"Infect-me-not": A User-centric and Site-centric Study of web-based malware

Huy Hang*, Adnan Bashir†, Michalis Faloutsos*, Christos Faloutsos‡, and Tudor Dumitras§

* University of California, Riverside
Department of Computer Science and Engineering
Riverside, CA 92521
{hangh, michalis}@cs.ucr.edu

‡ Carnegie Mellon University
Department of Computer Science
Pittsburg, PA 15213
christos@cs.cmu.edu

† University of New Mexico, Albuquerque
Department of Computer Science
Albuquerque, NM 87131
abashir@cs.unm.edu

§ University of Maryland, College Park
ECE Department
College Park, MD 20742
tdumitra@umiacs.umd.edu

Abstract

Malware authors have been using websites to distribute their products as a way to evade spam filters and classic anti-virus engines. Yet there has been relatively little work in modeling the behaviors and temporal properties of websites, as most research focuses on detecting whether a website distributes malware. In this paper we ask: How does web-based malware spread? We conduct an extensive study and follow a website-centric and user-centric point of view. We collect data from four online databases, including Symantec's WINE Project, for a total of more than 600K malicious URLs and over 500K users. First, we find that legitimate but compromised websites constitute 33.1% of the malicious websites in our dataset. In order to conduct this study, we develop a classifier to distinguish between compromised vs. malicious websites with an accuracy of 95.3%, which could be of interest to studies on website profiling. Second, we find that malicious URLs can be surprisingly long-lived, with 10% of malicious sites staying active for three months or more. Third, we observe that a significant number of URLs exhibit the same temporal pattern that suggests a flush-crowd behavior, inflicting most of their damage during the first few days of appearance. Finally, the distribution of the visits to malicious sites per user is skewed, with 1.4% of users visiting more than 10 malicious sites in 8 months. Our study is a first step towards modeling web-based malware propagation as a network-wide phenomenon and enabling researchers to develop realistic assumptions and models.

I. Introduction

Distributing malware indirectly via web-pages has become a very popular way for spreading malware in the last 8 years. In 2012, Google reported that they identify 9,500 of malware-spreading websites each day [1]. These websites infect their visitors, but we can identify two different types: (a) the **born-malicious**, which are registered and operated by the malicious entities, and (b) the **compromised**, legitimate websites infiltrated by hackers and injected with malware.

ISBN 978-3-901882-83-8

"How does web-based malware spread?" is the key question that motivates this work. We consider a site-centric and a user-centric point of view: (a) what is the behavior and the lifecycle of the website that spreads malware, and (b) what is the behavior of the users that visit such websites. Our goal is three-fold: (a) investigate the composition of the websites to find out how many of them are born-malicious and how many compromised, (b) understand the life of a malicious URL and its impact, and (c) identify patterns in the way users visit malicious URLs. For the remainder of this paper, we use the term malware to refer to web-based malware. In our work, we focus on the spatiotemporal patterns of how malware is distributed from malicious sites to users.

Most previous work has focused more on identifying malicious websites, and less on their propagation patterns. In more detail, we identify four areas of focus in the literature: (a) the identification of websites vulnerability to infiltration, (b) the detection of websites actively distributing malware, (c) the study of the ecosystem and the techniques used by hackers, and (d) the analysis of the web-based malware themselves. We describe research efforts in these areas in section V.

Our key contribution is an extensive study of user exposure to web-based malware following both a site-centric and user-centric point of view. We use two data sets: (a) D_{ODB}, with roughly 66K malicious URLs collected from four online databases between December 2013 to September 2014, and (b) D_{WINE}, which captures visits to malicious websites from roughly 530K users from January 2011 to August 2011 collected by the Symantec's WINE Project. Note that Symantec's data captures the exposure of the users to malware as seen by its anti-virus products, as we explain in section II.

Our work can be summarized into the following major observations.

a) **Compromised websites play a significant role in malware dissemination.** We find that among all the domains in our D_{ODB} dataset, 33.1% of them belong to compromised websites. For our study, we developed a

Machine Learning-based method to distinguish compromised websites from born-malicious sites. Our approach exhibits a 95.3% accuracy. We want to stress that the ML method we developed is strictly for a forensics purpose: we want to measure how prevalent the phenomenon of compromised websites is and raise awareness about the danger that they may pose. We did not intend the method to be a detection tool for compromised sites.

b) **A malicious URL often distributes many different malware binaries but each malicious binary is typically distributed by one URL.** We find that 33% of the URLs with at least 5 visits in D_{WINE} distribute two or more different binaries (different MD5 hash values). This percentage increases to 46% among all websites with more than 20 visitors. These website are either: (a) distributing completely different malware, (b) using polymorphism to distribute the mutated versions of the same malware to escape detection. In contrast, most malicious binaries (94.6%) are distributed by one URL in our data set.

c) **Most malicious URLs are short-lived, but 10% of them are active for more than three months.** Although 71.6% of URLs in D_{WINE} appear for only one day during the 8 months, roughly 10% stay active for at least three months and a much smaller number have been active for four years. This suggests there may not be an efficient technical and/or legal process to clean up or take down a malicious website.

d) **The "Space-needle" pattern: Many URLs exhibit the same bursty temporal pattern aligned with a campaign-like behavior.** Here, we focus on Highly Active URLs in D_{WINE}, which have lifespans of at least 30 days and have at least 100 visitors each. We find that the time series of the visits to 45.6% of those URLs follow a bursty pattern, which we refer to as "space-needle" due to its shape. URLs following this pattern usually peak within the first two days of their life, and the maximum number of daily visits is at least an order of magnitude larger than the median, as we discuss in section IV.

e) **The distribution of the visits to malicious sites per user is skewed and can be described by a power law of exponent $-\frac{1}{2}$.** A small percentage of users in D_{WINE} are highly susceptible to visiting malicious URLs. For example, we find that 1.4% of all users (close to 7500 users!) in our data set visited at least 10 malicious URLs during the 8 months.

Data Archive and Acknowledgment. The data is available for follow-up research as reference data set WINE-2014-002 in Symantec's WINE repository. We are grateful to Drs. Matthew Elder and Daniel Marino of Symantec Research Lab for their support and feedback.

II. Our Data Sets and Background

We present the sources of information and data sets that we use in our work.

A. Sources for URL characterization

We rely mostly on two sources of information regarding the status of a URL. First, VirusTotal is a popular online service where a user can submit a binary or a URL or a domain so it can be scanned by at least 50 anti-virus engines. Once the scan concludes, the user can retrieve a report that shows, in the case of a domain, the number of AV engines that considered the domain to be of a malicious website. We call this value the VirusTotal Malicious Score of a domain. Second, the Web of Trust (WoT) Reputation Score is a numerical value between 0 and 100, inclusively, given to a website by WoT [2], which relies on its user community to rate the websites the users came across. The higher score a website has, the more trustworthy. A poor reputation score does not imply a website is malicious.

B. Our data sets

We use the following sources to build our data sets.

1) Online databases: We collected malicious URLs from four different online databases: Cybercrime Tracker [3], Malc0de [4], Malware Domain List [5], and VX Vault [6]. These online databases are maintained by communities and publish new malicious URLs on a regular basis. We began collecting the URLs in March 2014 and continued to do so every day until September 2014. This data set, which we call D_{ODB} from this point onward, will be used to build a classifier to distinguish born-malicious from compromised websites.

	URLs	Domains	Clients	MD5s
O.D.B.	71,542	8,724	-	-
WINE	626,472	106,026	530,061	504,324

TABLE I
Data from Online Databases (O.D.B.) and WINE

This dataset is used exclusively to train and test our classifier of born-malicious and compromised websites, as will be shown in section III.

2) Symantec's WINE data: Symantec's Worldwide Intelligence Network Environment (WINE) [7] is a massive corpus of telemetry data sampled from more than 120 million machines, both enterprise and consumer, and made available to the research community. This dataset was also used in the analysis of zero-day attacks in [8] as well as the study to expose the change in cyber threat landscape and the emergence of new attack surfaces [9].

The WINE database is divided into five datasets, each containing data from different aspects of the data collection process. The data that we collected from WINE belong to two specific datasets:

1) AV Telemetry: data collected from all clients any time a Symantec AV product detected that a *malicious* binary **executable** was downloaded.
2) Binary Reputation: data collected on binary **executables** downloaded by users in Symantec's reputation-based security program. Even though this dataset contains information on both malicious and benign binaries, it does

not contain information on whether a specific binary is malicious.

We use the **machine ID identifier** to distinguish users. This is a unique ID that each Symantec software installation. This way, we eliminate the "noise" that can be introduced by using IP addresses, which are dynamically assigned and often obfuscated by Network Address Translators (NATs).

C. Modeling the exposure to malware

The WINE data we collected focuses exclusively on *visits to malicious URLs*. Every entry in the dataset represents a report any time a user **downloaded** a malicious binary from a URL. However, we do not make a claim as to whether the user was infected or not. In fact, we believe that the malicious binaries were detected and the users would have been protected, unless they explicitly overrode the antivirus warning.

Each data point in our data set, which we will call D_{WINE}, contains: (a) the timestamp of the receipt of the report at a Symantec server, (b) the URL from which the binary was downloaded, (c) the MD5 hash of the binary, and (d) the `ID` of the client machine.

We begin with collecting the information about malicious URLs from AVs Telemetry from January to August 2011. We then correlate with Binary Reputation to obtain the information about the URLs from *before* Symantec determines that the URLs were distributing malware so that we get the complete history about each URL (including which binaries they distributed and who downloaded from them).

We use this dataset to conduct analysis of the spatiotemporal characteristics and malicious websites and study the behaviors of the users who visit the malicious sites, the details of which will be shown in section IV.

Representativeness. The classic question for any real measurement data is how representative is the data. We rely on users that have Symantec AV products installed, and this may be introducing some bias, though we don't have any reason to believe that users of other anti-virus solutions will have an i ntrinsically different behavior. At the same time, D_{WINE} is data collected from roughly 550K users spanning eight months and WINE is drawn from more than 120 million machines worldwide, so our dataset consist of a reasonably wide cross-section of users.

III. Provenance of malicious websites

Given a malicious website, we would like to determine if it is born-malicious or compromised (which we defined in the introduction). We present our Machine Learning-based method that accomplishes this with high accuracy. Note that we use the dataset D_{ODB} exclusively in building the classifier and performing testing.

Why do we want to study malware-spreading websites, after they have been identified as such? There are two reasons. First, compromised websites are not very well studied, to the best of our knowledge despite the fact that alarms had been raised about them. In their 2014 Threat Report [10], Symantec discovered that one in eight legitimate websites have unpatched critical vulnerabilities, making them ripe for an attack. Second, hackers try to infect their victims by abusing the *trust* that legitimate sites have established over time instead of getting around domain or IP blacklists by creating new websites or constantly switching to new IPs via fast-fluxing.

The proposed technique is arguably the first that focuses on this problem. As such, we would offer it as a publicly-available tool for studying the provenance of websites (whether they are created for malicious purposes or hijacked) and providing a first-level forensics capability. Note that the course of action for stopping the spread of malware depends on this classificiation. In the case of a born-malicious site, the site needs to be taken down and possibly have the hosting entity notified.

A. Building the classifier.

We present the steps that we took for developing our classifier.

1) Data Preprocessing: Even though the D_{ODB} dataset includes 8,724 domains in total, we run the classifier on only 3,975 of them. We do not include in our study the domains that belonged to any of the following categories:

a) Domains not resolving to IPs. These 1,550 domains no longer provide valid DNS records, because there was a time gap between when the domain was reported as malicious and when we attempted to crawl them. A close examination of such domains shows most have poor reputation scores and created recently. It is likely these domains were deactivated for distributing malware.
b) Domains returning `40X` codes or no content.
c) Domains belonging to known Content Delivery Networks, file-sharing sites (e.g. `mediafire`) and websites hosting free software (e.g. `softpedia`). By nature, these websites allow the posting of user content, which can often point to malicious websites or even contain malware.

2) Our training and testing data: From the remaining 3,975 domains (which we call D_{classify}), we randomly selected 609 domains and split them into two used for training and then testing our binary classifier:

a) $D_{\textbf{train}}$ has 200 domains, 139 labeled as compromised and 61 born-malicious
b) $D_{\textbf{test}}$ has 409 domains, 280 labeled compromised and 129 malicious.

We label the domains manually and carefully: we visit each domain in a browser in a virtual machine, carefully examine each landing page we come across, and explore each link on the landing page to see if we can reach other pages that may contain legitimate content. The richer the content is, the more confident we are that the website is a legitimate website that was compromised.

3) Features: We first present the features that our classifier uses, and later we discuss other features that we considered but did not ended up using. We use the following features:

a) Number of URLs embedded in the landing page.

b) Number of images on the landing page.
c) Age (days) of domain since registration.
d) Web of Trust's reputation score for domain.
e) VirusTotal's malicious score for domain.

Intuitively, a legitimate but compromised website tends to bear the following characteristics:

a) Its landing page is much more complex than that of a website created to deliver malware, meaning that it has richer content, more hyperlinks that lead to its other pages, and more images in general.
b) It has been around for longer than a born-malicious website, as malware authors stage new websites very often, knowing that it is likely that a malicious site could be taken down or blacklisted quickly. The older a website is, the more trustworthy it is.
c) Compromised websites have relatively higher Web of Trust's score.

These aforementioned characteristics are covered by features (1), (2), and (3), which are not enough, as using only these three means that we run the risk of classifying a newer, simple website as malicious or classifying an older and more complex malicious site as benign. To avoid this, we also make use of features (4) and (5), which allows the classifier to factor into its classification decision how many anti-virus engines consider the site to be malicious and how good a reputation a site may have.

To obtain relevant statistics for each domain in D_{ODB}, we created an automatic web crawler using Selenium [11] and connected the Selenium-driven browser to a proxy to keep track of every image downloaded by the browser.

4) Other features: We have also considered other features to use in our classifier such as: the total number of pages hosted by a website, the total number of images and links that could be found on all of the pages, etc. We opted to not use these features. First, we want to create a light-weight classifier, so we avoid features that intense in computation and resource utilization. Using a feature such as the total number of pages on a website would mean that our crawler would have to crawl the entire website and explore every single link that can be discovered while making sure that the crawler would not follow a link that leads out of the website. Further, as we show below, we were able to achieve good classification accuracy using the selected light-weight features.

We also considered using IP blacklists as an additional means of pre-filtering to quickly identify a malicious website. We also decided against using the black-lists because a single IP address may be home to multiple websites and we run the risk of labeling as malicious a benign website hosted on the same server.

B. Training the classifier.

We use D_{train} to train our classifier, and we select the Random Tree method, because it gives the best performance among all others included in the WEKA Machine Learning framework [12]. We tried to create single-feature classifiers to test the accuracy of each feature but **none** of the classifiers exceeded 85% in accuracy (as seen in Table II) when applied on D_{train}, where we define accuracy as the ratio of the number of correctly labeled domains and the total number of domains. Applying a classifier built from all features on D_{train} yields no misclassified instances. Note that we define accuracy as the number of correctly labeled instances over the total number of instances.

Feature name	Accuracy
Number of URLs on landing page	82.3%
Number of images on landing page	83.7%
Age since registration	84.7%
Web of Trust score	77.4%
Virus Total score	73.5%

TABLE II
ACCURACY USING A CLASSIFIER WITH ONLY ONE FEATURE

We also performed **cross-validation** of our training set using the 10-fold cross validation function of the WEKA suite, and the result (using all five features) is just as good in that there is no instance misclassified.

C. Testing the classifier.

For testing, we use the D_{test} dataset.

1) Achieving a classification accuracy of 95.3%: We find that the number of correctly classified instances is 390 out of 409. We investigated the misclassifications to understand the limitations of our approach. Among the 19 misclassified domains, we find: 9 compromised that were labeled as born-malicious and 10 malicious domains that were labeled as compromised. Our careful investigation shows that some misclassifications happened due to several reasons: (a) the domains were hosted by dynamic DNS services, so the age values reported, which are very high, are of the DNS services themselves, (b) some compromised websites have extremely simple home pages with few images and embedded URLs.

2) 33.1% of malicious domains are compromised: With our classifier, we classify all the domains in D_{classify}. We find 2,885 compromised and 1,090 born-malicious. This means that roughly 33.1% of the domains from D_{ODB} are benign websites infiltrated by hackers and used to distribute malware. We will revisit the phenomenon of compromised domains again in the next section.

IV. PROFILING MALICIOUS URLS & THEIR VISITORS

We present our findings on the temporal properties of malicious URLs and the browsing behavior of the users.

A. Malicious URLs distribute many different malware binaries but each binaries is usually distributed from one URL.

In D_{WINE}, we observed many instances where the same URL yielded binaries with different MD5 hashes, often within the same hour. This phenomenon can be observed in Figure 1, where the each data point represents a single binary executable (X-axis) and the number of distinct URLs (Y-axis) from which it can be downloaded.

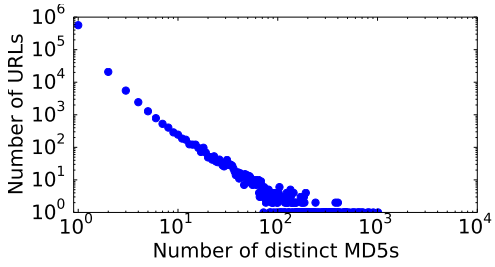

Fig. 1. Distribution of the number of unique MD5 hashes seen per URL. For example, the datapoint (x, y) indicates that there are x binaries each of which can be downloaded from exactly y URLs.

In this paper, we will call this phenomenon *URL-centric polymorphism*. Note that malware polymorphism, in general, refers to distributing the same malware in many different versions. We choose a different name to stress that we only focus on MD5-hash-based similarity: we were not given access to the malware itself to determine if two MD5 hashes correspond to the same malware.

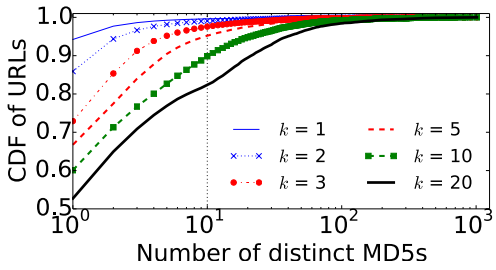

Fig. 2. URL-centric polymorphism observed more clearly on highly-active URLs.

1) One website many MD5s: In Figure 2, we show the distribution of unique MD5 hashes for each URL. Each curve, represented by a k value, shows the distribution of MD5 count per URL where the URLs have at least k visitors. We can see for 94% of URLs, each is associated with only one MD5 ($k = 1$). Most of these low-access URLs have one or two visitors, which makes it difficult to observe MD5 variations. The polymorphism becomes more evident for URLs with more visitors. For the URLs with at least $k = 5$ visitors, 33% of them distributed more than one binary, but this number increases to 46% when for URLs with at least $k = 20$ visitors.

2) Each MD5 is typically distributed from one website.: Reversing the question, we examine how many websites distribute the same MD5 malicious binary in D_{WINE}. Towards this goal, we look at each MD5 hash value in our D_{WINE} dataset and count the number of distinct malicious URLs from where the binary with the MD5 was observed to be downloaded. In Figure 3, we plot the distribution of MD5s according to the number of URLs that distribute them. We observe that 92.2% (more than 464K binary executables) are distributed by only one single URL.

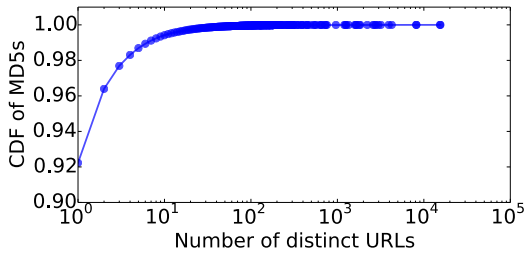

Fig. 3. Distribution of the number MD5s that are distributed by a certain number of distinct URLs (x-axis).

3) A few MD5s are widely distributed by more than 100 URLs.: There are some binary executables that appeared on more than a hundred URLs, which we did not expect. To investigate, we randomly picked 600 of these binaries and examined the related URLs. We observed:

1) The majority of these binaries (76.2%) appeared on multiple born-malicious domains, which seemed "disposable": the domains typically had random sequences of characters, pointing to an *automated* name-generation process.
2) A relatively small percentage (14.5%) of these binaries appeared on multiple domains that belong to popular file-sharing websites or well-known software distributors. This means malware authors rely on the many free file-sharing services or embed malicious code into popular and often pirated software to distribute the files. Note that although we filtered out such domains from D_{ODB} (as noted in section III-A1, we did not do so for D_{WINE} because our goal is to study how malicious binaries are distributed at large.
3) 9.3% appeared on what seem like compromised sites, many of them active and containing legitimate content. These binaries were distributed by URLs that have similar structure. For example, one such binary was distributed by URLs of the form: `http://{D}/images/facebook-pic-{X}.exe`, where `{D}` represents different domain names and `{X}` a sequence of random digits. This suggests that the sites were compromised: (a) through the use of the same hacking toolkit, and/or (b) by the same hacker.

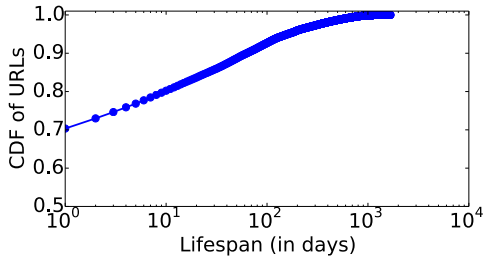

Fig. 4. Cumulative distribution of URLs according to their lifespan (x-axis)

B. Malicious URLs exihibit short lifespans and the number of visitors who visit them follow a skewed distribution

1) Most malicious URLs have short lifespan, but a small percentage live for more than three months: In Figure 4, we study the distribution of the lifespan of websites. We find that 70.6% of all malicious URLs are what we call *single-day URLs* as they appear for only one day in our dataset. Surprisingly, 10% of these websites managed to stay "alive" and actively distributed malware for more than three months and there are 194 malicious URLs that were around for four years. Furthermore, a small percentage (2,427 URLs, making up 0.4% of all URLs) attracted at least a hundred users during their lifespan.

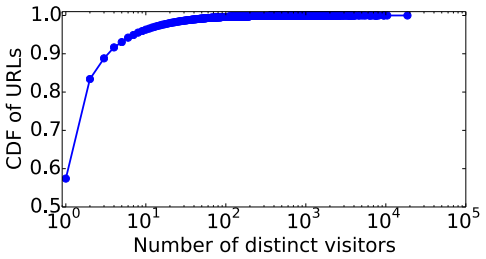

Fig. 5. Cumulative distribution of URLs according to their visitor count (x-axis)

2) The number of visitors per URL follows a skewed distribution: In Figure 5, we plot the distribution of unique visitors per URL. We find that this distribution is highly skewed with 57.4% of the URLs having one visitor, while 11.2% have least three visitors. It can be observed from the Figure that there are malicious URLs whose visitor counts exceed one thousand visitor apiece. We then investigated the top five URLs with the highest number of visitors and observed that they are URLs whose first appearances date back as far as 2010, individually spanning at least a year and a half. One such URL, for example, was distributing a screen-saver program that was flagged by Symantec as distributing Trojan viruses. This observation (a) explains why they managed to attract such a high number of visitors and (b) underscores the fact that even though they have been distributing malicious content, they were never shut down in a timely fashion.

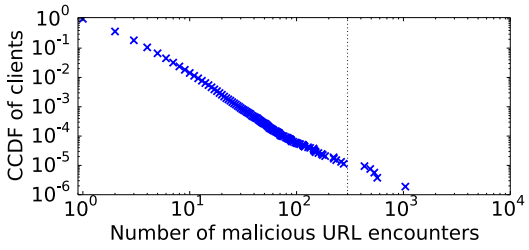

Fig. 6. CCDF of clients with respect to malicious URLs encounter

3) The distribution of the visits to malicious sites per user is skewed and can be described by a power law of exponent -1/2: In Figure 6, we plot the CCDF of the number of visits to malicious sites for each user. We find that the distribution of the number of malicious URL encounter per person seems to follow a power law distribution with exponent $\alpha = -\frac{1}{2}$. Thus, the good news is that most users in D_{WINE} encounter malicious URLs very infrequently. In Figure 7, we plot the CDF of the same distribution and show that 63% of all users visited a malicious URL only once during the entire eight months. However, 1.4% (roughly **7,500** users) visited malicious URLs at least ten times during the same amount of time. This in general suggests that a small group of users were far less cautious than others in their browsing activities.

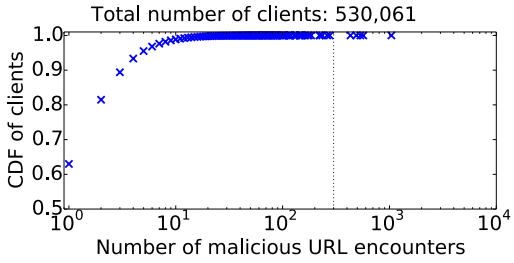

Fig. 7. CDF version of Figure 6

4) Outliers: users with more than 400 visits to malicious URLs: In both Figures 6 and 7, we see that a few data points are well separated from the rest of the distribution (to the left of the dotted vertical lines). Each of these points represents users who visited roughly 400 *distinct* malicious URLs during the eight-month period, averaging to at least *two* malicious URLs a day. We want to stress that each time one of these users visited one such malicious URL, a malicious file was downloaded by their browser and blocked from execution by the Symantec anti-virus product and the user would be subsequently notified. The most active user in our dataset visited a total of 1042 distinct malicious URLs for a duration of 242 days, averaging at least 4 a day. This behavior seems unlikely for a human, so we rule out this possibility.

Upon further investigation, we arrived at two possible explanations for these outlier points.

(i) These behaviors were generated by automated programs, for example a crawler whose purpose is to measure the uptime or downtime of a website or to seek out malicious domains. These programs could have been deployed by researchers.

(ii) Recall that each user in the D_{WINE} dataset is identified by a unique Machine ID, which is given by the Symantec software (think product number). It is possible for a user to install the AV product in a Virtual Machine and clone it, thereby allowing multiple Virtual Machines to report their activities to Symantec with the same Machine ID.

C. Space-needle: Many highly active URLs exhibit the same bursty temporal pattern that suggests a campaign-like behavior.

We discover that the visits to many URLs exhibit a bursty behavior, as can be seen in Figure 8. This temporal pattern,

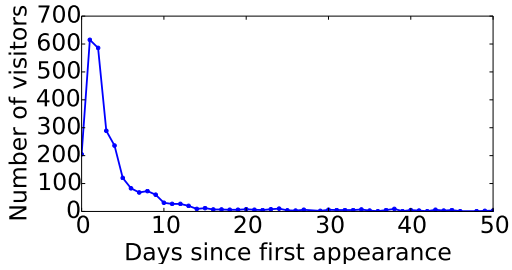

Fig. 8. One example of the space-needle propagation pattern

which we will call "space-needle", could be the result of an active campaign, staged by the hacker, to drive traffic to a newly infected or created site. Consequently, most of their visits take place during the first few days when the site appears, since after the first few days, the spam filters and black lists catch up and reduce the number of visitors.

We want to study the extent of the "space-needle" phenomenon in more detail. We start with focusing on URLs with a lifespan of 30 days or more and at least 100 visitors. Having at least 100 visitors *alone* does not qualify a URL to be highly active, a URL may just have that many visitors on the first few days since its appearance and is taken down afterwards. What we would like to study is, after all, the interesting malicious URLs that are both *long-lived* and had attracted substantial amounts of visitors.

We identify 2,402 URLs in D_{WINE} that meet these criteria. We will refer to these URLs from this point on as **Highly Active URLs**. We then do the following analysis to jointly define the "space-needle" pattern and quantify its presence with a technique that is commonly used in data mining and the steps of which are described below.

Given the set U of Highly Active URLs mentioned above, we begin with the following preprocessing steps with each $u \in U$:

1) We represent the user-visit pattern of each u during the first thirty days of its lifespan with the ordered sequence $V_u = \{(i, v_i^u)\}$ where i is the i^{th} day since u's first appearance and v_u^i is the number of distinct users who visited u on that same day.
   We only preserve the user visits during the first thirty days because (i) they are sufficient to capture most of the user visits to the URLs and (ii) we need to make sure that the activities that were captured from each URL span a uniform amount of time for the purpose of comparison.
2) For each V_u, we proceed to create the time series T_u by using linear interpolation to fill in any existing "gap" (which can be any day that there is no recorded user visit to the URL).

Once we have $T = \{T_u \quad \forall u \in U\}$, we:

1) Select a representative time series T_r that intuitively captures the essence of the "space-needle" (seen in Figure 8) shape and remove it from T
2) Compute the Euclidean distance between each $T_u \in \{T - T_r\}$ to T_r.
3) Sort each $T_u \in \{T - T_r\}$ so that if T_u precedes $T_{u'}$, $E(T_u, T_r) \leq E(T_{u'}, T_r)$ where E denotes the Euclidean distance function.
4) Manually inspect the "shape" of each time series from the beginning of the sorted list until we come across a time series that no longer visually resembles that of the representative time series T_r.

At the end of this process, we find 1,095 URLs or 45.6% of the 2,402 highly-active URLs, which we will call **Space-Needle URLs**.

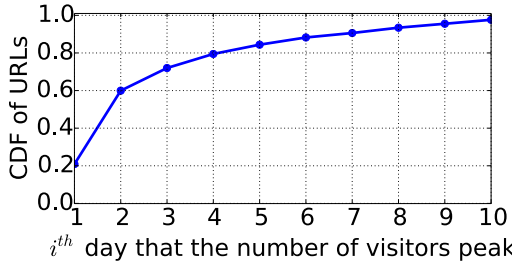

Fig. 9. Distribution of which day the URLs gained the maximum number of visitors

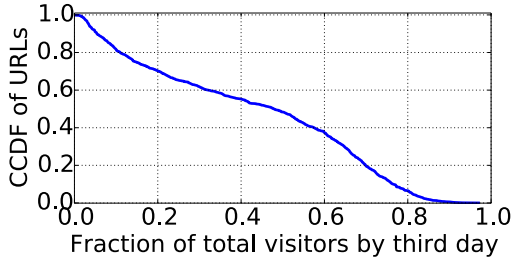

Fig. 10. Distribution of the fraction of total number of visitors each URL accumulated by the third day

The next step is to quantify the properties of the Space-Needle URLs. We can see from Figures 9 and 10 that 60% of these URLs achieved their peak number of daily visitors either on their first day of appearance, or the very next day. By the end of the third day, 50% of all the Space-Needle URLs have seen at least half of their total number of visitors. Moreover, for 80% of the Space-Needle URLs, the peak number of daily visitors is at least one order of magnitude larger than the median value of daily visitors. Note that we only consider days with at least one visitor to compute the value of the median.

D. Where do malicious domains end up?

When we performed DNS queries on all of the domains of the malicious URLs reported in D_{WINE}, we found that roughly a third of the malicious domains (35.9%) continue to be active (by which we mean that doing DNS queries on them yield IP addresses). We were intrigued as to why these domains (presumably of malicious websites) are still active even though they were first reported in 2011. To further investigate, we

randomly selected 600 still-active domains and accessed them in a browser in a virtual machine and we identified the following categories.

1) Recovered: 36.7% of the domains seem to be benign, bearing legitimate content. We believe they might have been compromised when Symantec detected malicious binaries being distributed by their servers. This suggests that compromised websites seem to have also played a significant role in malware delivery in 2011.
2) File and content sharing: 23.0% of the domains are file-sharing websites and software distributors. This suggests that malware had been uploaded to these sites and long since removed.
3) Parked: 20.2% of the domains are now under control of domain parkers and serving as advertisement space.
4) Not accessible: 20.1% of the domains were not accessible when we tried them, e.g. they returned `40X` error codes or blank pages.

E. Some users are more prone to careless surfing behavior.

We tried to estimate the probability of a user visiting a malicious URL given their history by executing the following steps.

1) Select one month from January to *July* of 2011.
2) Calculate how many URLs each client encountered during that month.
3) Let C_x^i be the set of clients who visit x URLs during month i, and let V^{i+1} be the set of the visitors to malicious URLs during the following month. We compute the percentage of users in C_x^i who will be **repeat offenders**: $P_x^{i \to i+1} = |C_x^i \cap V^{i+1}| / |C_x^i|$.
4) Repeat the steps above for every other month.
5) Compute the *average* $P_x^{i \to i+1} \quad \forall x$ across all months.

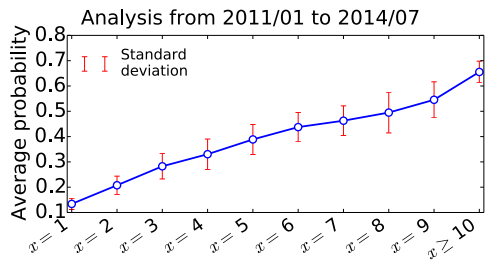

Fig. 11. Probability of a user visiting a malicious URL given the number of malicious URLs visited during previous month

In Figure 11, we plot the average probability of a user visiting a malicious URL within a month given the number of such visits (x) the month before. The average probability is computed across all users with the same number of visits to malicious sites for all pairs of consecutive months. We observe significant increase in the average probability as the number of URLs visited grows, indicating that there are indeed users who are prone to careless surfing behaviors even though they have anti-virus products installed on their computers.

We noted above in section IV-B4 that there are cases where a single machine ID visited thousands of sites. We do not believe that these outliers contribute meaningfully to the phenomenon we described in Figure 11, as there are very few outlier machine IDs and there are more than half a million machine IDs that are observed for this part of the study.

V. Related Work

The aim of our work is different from that of the majority of URL classification methods [13] [14], which focus on distinguishing malicious URLs from non-malicious ones while we focus on identifying whether a site identified as malicious are born-malicious or in fact compromised by hackers.

The most related work to profiling the behaviors of binary distribution by Papelaxakis et al. [15] focuses on benign binaries and presents a model called SHARKFIN that describes the propagation pattern of popular software. A recent work by Kuhrer et. al. [16] evaluates the completeness of black lists and presents a method to identify parked domains and sink holes. In recent work, Li et. al. [17] describe a method to identify a website that is compromised by re-direct script injection. Although this is relevant to our work, the proposed method targets a very specific type of compromised websites, while we need a general method to distinguish compromised websites at large from born-malicious ones.

Overall, there are four areas that touch on various aspects of web-based malware study.

A. Investigating the landscape of web-based malware distribution.

This first area then is split into two smaller ones: (a) how to actively seek out new malicious sites [18][19][20] and (b) the detection of malicious URLs [21][22], drive-by-downloads website [14][23], or malware-infected machines [13]. In [14], the authors statically analyze the content of websites to accomplish the goal of detection of drive-by-download sites and in [23], the authors attempt to detect when a user is redirected multiple times and eventually delivered to a website managed by malware distribution networks by analyzing the URLs in the redirect chains themselves.

In [13], Invernizzi et. al. invest their effort into the detection of machines in large-scale networks that meet the following criteria: (i) the machines have already been infected by drive-by-download attacks and (ii) the small piece of code dropped into each machines is sending HTTP requests to remote hosts to down- load the full payload for the installation of the malware. This work is unlike ours in that the goal of the work is to identify the infected hosts and, consequently, can be used for identify malicious websites that provide the malware payload. The author, however, never put a focus on identifying compromised websites.

B. Detecting vulnerable sites.

This area focuses more on the identification of websites that may be at risk of infiltration [24][25]. In this work [25], the authors created a classifier that identifies websites that may

become compromised in the future by automatically extracting content-based features from a large corpus of labeled websites as well as using out-of-band information such as Alexa ranking the Amazon Web Information Service. In [24], Canali et. al. hosted vulnerable websites on many hosting providers and tried to compromise the sites themselves. They showed, alarmingly, afterwards that the providers are unable to detect simple signs of malicious activities.

C. Studying the malware ecosystem.

This area studies the ecosystem that supports the malware distribution, providing insights on the attacks carried by malicious websites on the users [26] or on the infrastructure that supports malware authors [14][27], enabling them to spread their malicious software for monetary gains.

D. Malware binary analysis and classification.

This area focuses on the analysis of the web-based malware binaries [28][29][30]. In [28], the authors extracted features from the HTTP traffic traces generated by the malware installed on safe environments and used those features (which included total number of requests, average number of parameters, etc.) to cluster the malware samples. From the clusters, signatures can be generated to detect when a computer may be infected. Rossow et. al. monitored more than 100,000 malware samples at runtime in their Sandnet environment [29] and observed their network behaviors, thereby showing that DNS and HTTP are the two protocols most common among those used by the malware. In [30], Rossow et. al. extended their work to 23 different malware downloaders, most of which were yet documented. The authors characterized them according to their communication models, investigated their resilience, and analyzed how they they used DNS and fast-flux techniques to carry out their operations.

The work in this fourth area, while dealing directly with network-based malware, is of little help to us as they are malware-centric, as never got access to the binaries.

VI. Conclusion

In this paper, we focus on modeling the user exposure to web-based malware by analyzing more than 500K users accessing roughly 600K URLs from a data set collected from Symantec's WINE Project. We find that:

a) Compromised websites play a significant role in malware dissemination, as 33.1% of them in D_{ODB} dataset are compromised websites. In section III, we showed the different ways in which a compromised websites are fundamentally different from a born-malicious one.

b) A malicious URL often distributes many different malware binaries but each malicious binary is typically distributed by one URL.

c) Most malicious URLs (71.6%) are short-lived, but 10% of them are active for more than three months in D_{WINE}.

d) The number of visitors of many malicious website exhibit a bursty campaign-like temporal pattern, which we refer to as the "Space-needle" pattern.

e) The distribution of the visits to malicious sites per user is skewed and can be described by a power law of exponent $-\frac{1}{2}$.

Our study is a first step towards modeling web-based malware exposure and could help us understand malware distribution as a network-wide phenomenon.

This work was supported by the NSF Grant SaTC 1314935

References

[1] E. Mills, "Google finds 9,500 new malicious Web sites a day," www.cnet.com/news/google-finds-9500-new-malicious-web-sites-a-day/, June 2012.

[2] "WoT Reputation API," https://www.mywot.com/wiki/API.

[3] "Cybercrime Tracker," http://cybercrime-tracker.net/.

[4] "Malc0de Database," http://malc0de.com/database/.

[5] "MDL," http://www.malwaredomainlist.com/.

[6] "Vx vault," http://vxvault.siri-urz.net/ViriList.php.

[7] T. Dumitras and D. Shou, "Toward a standard benchmark for computer security research: The worldwide intelligence network environment (wine)," in *BADGERS 2011*. ACM.

[8] L. Bilge and T. Dumitras, "Before we knew it: an empirical study of zero-day attacks in the real world," in *CCS 2012*. ACM, pp. 833–844.

[9] K. Nayak, D. Marino, P. Efstathopoulos, and T. Dumitras, "Some Vulnerabilities Are Different Than Others: Studying Vulnerabilities and Attack Surfaces in the Wild ," *RAID 2014*.

[10] "Symantec's 2014 Security Threat Report," http://www.symantec.com/security_response/publications/threatreport.jsp.

[11] "Selenium, Web Browser Automation," http://www.seleniumhq.org/.

[12] I. H. Witten, E. Frank, L. E. Trigg, M. A. Hall, G. Holmes, and S. J. Cunningham, "WEKA: Practical machine learning tools and techniques with Java implementations," 1999.

[13] L. Invernizzi, S.-J. Lee, S. Miskovic, M. Mellia, R. Torres, C. Kruegel, S. Saha, and G. Vigna, "Nazca: Detecting malware distribution in large-scale networks," in *NDSS 2014*.

[14] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: a fast filter for the large-scale detection of malicious web pages," in *WWW 2014*. ACM.

[15] E. E. Papalexakis, T. Dumitras, D. H. P. Chau, B. A. Prakash, and C. Faloutsos, "Spatio-temporal mining of software adoption & penetration," in *ASONAM 2013*. ACM.

[16] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *RAID 2014*.

[17] Z. Li, S. Alrwais, X. Wang, and E. Alowaisheq, "Hunting the red fox online: Understanding and detection of mass redirect-script injections," in *S&P 2014*.

[18] Z. Li, S. Alrwais, Y. Xie, F. Yu, and X. Wang, "Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures," in *S&P 2013*. IEEE.

[19] C. Seifert, I. Welch, P. Komisarczuk, C. U. Aval, and B. Endicott-Popovsky, "Identification of malicious web pages through analysis of underlying dns and web server relationships." in *LCN 2008*.

[20] J. W. Stokes, R. Andersen, C. Seifert, and K. Chellapilla, "Webcop: Locating neighborhoods of malware on the web," in *LEET 2010*.

[21] A. Le, A. Markopoulou, and M. Faloutsos, "Phishdef: Url names say it all," in *INFOCOM 2011*. IEEE.

[22] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in *SIGKDD 2009*.

[23] J. Zhang, C. Seifert, J. W. Stokes, and W. Lee, "Arrow: Generating signatures to detect drive-by downloads," in *WWW 2011*.

[24] D. Canali, D. Balzarotti, and A. Francillon, "The role of web hosting providers in detecting compromised websites," in *WWW 2013*.

[25] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in *Usenix Security 2014*.

[26] N. P. P. Mavrommatis and M. A. R. F. Monrose, "All your iframes point to us," in *Usenix Security 2008*.

[27] C. Grier, L. Ballard, J. Caballero, N. Chachra, C. J. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis *et al.*, "Manufacturing compromise: the emergence of exploit-as-a-service," in *CCS 2012*. ACM.

[28] R. Perdisci, W. Lee, and N. Feamster, "Behavioral clustering of http-based malware and signature generation using malicious network traces." in *NSDI 2010*.

[29] C. Rossow, C. J. Dietrich, H. Bos, L. Cavallaro, M. Van Steen, F. C. Freiling, and N. Pohlmann, "Sandnet: Network traffic analysis of malicious software," in *Proceedings of the Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*. ACM, 2011, pp. 78–88.

[30] C. Rossow, C. Dietrich, and H. Bos, "Large-scale analysis of malware downloaders," *DIMVA 2013*.