdoors, and software bundlers. Malware instances from all these classes appear in our data set. Table 1 shows that the rate of malware in the population of webmail attachments is significantly higher than in the general population. Are there also significant differences in terms of malware classes?

Fig. 1b compares the malware-class distributions of the webmail malware population and the general web-downloaded malware population. There are huge differences between these distributions. Almost 80% of the general malware population is labelled as *software bundlers*. The corresponding figure for webmail malware is negligible, which is to be expected, since software installation programs are seldom sent via email. On the other hand, approximately 50% of webmail malware are *trojan downloaders*, whereas only 3% of the general malware population (which are approximately 13% of this population after disregarding software bundlers) are classified as such. Another significant difference is in the rate of worms, which account for approximately 9% of webmail malware but for less than 1% of the general malware population.

Malware Reach

We define the *reach* of a file f to be the total number of client machines to which the file is downloaded. We note that it is not necessarily the case that the malware gains access to all these machines, as the malicious file's signature may have already been known when (some or all of) these downloads took place.

Fig. 1c presents the reach distributions of webmail malware and of the rest of the malware population. The x-axis is the number of downloads in log 2 scale. It can be seen that the webmail malware's distribution is relatively skewed to the right, hence the average webmail malware is downloaded to more machines than the average non-webmail malware. The vast majority of malware are downloaded to less than 500 machines, but there are exceptions.

Fig. 1d focuses on webmail files and presents the reach distributions of webmail malware files and benign files. The distribution of the malware files is skewed to the right relative to that of the benign files, indicating that malware files have larger reach on average.

A file that is found to be malicious is associated with the family to which the malware belongs. As we received family IDs rather than family names, we refer to these families as *family1*, *family2*, etc. We examined the malware families that are most "popular" in our data set. For each family, we computed its *webmail collective reach*, defined as the total number of machines to which at least one member of the family was downloaded via webmail. We compute a family's *non-webmail collective reach* in a similar manner.

Fig. 1e presents the distributions of the webmail and non-webmail collective reach of the 15 most popular malware families. It can be seen that the collective reach of webmail malware is much larger in all of these families and that the members of a few malware families are distributed almost exclusively via webmail. In 152 of the 228 families that contain both webmail files and non-webmail files, the collective webmail reach was larger.

Fig. 1f presents, for each family, the average number of machines to which malware files from that family are downloaded. The average is computed separately for webmail and non-webmail malware. The differences in average reach are even more pronounced, and the average reach of webmail malware exceeds that of non-webmail malware by a factor of between 5.2-87.

We have seen that webmail-attached malware has unique characteristics in terms of relatively simple features such as type, reach, malware class, and family distributions. In the following section, we show that webmail attachments are distinct also in terms of the manner in which they propagate in both space and time.

3. MODELING WEBMAIL ATTACHMENT PROPAGATION

In this section, we show that the propagation patterns of malicious webmail attachments are unique and can form the basis of effective detection of this type of malware.

The propagation of files through the network may be ob-

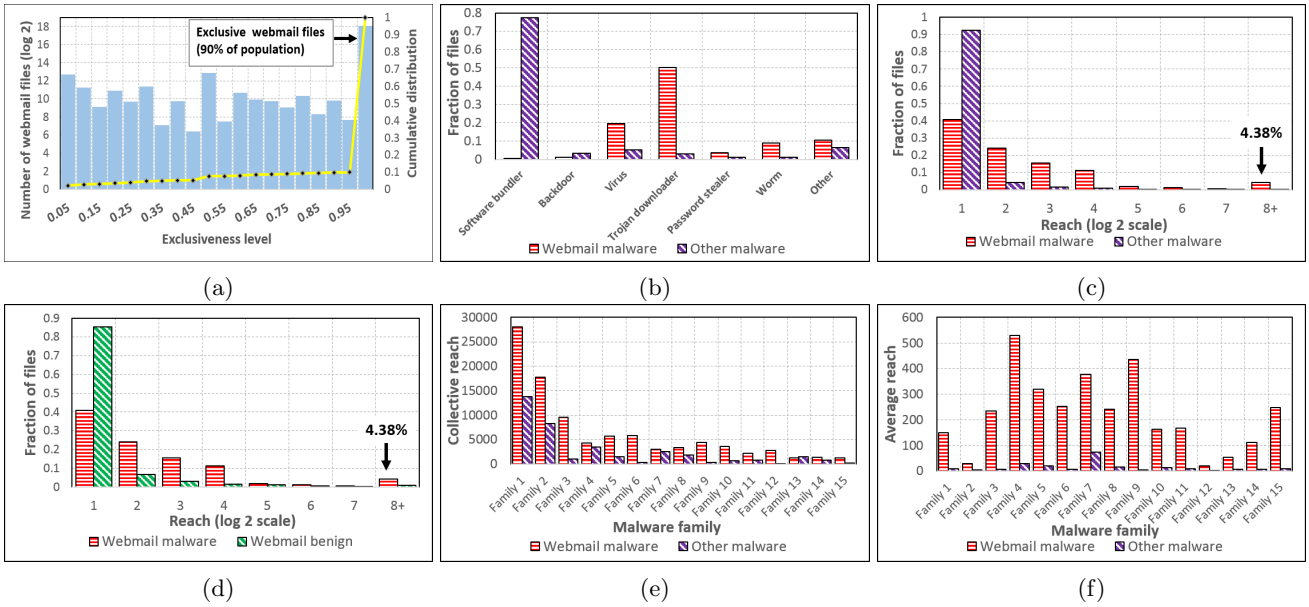


Figure 1. (a) Distribution of webmail file exclusiveness (b) Comparison of the class distributions of webmail malware & other malware (c) Malware reach distributions for webmail malware & other malware (d) Webmail reach distribution for malware & benign files (e) Distributions of malware families collective reach (f) Average reach for webmail & non-webmail malware.

served and analyzed along two different dimensions. On the one hand, files propagate over time, with varying download rate and burst patterns. This propagation dimension has a natural representation as a time-series. On the other hand, a file propagates in both virtual and physical spaces when it is downloaded from origin IPs and URLs to physical machines. We represent this dimension of propagation using what we call *file-location graphs*. In what follows, we describe the representation and derived features we use to model each of these two propagation dimensions.

3.1 Modeling Time Propagation

We associate with each data set file a *Download Rate Vector* (DRV), which is a time-series that models the rate in which the file propagates through the network over time. To compute the DRV for file f, the analyzed time period (e.g., the 14 days covered by our data set, or a single day) is divided to multiple equal-length *time bins*, which are smaller time intervals (e.g., days or hours). The number of times f was downloaded in the course of each time bin is then computed and stored to the corresponding DRV entry. Our analysis of the data set reveals that the propagation rate of webmail-attached malware over time is extremely bursty. This is unlike other types of files, both malware and benign, whose propagation rate over time is much more stable.

Figs. 2a-2b show the DRVs of a few representative webmail-attached malicious files found in our data set. Fig. 2a depicts a couple of relatively higher-volume malware, each with an overall download count exceeding 50, whereas Fig. 2b depicts a couple of lower-volume malware, each with an overall download count of less than 50. Each bin extends over a single day. It can be seen that, regardless of volume scale, the DRVs of these malware exhibit sharp but relatively short peaks, interspersed by relatively long periods of little to no activity. This indicates that the propagation process of webmail-attached malware is characterized by short bursts in which the malware is downloaded to numerous machines, separated by relatively extended periods of quiescence.

In general, other types of files are characterized by a much more stable propagation process. Figs. 2c-2d respectively present the DRVs of a few representative higher-volume and lower-volume webmail-attached legitimate files. As can be seen, there are no sharp peaks separated by extended periods of quiescence, and changes in the volume of activity over time are gradual. Our analysis shows that the propagation process of non webmail-attached malware is relatively stable as well (graph omitted for lack of space). Collectively, these results establish that *the propagation process of webmail-attached malware possesses unique characteristics.*

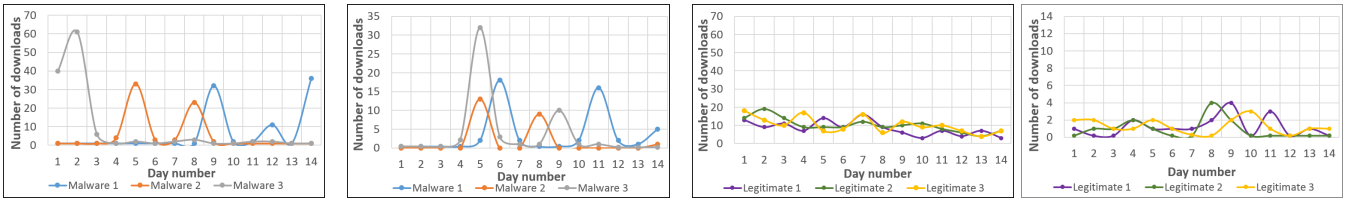
How can these characteristics be explained? In other words, what are the unique underlying propagation mechanisms of webmail-attached malware that result in the sharp download-rate peaks we observe in Figs. 2a-2b?

Unlike legitimate emails that are typically sent to a few recipients only [13, 29], *malicious mail traffic is usually driven by automatic processes* that send messages to a relatively large list of recipients per email ([1, 12, 15]).

In order to leverage file propagation characteristics for high-quality early detection, we need to be able to determine that a file is distributed in a suspicious manner as quickly as possible. Consequently, per-day measurements (as done in the DRVs shown in Fig. 1) are too coarse; we require time bins that are much shorter. Luckily, as shown by Fig. 3, the distinctive bursty behavior of email-distributed malware can be observed also when using per-hour measurements.

3.2 Time Propagation Features

Given a DRV representing an unknown file, what algorithm should we use to determine whether or not it corresponds to a webmail-attached malware? A widely-used method that may be employed for this purpose is the *k-Nearest Neighbors* (k-NN) classification algorithm. It de-



(a) Higher-volume malware. (b) Lower-volume malware. (c) Higher-volume legitimate. (d) Lower-volume legitimate.

Figure 2. Time-series of representative webmail-downloaded files, per day measurements. "Higher-volume" ("Lower-volume") implies at least (at most) 50 total downloads.

termines the class to which an unknown instance belongs based on a majority vote of its k nearest neighbors. More generally, the k nearest neighbors may be used for deriving features for classification. For instance, in our setting, a relatively high number of DRVs associated with files known to be malicious within the k-nearest neighbors of the DRV of an unknown file may indicate that it is malicious as well.

The k-NN algorithm requires a distance metric. The most common metric is the Euclidean distance. However, as can be seen in Figs. 2a-b and 3a-b, the DRVs of different webmail-attached malware are not aligned in terms of their scale and in terms of the number and times of their peaks. Euclidean distance is therefore not a good choice for quantifying the similarity of DRVs, as even two vectors that are identical up to a time-shift may be considered very distant from one another. What we require is a measure of distance that allows flexibility in the alignment of two time series, by permitting some level of shifting and/or stretching.

The well-known *Dynamic Time Warping* [4, 23] (DTW) distance measure does exactly that. Given two time-series TS_1, TS_2, of lengths respectively m and n, their DTW distance is defined by the following recursive expression, where $d(i,j)$ is a shorthand notation for $\left|TS_1[i] - TS_2[j]\right|$.

$$DTW(m,n) = \begin{cases} \infty & m{=}0 \text{ or } n{=}0, \\ d(m,n) + min\begin{pmatrix} DTW(m-1,n) \\ DTW(m,n-1) \\ DTW(m-1,n-1) \end{pmatrix} & \text{otherwise.} \end{cases} \quad (1)$$

We use the DTW measure in order to compute the distances between pairs of DRVs. We then use the k-NN algorithm to derive the following sets of time-propagation related features for each unknown file f. Each of the following two feature sets contains 4 features, corresponding to $k \in \{1, 5, 10, 20\}$.

k-Malicious Neighbours (k-MN): For each value of k, we find the set S of the k DRVs closest to f's DRV. Let $S' \subseteq S$ denote the set of those DRVs in S that correspond to files known to be malicious. The feature's value is the cardinality of S'. High feature values indicate that the manner in which the file propagates over time is suspicious.

Fig. 4a presents the distribution of the *k-MN* feature for $k = 20$. As expected, the DRVs of webmail-attached malware files tend to be relatively close to each other, whereas the DRVs of legitimate webmail attachments have significantly fewer malicious neighbors on average.

Closeness to malicious neighbours: Let S and S' be defined as above, we calculate the average DTW distance between f's DRV and the DRVs of S'. If $S' = \phi$, the value of this feature is set to ∞. Relatively low feature values indicate that the manner in which the file propagates over time is suspicious.

3.3 Modeling Space Propagation

Files are downloaded to machines from origin network locations. The network location is specified by the originating IP and originating URL report fields. A domain name and a host name are also derived from the URL.

The key principle underlying security mechanisms such as blacklists and reputation systems is that entities that behaved maliciously in the past are likely to do so again in the future. For instance, if an allegedly legitimate file f is now downloaded from an IP address from which malicious files were recently downloaded, then f is more likely to be malicious as well. This principle may be applied also to domain names and host names.

Community detection can be used for implicitly propagating and utilizing reputation data. Indeed, community detection was used in the past for identifying malicious groups of nodes, e.g. in email networks [5, 8, 17, 16]. We model the propagation of files from network locations to physical machines by constructing several graphs based on our data and applying the well-known Louvain method [6]. Given the community structure of each graph, we compute reputation-related features locally within each community.

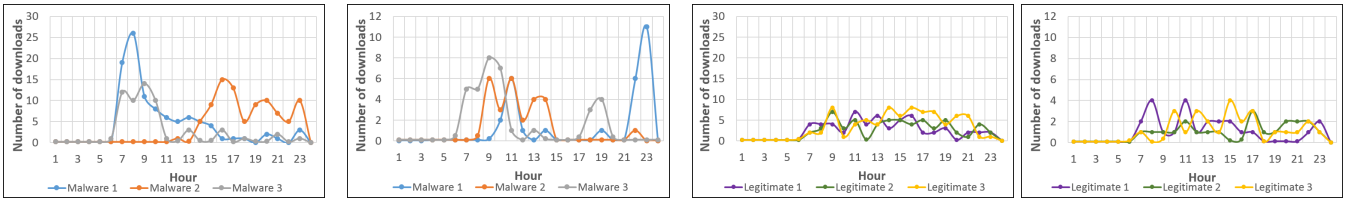
3.4 Space Propagation Features

We represent the manner in which webmail-attached files propagate in space using *file-location graphs*. A file-location graph is a bipartite graph that contains two sets of nodes. One set represents files that were sent as email attachments and downloaded from a webmail server to host machines, and the second set represents the origins (network locations) from which these files were downloaded or the machines to which they were downloaded. An undirected weighted edge (f, l) is added between each file f and each location l from/to which f was downloaded. When l represents a network location, the edge's weight is the number of different machines to which f was downloaded from l. When l represents a machine to which f was downloaded, the edge's weight is 1.

We generate four file-location graphs, each corresponding to a different type of network location: domain names, IPs, URLs, and host names. We also generate a fifth file-location graph for representing the relations between files and the machines to which they are downloaded.

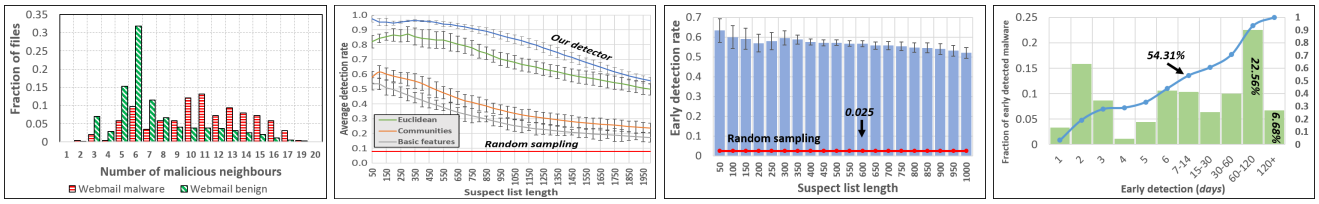
We apply the Louvain community detection algorithm to each of these graphs. For each file f, we then derive the following features based on the community C to which it belongs.

Community maliciousness. We define the maliciousness



(a) Higher-volume malware. (b) Lower-volume malware. (c) Higher-volume legitimate. (d) Lower-volume legitimate.

Figure 3. Time-series of representative webmail-downloaded files, per hour measurements.



(a) (b) (c) (d)

Figure 4. (a) k-MN distribution of webmail malware and benign files, for $k = 20$ (b) Detection rate for different feature sets used, for varying suspect list lengths (c) Early detection rate for varying suspect list lengths (d) Early detection period distribution for files recognized by VirusTotal.

of a community as the fraction of files in it that are known to be malicious out of its total size. We use the maliciousness of C as a feature for f.[3]

Number of malwares. The total number of files known to be malicious in C.

Number of files. The total number of files in C.

Number of malicious locations. A location $l \in C$ is considered malicious w.r.t. f if there is at least a single malicious file $f' \in C$, $f' \neq f$, that was downloaded from/to l. To compute this feature for file f, we count the number of locations that are malicious w.r.t. f.

Fraction of malicious locations. The fraction of locations in a file's community, from/to which a file known to be malicious was downloaded, out of the total number of locations in that community.

4. WEBMAIL-ATTACHED MALWARE DETECTION

We implemented a detector for malicious webmail attachments that uses the time and space propagation features described above, as well as the following *basic features*:

Number of distinct file names. The number of different file names (excluding file name extension) that are associated with the same file. Each of these names appeared in at least one report associated with the file's hash.

File type. The use of this feature is motivated by Table 2.

File reach. The use of this feature is motivated by Fig. 1d.

Number of countries. The number of different countries to which a file was downloaded.

Number of distinct origins. This is a set of 4 features. The number of distinct URLs, IPs, host names and domain names from which the file was downloaded.

Number of origin malware. For each of the 4 origin types, the number of malicious files downloaded from the same origins from which the file was downloaded.

Rate of origin malware. For each of the 4 origin types, the rate of malicious files downloaded from the same origins from which the file was downloaded.

Our detector employs a supervised classification machine learning algorithm that, given a set of labelled training instances (a.k.a. training examples) to learn from, produces a machine learning model that predicts the class to which an unknown instance belongs. In our setting, each instance corresponds to a file that may be classified as either malicious or benign. An instance is represented by a feature vector, the components of which store the values of the features computed for the file. For the training examples, we also require ground truth labels (or simply labels), designating to which class each of them belongs. We now describe how ground truth labels are obtained. This is followed by a description of the tests we conduct to evaluate our detector and a discussion of their results and of deployment considerations.

4.1 File Ground Truth

Our ground truth data is composed of sets of file labels that we received from Microsoft and from data that we collected from VirusTotal.

Microsoft. We received from Microsoft two sets of labels. *Labels set 1* contains the hash values of files that appear in our data set and were known to be malicious at some point *before* the data set's logging period. *Labels set 2* contains the hash values of files that appear in our data set and were known to be malicious at some point *after* the data set's logging period. Thus, labels set 2 is a superset of labels set 1. A malware family ID is associated with each hash value in labels set 2.

VirusTotal. VirusTotal is a free online service provided by Google. As of this writing, 54 antivirus products are integrated in VT. It allows uploading a suspicious file sample to the VT site, applies integrated AV scanners to it, and reports back the aggregated results. It is also possible to

[3]We exclude f itself from the computation of this feature in order to avoid a feature bias. This is done also in the computation of the "number of malwares" feature.

Table 3. A comparison of 10-fold cross-validation results for several ensemble classifiers.

Classifier	FPR	Precision	Recall	F-Measure	AUC
Rotation Forest	0.017	0.978	0.968	0.973	0.995
Random Forest	0.034	0.966	0.960	0.963	0.990
AdaBoost	0.037	0.963	0.969	0.966	0.990
Decorate	0.031	0.969	0.961	0.965	0.989

search VT for a specific file hash value and to retrieve its report, if it exists. A report includes the *first seen* timestamp, specifying the first time the sample was uploaded to the service, and the number of AV scanners that identified the sample as malicious.

We used labels set 2 in Section 2. We use data obtained by VT queries, as well as the label sets we received from Microsoft, in order to conduct the experimental evaluation of our detector, which we describe next.

4.2 Performance Evaluation

We conducted three experiments for evaluating our detector's performance, in increasing order of the difficulty of the challenge they pose to it. In all these experiments, we use a training set that contains both malicious and benign webmail files. Since there are many more benign files in our data set than there are malicious files (see Table 1), we use *undersampling* [22], a standard technique for avoiding a bias of the classifier towards the more frequent population, to select an equal number of positive and negative training examples.[4] We now describe the design of these experiments and analyze their results.

Cross-Validation

Using the ground truth provided in labels set 1, we created, for every day d in the data set's logging period, a balanced training set composed as follows. First, we added all the webmail files that were downloaded in day d and were tagged as malicious by labels set 1. To obtain a balanced training set, we added an equal number of randomly selected webmail files downloaded in day d that were not tagged as malicious in labels set 1, labelling them as benign. We then applied 10-fold cross-validation to the training set.

We experimented with the following well-known ensemble classifier algorithms in WEKA [20]: Rotation Forest, Random Forest, AdaBoost and Decorate. Table 3 presents their average daily results.[5] Rotation Forest (using an ensemble of 50 trees) consistently provided the best results, with true positive rate (a.k.a. recall) of 0.968 and false positive rate of 0.017. We therefore employed Rotation Forest also in the rest of our experiments.

New Malware Detection

The cross-validation experiment estimates our detector's performance in terms of detecting malicious files that were already known prior to the data set's logging period. In the detection rate experiment we describe now, we evaluate our detector's ability to detect *new malware*. These are malicious files whose labels do not appear in labels set 1.

[4]We experimented using a few ratios between positive and negative examples without observing significant differences in performance.

[5]There was only small variance in daily results.

The experiment was conducted by creating, for every day d in the data set's logging period, a balanced training set composed in the same manner done for the cross-validation experiment. After training the rotation forest algorithm using this set, we obtain a *detection model for day d*.

We apply this model to all the webmail files that were downloaded on day d and were not included in the training set. When applied to file f, the model returns a *confidence score* value $\in [0, 1]$, quantifying the level of suspicion that the file is malicious (the higher the score, the higher the suspicion that the file is malicious). Our detector outputs a *top suspects list* whose length is an input parameter, sorted in decreasing order of the scores.

We evaluate the quality of a top suspects list by counting the number of *detections* in it. We count a file as a detection, if at least one of the following holds: 1) the file's hash appears in labels set 2, or 2) the file exists in VT and is flagged as malicious by at least 3 AV scanners. The list's precision is the rate of detections in it.

Fig. 4b presents the average daily detection rate, as a function of the suspects list length. Each curve corresponds to a different combination of feature sets. The short vertical bars centered around each curve show the range of daily detection rates within one standard deviation in each direction.

Our detector uses the full set of features (basic + time propagation + space propagation) and obtains the best performance. The daily average number of detections in the list of top 50 suspects is 0.975 and the rate remains above 90% for all lists of length up to 750. Detection rate decreases as the suspects list grows longer, because suspects with lower confidence scores are being added. Nevertheless, the average detection rate is above 50% even for lists of length 2000.

When the time propagation features are computed by using Euclidean distance instead of DTW, detection rate decreases for all lengths by up to 20%. This establishes empirically that DTW is significantly superior to Euclidean distance for quantifying the level of similarity between download rate vectors.

The curve labelled by "communities" presents performance when time propagation features are entirely removed and only the community and basic features are used. Detection rates decrease drastically in comparison with Euclidean distance by up to more than 50%. This shows the large contribution of time propagation features, even when they are computed by using Euclidean distance, which, as we saw, is not the optimal distance measure in our setting.

Detection rates when using basic features only start at around 55% for short lists and decrease with list length to approximately 17% for suspects list of length 2000. This is a significant drop in performance. Although our detector uses a few basic features that take into consideration origin reputation (such as the number and rate of origin malware), they are computed globally rather than locally within a file's community. These results highlight the contribution of community-based features.

The red horizontal curve labelled by "random sampling", fixed at detection rate 0.079, is the expected "detection rate" when randomly selecting suspect files. To compute this baseline rate, for each day d, we count the number of files that 1) appear in labels set 2, 2) do not appear in labels set 1, and 3) were downloaded in day d. Let q denote this number of *potential detections for day d*. We then divide q by the number of distinct files downloaded in day d (excluding

those in the training set) to obtain the probability p_d of randomly picking a malicious file out of the files seen in day d. Finally, we average p_d over the days of the logging period.

Early Detection of New Malware

In this experiment, we estimate our detector's ability to detect new malware earlier than their first submission to VirusTotal. The training set and model are generated in exactly the same manner as for the detection experiment.

We evaluate the quality of a top suspects list for day d by counting the number of *early detections* in it. We count a file as an early detection, if the file is detected and if, in addition, one of the following holds: 1) the file does not exist in VT, or 2) the file exists in VT, but its *first seen* timestamp is later than day d. We note that this is a lower bound on our detector's rate of early detections, because a VT file sample may be flagged by AV scanners only long after it is uploaded to VT.

Fig. 4c presents the average daily early detection rate, as a function of the suspects list length. The average rate of early detection is approximately 63% for length-50 lists and decreases mildly as the suspects list grows longer. The average early detection rate remains above 52% even for lists of length 1000.

The red horizontal line at the bottom of the figure, fixed at a rate of 2.5%, is the expected "early detection rate" when randomly selecting suspect files. The baseline for day d is computed similarly to the random baseline of the detection experiment, and the denominator is exactly the same. The numerator, however, is smaller and counts the number of files that 1) appear in labels set 2 and/or are flagged in VT by at least 3 scanners, 2) do not appear in labels set 1, 3) were downloaded in day d, and 4) either do not have a VT report or their *first seen* flag is later than day d. Other than that, the computation is the same.

On average, more than 75% of our daily early detections are not recognized by VirusTotal as of the time of this writing. Figure 4d presents the distribution of the early detection period of our detector on those files that *are* recognized. The early detection period as compared with VT for almost half these files is two weeks or more, and almost 7% of them are detected by us at least 4 months earlier.

5. RELATED WORK

To the best of our knowledge, empirical research on the detection of malicious email attachments based on their propagation patterns, using real-world data, is virtually nonexistent; prior works on this topic rely on simulations and theoretical models and we survey a few of them here.

Chen et al. [7] model the propagation of *topological malware* that spreads based on topology information (e.g. email address books). They use the independent and Markov models to predict malware propagation in arbitrary topologies.

Zou et al. [35] present an email worm simulation model that takes into consideration factors affected by email users' behavior, such as email checking time and the probability of opening email attachments. In later work along the same lines, Wen et al. [33] propose a mathematical model for the propagation of email-based malware. They evaluate the predictions of their model by comparing them with simulation results. Gao et al. [18] propose an interactive model for characterizing malware propagation in email networks with various topologies and consider several immunization strategies, in which some fraction of the network nodes are immunized (protected) so they cannot be infected.

Another line of research relevant to our work, more practical in nature, attempts to overcome the shortcomings of static detection by analyzing malware's network-level behavior. Vadrevu et al. [32] present AMICO, a system for measuring and detecting malware downloads. The underlying intuition is that malware behaves in a more "dynamic" manner in order to evade AV detection. Invernizzi et al. [14] present Nacza, a system for detecting HTTP requests that download malicious files, designed to operate in large-scale networks, such as ISPs. Nacza correlates multiple malware downloads pertaining to a specific spam campaign, attempting to identify malicious connection behavior, such as file mutations (a.k.a. server-side polymorphism) and domain fluxing. Kwon et al. [21] focus on the analysis and detection of malicious downloaders. They introduce the *downloader-graph abstraction*, which captures the download activity on end hosts. Their analysis shows that there are significant differences between the graphs of malicious and benign software. Nelms et al. [26] developed a technique to passively trace back and automatically label network events that precede different types of malware downloads.

Rahbarinia et al. [28] present "Mastino", a defense system that detects *download events*. The proposed method consists of two classifiers, one dedicated to distinguishing between malicious and benign URLs, and another dedicated to detecting malicious downloaded files.

Some related studies [25, 31] use the Belief Propagation (BP) algorithm in order to infer a file's classification based on the topology of bipartite association graphs. For instance, Tamersoy et al. [31] identify close relationships between files that co-occur often on the same machine.

Malware infections can also be detected by analysing the traffic generated by hosts. For instance, Oprea et al. [27] exploit the relationship between suspicious external destinations. They propose a graph-theoretic framework based on belief propagation to identify small communities of related domains that are indicative of early-stage malware infections.

6. CONCLUSIONS

We conduct the first comprehensive study of malicious webmail attachments and their unique characteristics. We show that the webmail attachments and the general web-downloaded files populations are very different from one another in terms of file type, reach, malware classes, and malware family distributions. We then find that the propagation process of malicious webmail attachments is characterized by short bursts in which a file is downloaded to many machines, interspersed by extended periods of low activity.

We use dynamic time warping for deriving features, based on time-series representation of file download rates, for identifying this propagation process. In addition, we derive reputation-based features from local community properties.

We implement a detector for malicious webmail attachments that uses our propagation-based features, as well as a set of additional basic features. Our experimental evaluation shows that it achieves high accuracy in cross-validation tests and also achieves high-quality detection of new malware. Moreover, it is able to detect new webmail malware significantly earlier than their first submission to VirusTotal.

7. REFERENCES

[1] Cryptolocker alert: Millions in the UK targeted in mass spam campaign. http://www.symantec.com/connect/blogs/locky-ransomware-aggressive-hunt-victims.

[2] github webmail-domains list. https://github.com/tarr11/Webmail-Domains.

[3] List of domains that offer freemail accounts. http://www.0spam.com/freemail/.

[4] D J Berndt and J Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370, 1994.

[5] S Y Bhat and M Abulaish. Community-based features for identifying spammers in online social networks. In *Proceedings of ASONAM 2013*, pages 100–107, 2013.

[6] V D Blondel, J-L Guillaume, R Lambiotte, and E Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

[7] Z Chen and C Ji. Spatial-temporal modeling of malware propagation in networks. *Neural Networks, IEEE Transactions on*, 16(5):1291–1303, 2005.

[8] Y Cohen and D Hendler. Birds of a feather flock together: The accidental communities of spammers. In *ASONAM 2015*, pages 986–993.

[9] Microsoft corporation. Security intelligence report. volume 20, July-December, 2015. Technical report.

[10] Symantec corporation. Dridex: Tidal waves of spam pushing dangerous financial trojan, December, 2015. Technical report.

[11] Symantec corporation. Internet security threat report. volume 21, April, 2015. Technical report.

[12] S Dinh, T Azeb, F Fortin, D Mouheb, and M Debbabi. Spam campaign detection, analysis, and investigation. *Digital Investigation*, 12:S12–S21, 2015.

[13] O Engel. Clusters, recipients and reciprocity: Extracting more value from email communication networks. *Procedia-Social and Behavioral Sciences*, 10:172–182, 2011.

[14] Invernizzi et al. Nazca: Detecting malware distribution in large-scale networks. In *NDSS*, volume 14, pages 23–26, 2014.

[15] Sheikhalishahi et al. Digital waste sorting: A goal-based, self-learning approach to label spam email campaigns. In *Security and Trust Management*, pages 3–19. Springer, 2015.

[16] Xu et al. Revealing social networks of spammers through spectral clustering. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[17] M Fire, G Katz, and Y Elovici. Strangers intrusion detection-detecting spammers and fake proles in social networks based on topology anomalies. *HUMAN*, 1(1):pp–26, 2012.

[18] C Gao, J Liu, and N Zhong. Network immunization and virus propagation in email networks: experimental evaluation and analysis. *Knowledge and information systems*, 27(2):253–279, 2011.

[19] The Radicati Group. Email statistics report, 2015. Technical report.

[20] G Holmes, A Donkin, and I H Witten. Weka: A machine learning workbench. In *Proc. of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems*, pages 357–361, 1994.

[21] B J Kwon, J Mondal, J Jang, L Bilge, and T Dumitras. The dropper effect: Insights into malware distribution with downloader graph analytics. In *CCS 2015*, pages 1118–1129.

[22] X-Y Liu, J Wu, and Z-H Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):539–550, 2009.

[23] M Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[24] M Musale, T H Austin, and M Stamp. Hunting for metamorphic javascript malware. *Journal of Computer Virology and Hacking Techniques*, 11(2):89–102, 2015.

[25] C Nachenberg, J Wilhelm, A Wright, and C Faloutsos. Polonium: Tera-scale graph mining and inference for malware detection. 2011.

[26] T Nelms, R Perdisci, M Antonakakis, and M Ahamad. Webwitness: investigating, categorizing, and mitigating malware download paths. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1025–1040.

[27] A Oprea, Z Li, T-F Yen, S H Chin, and S Alrwais. Detection of early-stage enterprise infection by mining large-scale log data. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 45–56, 2015.

[28] B Rahbarinia, M Balduzzi, and R Perdisci. Real-time detection of malware downloads via large-scale url-> file-> machine graph mining. In *Proc. of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 783–794, 2016.

[29] Z Sofershtein and S Cohen. Predicting email recipients. In *ASONAM 2015*, pages 761–764.

[30] Litmus Software. State of mail report 2016. Technical report.

[31] A Tamersoy, K Roundy, and D H Chau. Guilt by association: large scale malware detection by mining file-relation graphs. In *Proc. of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1524–1533, 2014.

[32] P Vadrevu, B Rahbarinia, R Perdisci, K Li, and M Antonakakis. Measuring and detecting malware downloads in live network traffic. In *ESORICS 2013*, pages 556–573. 2013.

[33] S Wen, W Zhou, J Zhang, Y Xiang, W Zhou, W Jia, and C C Zou. Modeling and analysis on the propagation dynamics of modern email malware. *IEEE Transactions on Dependable and Secure Computing*, 11(4):361–374, 2014.

[34] I You and K Yim. Malware obfuscation techniques: A brief survey. In *2010 International conference on broadband, wireless computing, communication and applications*, pages 297–300.

[35] C C Zou, D Towsley, and W Gong. Modeling and simulation study of the propagation and defense of internet e-mail worms. *IEEE Trans. on Dependable and Secure Computing*, 4(2):105–118, 2007.