usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents

Yang Liu, Armin Sarabi, Jing Zhang, and Parinaz Naghizadeh, *University of Michigan;*
Manish Karir, *QuadMetrics, Inc.;* Michael Bailey, *University of Illinois at Urbana-Champaign;*
Mingyan Liu, *University of Michigan and QuadMetrics, Inc.*

https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/liu

This paper is included in the Proceedings of the
24th USENIX Security Symposium

August 12–14, 2015 • Washington, D.C.

ISBN 978-1-931971-232

Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents

Yang Liu[1], *Armin Sarabi*[1], *Jing Zhang*[1], *Parinaz Naghizadeh*[1]
Manish Karir[2], *Michael Bailey*[3], *Mingyan Liu*[1,2]
[1] *EECS Department, University of Michigan, Ann Arbor*
[2] *QuadMetrics, Inc.*
[3] *ECE Department, University of Illinois, Urbana-Champaign*

Abstract

In this study we characterize the extent to which cyber security incidents, such as those referenced by Verizon in its annual Data Breach Investigations Reports (DBIR), can be predicted based on externally observable properties of an organization's network. We seek to proactively forecast an organization's breaches and to do so without cooperation of the organization itself. To accomplish this goal, we collect 258 externally measurable features about an organization's network from two main categories: mismanagement symptoms, such as misconfigured DNS or BGP within a network, and malicious activity time series, which include spam, phishing, and scanning activity sourced from these organizations. Using these features we train and test a Random Forest (RF) classifier against more than 1,000 incident reports taken from the VERIS community database, Hackmageddon, and the Web Hacking Incidents Database that cover events from mid-2013 to the end of 2014. The resulting classifier is able to achieve a 90% True Positive (TP) rate, a 10% False Positive (FP) rate, and an overall 90% accuracy.

1 Introduction

Recent data breaches, such as those at Target [35], JP Morgan [25], and Home Depot [49] highlight the increasing social and economic impact of such cyber incidents. For example, the JP Morgan Chase attack was believed to be one of the largest in history, affecting nearly 76 million households [25]. Often, by the time a breach is detected, it is already too late and the damage has already occurred. As a result, such events call into the question whether these breaches could have been predicted and the damage avoided. In this study we seek to understand the extent to which one can forecast if an organization may suffer a cyber security incident in the near future.

Machine learning has been used extensively in the cyber security domain, most prominently for *detection* of various malicious activities or entities, e.g., spam [44, 45] and phishing[39]. It has been used far less for the purpose of *prediction*, with the notable exception of [51], where textual data is used to train classifiers to predict whether a currently benign webpage may turn malicious in the near future. The difference between detection and prediction is analogous to the difference between diagnosing a patient who may already be ill (e.g., by using biopsy) vs. projecting whether a presently healthy person may become ill based on a variety of relevant factors. The former typically relies on identifying known characteristics of the object to be detected, while the latter on factors believed to correlate with the prediction objective.

To explore the effectiveness of forecasting security incidences we begin by collecting *externally* observed data on Internet organizations; we do not require information on the internal workings of a network or its hosts. To do so, we tap into a diverse set of data that captures different aspects of a network's security posture, ranging from the *explicit* or *behavioral*, such as externally observed malicious activities originating from a network (e.g., spam and phishing) to the *latent* or *relational*, such as mismanagement and misconfigurations in a network that deviate from known best practices. From this data we extract 258 features and feed them to a Random Forest (RF) classifier. We train and test the classifier on these features and more than 1,000 incident reports taken from the VERIS community database [55], Hackmageddon [42], and the Web Hacking Incidents Database [31] that cover events from mid-2013 to 2014. The resulting classifier can be configured over a wide range of operating points including one with 90% True Positive (TP) rate, 10% False Positive (FP) rate and an overall accuracy of 90%.

We posit that such cyber incident forecasting offers a completely different set of characteristics as compared to detection techniques, which in turn enables entirely new classes of applications that are not feasible with detection techniques alone. First and foremost, prediction allows *proactive* policies and measures to be adopted rather than *reactive* measures following the detection of an incident. Effective proactive actions can substantially reduce the potential cost incurred by an incident; in this sense prediction is complementary to detection. Cyber incident prediction also enables the development of effective risk management schemes such as cyber insurance, which introduces monetary incentives for the adoption of better cyber security policies and technologies. In the wake of recent breaches, the market for such policies has soared, with current written annual premiums estimated to be between $500M and $1B [47].

The remainder of the paper is organized as follows. Section 2 introduces the datasets used in this study and details the rationale for their use as well as our processing methodology. We then define the features we use in constructing the classifier and show why they are relevant in predicting security incidents in Section 3. We present the main prediction results as well as their implications in Section 4. In Section 5 we discuss a number of observations and illustrate several major data breaches in 2014 in the context of this prediction methodology. Related work is detailed in Section 6, and Section 7 concludes the paper.

2 Data Collection and Processing

Our study draws from a variety of data sources that collectively characterize the security posture of organizations, as well as security incident reports used to determine their security outcomes. These sources are summarized in Table 1 and detailed below; a subset of these has been made available at [7].

2.1 Security Posture Data

An organization's network security posture may be measured in various ways. Here, we utilize two families of measurement data. The first is measurements on a network's misconfigurations or deviations from standards and other operational recommendations; the second is measurements on malicious activities seen to originate from that network. These two types of measurements are related. In particular, in [58] Zhang et al. quantitatively established varying degrees of correlation between eight different mismanagement symptoms and the amount of malicious activities from an organization. The combination of both of these datasets represents a fairly comprehensive view of an organization's externally discernible security posture.

2.1.1 Mismanagement Symptoms

We use the following five mismanagement symptoms in our study, a subset of those studied in [58].

Open Recursive Resolvers: Misconfigured open DNS resolvers can be easily used to facilitate massive amplification attacks that target others. In order to help the network operations community address this wide spread threat, the Open Resolver Project [14] actively sends a DNS query to every public IPv4 address in port 53 to identify misconfigured DNS resolvers. In this study, we use a data snapshot collected on June 2, 2013. In total, 27.1 million open recursive resolvers were identified.

DNS Source Port Randomization: In order to minimize the threat of DNS cache poisoning attacks [13], current best practice (RFC 5452 [34]) recommends that DNS servers implement both source port randomization and a randomized query ID. Many servers however have not been patched to implement source port randomization. In [58], over 200,000 misconfigured DNS resolvers were detected based on the analysis over a set of DNS queries seen by VeriSign's .com and .net TLD name server on February 26, 2013. This is the data used in this study.

BGP Misconfiguration: BGP configuration errors or reconfiguration events can cause unnecessary routing protocol updates with short-lived announcements in the global routing table [40]. Zhang et. al detected 42.4 million short-lived routes with BGP updates from 12 BGP listeners in the Route Views project [32] during the first two weeks of June 2013 [58]; this data is used in our study.

Untrusted HTTPS Certificates: Secure websites utilize X.509 certificates as part of the TLS handshake in order to prove their identity to clients. Properly configured certificates should be signed by a browser-trusted certificate authority. It is possible to detect misconfigured websites by validating the certificate presented during the TLS handshake [33]. An Internet scan performed on March 22, 2013 found that only 10.3 million out of a total of 21.4 million sites presented browser-trusted certificates [58]. We use this dataset in our study.

Open SMTP Mail Relays: Email servers should perform filtering on the message source or destination to only allow users in their own domain to send email messages. This is documented in current best practice (RFC 2505 [38]), and misconfigured servers can be used in large scale spam campaigns. Though small in number, these represent a severe misconfiguration in an organizations' infrastructure. In this study, we use data collected on July 23, 2013, which detected 22,284 open mail relays [58].

None of the datasets mentioned above is necessarily directly related to a vulnerability. The presence of misconfigurations in an organization's networks and infras-

Category	Collection period	Datasets
Mismanagement symptoms	February 2013 - July 2013	Open Recursive Resolvers, DNS Source Port Randomization, BGP misconfiguration, Untrusted HTTPS Certificates, Open SMTP Mail Relays [58]
Malicious activities	May 2013 - December 2014	CBL[4] , SBL[22], SpamCop[19], WPBL[24], UCEPROTECT[23], SURBL[20], PhishTank[16], hpHosts[11], Darknet scanners list, Dshield[5], OpenBL[15]
Incident reports	August 2013 - December 2014	VERIS Community Database [55], Hackmageddon [42], Web Hacking Incidents [31]

Table 1: Summary of datasets used in this study. Mismanagement and malicious activity data are used to extract features, while incident reports are used to generate labels for the training and testing of a classifier.

tructure is, however, an indicator of the lack of appropriate policies and technological solutions to detect such failures. The latter increases the potential for a successful data breach.

Also, note that all of the above datasets were collected during roughly the first half of 2013. As we shall be using the mismanagement symptoms as features in constructing a classifier/predictor, it is important that these features reflect the condition of a network *prior* to the incidents. Consequently, our incident datasets (detailed in Section 2.2) cover incidents that occurred between August 2013 and December 2014. Note also that we use only a single snapshot of each of the symptoms; this is because such symptomatic data is relatively slow-changing over time, as systems are generally not reconfigured on a daily or even weekly basis.

2.1.2 Malicious Activity Data

Another indicator of the lack of policy or technical measures to improve security at an organization is the level of malicious activities observed to originate from its network assets and infrastructure. Such activity is often observed by well-established monitoring systems such as spam traps, darknet monitors, or DNS monitors. These observations are then distilled into blacklists. We use a set of reputation blacklists to measure the level of malicious activities in a network. This set further breaks down into three types: (1) those capturing spam activities, including CBL[4] , SBL[22], SpamCop[19], WPBL[24], and UCEPROTECT[23], (2) those capturing phishing and malware activities, including SURBL[20], PhishTank[16], and hpHosts[11], and (3) those capturing scanning activities, including the Darknet scanners list, Dshield[5], and OpenBL[15]. We use reputation blacklists that have been collected over a period of more than a year, starting in May 11, 2013 and ending in December 31, 2014. Each blacklist is refreshed on a daily basis and consists of a set of IP addresses seen to be engaged in some malicious activity. This longitudinal dataset allows us to characterize not only the presence of malicious activities from an organization, but also its dynamic behavior over time.

2.2 Security Incident Data

In addition to the security posture data described in the previous section, we require data on reported cyber-security incidents to serve as ground-truth in our study; such data is needed for the purpose of training the classifier, as well as for assessing its accuracy in predicting incidents (testing). In general, we believe such incidents are vastly under reported. In order to obtain a good coverage, we employ three collections of publicly available incident datasets. These are described below.

VERIS Community Database (VCDB) [55]: This dataset represents a broad ranging public effort to gather cyber security incident reports in a common format [55]. The collection is maintained by the Verizon RISK Team, and is used by Verizon in its highly publicized annual Data Breach Investigations Reports (DBIR) [56]. The current repository contains more than *5,000* incident reports, that cover a variety of different types of events such as server breach, website defacements, and physically stolen assets. Table 7 (in the Appendix) provides some example reports from this repository; a majority (64.99%) is from the US.

Of the full set, roughly 700 unique incidents were relevant to our study: we include only incidents that occurred after mid-2013 so that they are aligned with the security posture data, and those directly reflecting cyber-security issues. We therefore exclude those due to physical attacks, robbery, deliberate mis-operation by internal actors (e.g. disgruntled employees) and the like, as well as unnamed or unverified attack targets. We show several such examples in Table 2. Also note that even though the same IPs may appear in both the malicious and incident data, the independence of the features from ground-truth data is maintained because malicious activities only reveal botnet presence, which is *not* considered an incident type by or reported in any of our incident datasets.

Incident report	Reason to exclude
Student of a college changed score	Unknown target
Road construction sign hacked	Physical tampering
Praxair Healthcare Inc. asset stolen	Physical theft
Lucile Packard Child. Hosp.l asset stolen	Physical theft
Medicare Privilege Misuse	Deliberate internal misuse

Table 2: Examples of excluded VCDB incidents.

Hackmageddon [42]: This is an independently maintained cyber incident blog that aggregates and documents various public reports of cyber security incidents on a monthly basis. From the overall set we extract 300 incidents, in which the reported dates are aligned with our security posture data, between October 2013 and February 2014, and for which we are able to clearly identify the affected organizations.

The Web Hacking Incidents Database (WHID) [31]: This is an actively maintained cyber security incident repository; its goal is to raise awareness of cyber security issues and to provide information for statistical analysis. From the overall dataset we identify and extract roughly 150 incidents, for which the reported dates are aligned with our security posture data, between January 2014 and November 2014.

A breakdown of the incidents by type from each of these datasets is given in Table 5. Note that Hackmageddon and WHID have similar categories while VCDB has much broader categories.

Incident type	SQLi	Hijacking	Defacement	DDoS
Hackmageddon	38	9	97	59
WHID	12	5	16	45
Incident type	Crimeware	Cyber Esp.	Web app.	Else
VCDB	59	16	368	213

Table 3: Reported cyber incidents by category. Only the major categories in each set are shown. The "Else" category by VCDB represents incidents lacking sufficient detail for better classification.

2.3 Data Pre-processing

Though our diverse datasets give us substantial visibility into the state of security at an organizational level, the diversity also presents substantial challenges in aligning the data in both time and space. All of the security posture datasets – mismanagement and malicious activities – record information at the host IP-address level; e.g., they reveal whether a particular IP address is blacklisted on a given day, or whether a host at a specific IP address is misconfigured. On the other hand, a cyber incident report is typically associated with a company or organization, not with a specific IP address within that domain.

Conceptually, it is more natural to predict incidents for an organization for the following reasons. Firstly, our interest is in predicting incidents broadly defined as a way to assess organizational cyber risk. Secondly, while some IP addresses are statically associated with a machine, e.g., a web server, others are dynamically assigned due to mobility, e.g., through WiFi. In the latter case predicting for specific IP addresses no longer makes sense.

This mismatch in resolution means that we will have to (1) map an organization reported in an incident to a set of IP addresses and (2) aggregate mismanagement and maliciousness information over this set of addresses. To address the first step we will first retrieve a *sample IP address* in the network of the compromised organization, which is then used to identify an *aggregation unit* – a set of IP addresses – that allows us to recover the network asset involved in the incident. Sample IP addresses are obtained by manually processing each incident report, and the aggregation units are identified by using registration information from Regional Internet Registries (RIR) databases. These databases are collected separately from ARIN [3], LACNIC [12], APNIC [2], AFRINIC [1] and RIPE [18], who keep records of IP address blocks/prefixes that are allocated to an organization. ARIN, APNIC, AFRINIC and RIPE databases keep track of the IP addresses that have been allocated, along with the organizations they have been allocated to, labeled with a maintainer ID. LACNIC provides a less detailed database, only keeping track of allocated blocks and not the owners. In this case, we take the last allocation that contains our sample IP address – note a single IP address might be reallocated several times, as part of different IP blocks – i.e., the smallest block, as its owner.

2.3.1 Mapping Process

In the following paragraphs we explain in detail the manual process of (1): (1a) extracting sample IP addresses through a number of examples, and (1b) identifying the aggregation unit using the sample IP address. The general outline of the process for (1a) is that we first read the report concerning each incident, and extract the website of the company involved. If the website is the intrusion point in the breach, or indicative of the compromised network, then we take the address of this website to be our sample IP address. The website is determined to be indicative of the compromised network when the owner ID for the sample IP address matches the reported name of the victim network. Occasionally the victim network can be identified separately regardless of the website address, but in most cases this is found to be an effective way of quickly obtaining the owner ID.

Our first example [21] is a website defacement targeting the official website of the City of Mansfield, Ohio. Since the point of intrusion is clearly the website, we take its address as our sample IP address for this incident. Note that in this case the website might be managed by a 3rd party hosting company, a possibility discussed further when we explain the process to address (1b). The second example [6] is on Evernote resetting all user passwords following an attack on its online system. For this incident we identify said domain (evernote.com), and trace it to an IP block in ARIN's database registered to Evernote Corporation. Since this network is maintained by Evernote itself, we take evernote.com to be our sample IP address. Our final example [10] involves the defacement of Google Kenya and Google Burundi websites. As the report suggests, the hackers altered the DNS records of the domains by hacking into the Kenya and Burundi NICs. Since the attack was not through directly compromising the defaced websites, we excluded this incident – the victim in this incident is neither Google Kenya nor Google Burundi, but the networks owned by the NICs.

The above examples provide insight into the manual process of mapping incident to a network address. For a large portion of the reports the incident descriptor is unique and should therefore be treated as such; this is the main reason that such a mapping is primarily done manually. For a significant portion ($\sim$ 95%) of the reports we are able to identify the compromised network with a high level of confidence – in such cases either the report explicitly cites the website as the intrusion point (first example), or the network identified by the website is registered under the victim organization (second example). When neither of these conditions is satisfied, this incident is excluded unless we can identify the victim network through alternative means; such cases are few. Overall our process is a conservative one: we only include an incident when there is zero or minimal ambiguity. Finally, we also remove duplicate owner IDs in order to avoid a bias against commonly used hosting companies (e.g. Amazon, GoDaddy) in our training and testing process.

We now explain the process used in (1b) to map an obtained sample IP address (as well as the identified owner ID) to network(s) operated by a single entity. The general outline of this process is as follows: we take all the IP blocks that have the same owner ID listed in the RIR databases, excluding sub-blocks that have been reallocated to other organizations, as our aggregation unit. Continuing with the same set of examples, in the case of Evernote (second example) we reverse search ARIN's database and extract all IP blocks registered to Evernote Corporation, giving us a total of 520 IP addresses. For the case of the City of Mansfield website, using records kept by ARIN we see that its web address belongs to Linode, a cloud hosting company. Obviously Linode is also hosting other entities on its network without reported incidents. Nonetheless, in this case we take the network owned by Linode as our aggregation unit, since we cannot further differentiate the source IP address(es) more closely associated with the city. The inclusion of such cases is a tradeoff as excluding them would have left us with too few samples to perform a meaningful study. More on this is discussed in Section 2.4.

2.3.2 A global table of aggregation units

The above explains how we process the incident reports to identify network units that should be given a label of "1", i.e., victim organizations. For training and testing purposes we also need to identify network units that should be given a label of "0", i.e., non-victim organizations. To accomplish this, we built a global table using information gathered from the RIRs that provides us with a global aggregation rule, containing both victim and non-victim organizations. Our global table contains 4.4 million prefixes listed under 2.6 million owner IDs. Note that the number of prefixes in the RIR databases is considerably larger than the global BGP routing table size, which includes roughly 550,000 unique prefixes [41]. This is partly due to the fact that the prefixes in our table can overlap for those that have been reallocated multiple times. In other words, the RIR databases can be viewed as a tree indicating all the ownership allocations and reallocations over the IP address space. On the other hand, the BGP table tends to combine prefixes that are located within the same Autonomous System (AS), in order to reduce routing table sizes. Therefore, the RIR databases provide us with a finer-grained look into the IP address space. By taking all the IP addresses that have been allocated to an organization, and have not been further reallocated, we can break the IP address space into mutually exclusive sets, each owned and/or maintained by a single organization. Out of the 4.4 million prefixes, 300,000 of them are assigned by LACNIC and therefore have no owner ID. Combined with the 2.6 million owner IDs from the other registries, the IP address space is broken, by ownership (or LACNIC prefixes), into 2.9 million sets. Each set constitutes an aggregation unit that is given a label of "0", except for those already identified and labeled as "1" by the previous process.

2.3.3 Aggregation Process

Once these aggregation units are identified, the second step (2) is relatively straightforward. For each mismanagement symptom we simply calculate the fraction of symptomatic IPs within such a unit. For malicious activities, we count the number of unique IP addresses listed on a given day (by a single blacklist, or by blacklists monitoring the same type of malicious activities) that belong to this unit; this results in one or more time series for each unit. This step is carried out in the same way for both victim and non-victim organizations.

2.4 A Few Caveats

As already alluded to, our data processing consists of a series of rules of thumb that we follow to make the data useable, some perhaps less clear-cut than others. Below we summarize the typical challenges we encounter in this process and their possible implications on the prediction performance.

As described in Section 2.3, the aggregation units are defined using ownership information from RIR databases. One issue with the use of ownership information is that big corporations tend to register their IP address blocks under multiple owner IDs, and in our processing these IDs are treated as separate organizations. In principle, as long as each of the aggregation units is non-trivial in size, each can have its own security posture assessed. Furthermore, in some cases it is more accurate to treat such IDs separately, since they might represent different sections of an organization under different management. The opposite issue also exists, where it may be impossible to distinguish between the network assets of multiple organizations; recall, e.g. our first example where multiple organizations are hosted on the same network. As mentioned before, we have chosen in such cases to use the owner ID as the aggregation unit. While this mapping process is clearly non-ideal, it is a best-effort attempt at the problem, and will instead provide the classifier with the average value of the features over all organizations hosted on the identified network.

The labels for our classifier are extracted from real incident reports, and we can safely assume that the amount of false positives in these reports, if any, is negligible. However data breach incidents are only reported when an external source detects the data breach (e.g. website defacements), or an organization is obligated to report the incident due to private customer information getting compromised. In general, organizations tend not to announce incidents publicly, and security incidents remain largely under-reported. This will affect our classifier in two ways: First, by failing to incorporate all incidents in our training set, we may fail to identify all of the factors that might affect an organization's likelihood of suffering a breach. Second, when choosing non-victim organizations, it is possible that we select some of them from unreported victims, which could further impact the accuracy of our classifier. We have tried to overcome this challenge by using three independently maintained incident datasets. Ultimately, however, this can only be addressed when timely incident reporting becomes the norm; more on this is discussed in Section 5.

Last but not least, all the raw security posture data (mismanagement symptoms and blacklists) could contain error, which we have no easy way of calibrating. However, two aspects of the present study help mitigate the potential impact of these noises. Firstly, we use many different datasets from independent sources; the diversity and the total volume generally have a dampening effect on the impact of the noise contained in any single source. Secondly and perhaps more importantly, our ultimate verification and evaluation of the prediction performance are not based on the security posture data, but on the incident reports (with their own issues as noted above). In this sense, as long as the prediction performance is satisfactory, the noise in the input data becomes less relevant.

3 Forecasting Methodology

The key to our prediction framework is the construction of a good classifier. We will primarily focus on the Random Forest (RF) method [37], which is an ensemble classifier and an enhancement to the classical random decision tree method. It uses randomly selected subsets of samples to construct different decision trees to form a forest, and is generally considered to work well with large and diverse feature sets. In particularly, it has been observed to work well in several Internet measurement studies, see e.g., [57]. As a reference, we will also provide performance comparison by using the Support Vector Machine (SVM) [27], one of the earliest and most common classifiers. To train a classifier, we need to identify a set of features from the measurement data. Below, we first detail the set of features used, and then present the training and testing procedures.

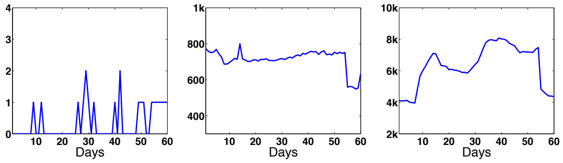
3.1 Feature Set

We shall use two types of features, a primary set and a secondary set. The primary set of features consists of the raw data, while the secondary set is derived or extracted from the raw data, i.e., in the form of various statistics. In all, 258 features are used, including 5 mismanagement features, 180 primary features, 72 secondary features, and a last feature on the organization size.

3.1.1 Primary Features (186)

Mismanagement symptoms (5). There are five symptoms; each is measured by the ratio between the number of misconfigured systems and the total number of systems in an organization. For instance, for the untrusted HTTPS certificates, this ratio is between the number of misconfigured certificates over the total number of certificates discovered in an organization. Similarly, for open SMTP mail relay this ratio is between the number of misconfigured mail servers and the total number of mail servers. The only exception is in the case of open recursive resolver: since we do not know the total number of open resolvers, this ratio is between the number of misconfigured open DNS resolvers and the total number of IPs in an organization. These ratios are denoted as $\mathbf{m}_i \in [0,1]^5$ for organization i.

Malicious activity time series (60 × 3). For each organization we collect three separate time series, one for each malicious activity type, namely spam, phishing, and scan. Accordingly, for organization i, its time series data are denoted by $\mathbf{r}_i^{SP}, \mathbf{r}_i^{PH}, \mathbf{r}_i^{SC}$. These time series data are directly fed in their entirety into the classifier. Several examples of $\mathbf{r}_i^{SP}$ are given in Fig. 1; these are collected over a two-month (60 days) period and show the total number of unique IPs blacklisted on each day over all spam blacklists in our dataset.



(a) Org. 1 (b) Org. 2 (c) Org. 3

Figure 1: Examples of malicious activity time series of three organizations; Y-axis is the number of unique IP addresses listed on all spam blacklists in each day over a 60-day period.

Size (1). This refers to the size of an organization in terms of the number of IP addresses identified within that organization's aggregation unit as outlined in the previous section. For organization i, this is denoted by s_i.

The relevance of these symptoms to an organization's security posture is examined more closely by comparing their distributions among the victim and the non-victim populations, as shown in Fig. 2. We see a clear difference between the two populations in their untrusted HTTPS and Openresolver distributions. This difference suggests that these symptoms are meaningful distinguishers, and thus hold predictive power. This is indeed verified later when these two symptoms emerge as the most indicative of the five. By contrast, the other three mismanagement symptoms appear much less powerful.

The relevance of the malicious activity time series will be examined more closely in the next section, within the context of their secondary features. Lastly, the organization size can to some extent capture the likelihood of an organization becoming a target of intentional attacks, and is therefore included in the feature set.

3.1.2 Secondary Features (72)

In determining what type of statistics to extract to serve as secondary features, we aim to capture distinct behavioral patterns in an organization's malicious activities, particularly concerning their dynamic changes. To illustrate, the three examples given in Fig. 1 show drastically different behavior: Org. 1 shows a network with consistently low level of observed malicious IPs (and possibly within the noise inherent in the blacklists), while Examples 2 and 3 show much higher levels of activity in general. These two, however, differ in how persistent they are at those high levels. Example 2 shows a network with high levels throughout this period, while Example 3 shows a network that fluctuates much more wildly. Intuitively, such dynamic behavior reflects to a large degree how responsive the network operators are to blacklisting, i.e., time to clean up, time to resurfacing of malicious activities, and so on.

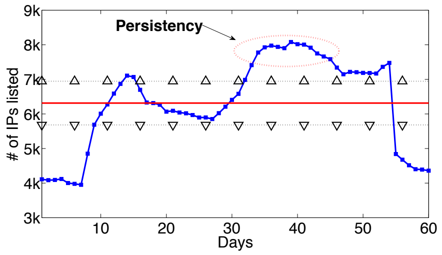


Figure 3: Extracting secondary features. The solid red line indicates time-average of the signal while the two dotted lines denote the boundary of different regions. The region above is "bad" with higher-than-average malicious activities, while the region below is "good" with lower-than-average activities. Persistency refers to the duration the time series persist in the same region.

These observed differences motivate us to collect statistics summarizing such behavioral patterns by measuring their persistence and change, e.g., how big is the change in the magnitude of malicious activities over time and how frequently does it change. To balance the expressiveness of the features and their complexity, we shall do so by first value-quantizing a time series into three regions relative to its time average: "good", "normal" and "bad". An illustration is given in Fig. 3 using one of the examples shown earlier (Org. 3). The solid line marks the average magnitude of the time series over the observation period; the dotted lines then outline the "normal" region, i.e., a range of magnitude values that are relatively close (either from above or below) to its time-average. The region above the top dotted line is accordingly referred to as the "bad" region, showing large number of malicious IPs, and the region below the bottom dotted line the "good" region, with a smaller number of malicious IPs, both relative to its average[1].

An additional motivation behind this quantization step is to capture certain onset and departure of "events", such as a wide-area infection, or scheduled patching and software update, etc. Viewed this way, the duration an orga-

[1] The choice on the size of the normal region may lead to differences in classifier performance, which is discussed in more detail in Section 5.3. In most of our experiments ±20% of the time average is used.

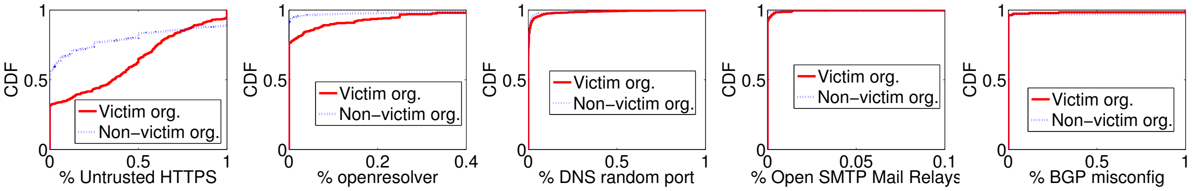


Figure 2: Comparison of mismanagement symptoms between the victim and non-victim populations. There is a clear separation under the first two, while the other three appear to be much weaker predictors.

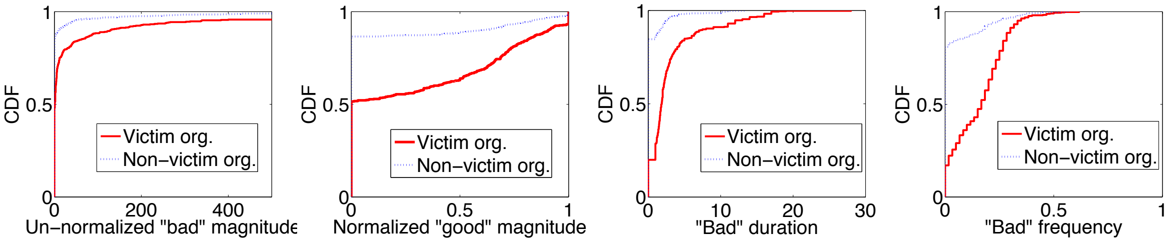


Figure 4: Profile of selected temporal features extracted from the scanning time series over the period Nov. 13-Dec.13.

nization spends in a "bad" region could be indicative of the delay in responding to an event, and similarly, how frequent it re-enters a "bad" region could be indicative of the effectiveness of the solutions taken in remaining clean.

Accordingly, for each region we then measure the average magnitude (both normalized by the total number of IPs in an organization and unnormalized), the average duration that the time series persists in that region upon each entry (in days), and the frequency at which the time series enters that region. This results in four summary statistics for each region, thus 12 values for each time series. Since each organization has three time series, one for each malicious activity type, we obtain a total of 36 derived features per organization i. These will be collectively denoted by the feature vector F_i. Note that the set of 36 values are collected from time series of a certain duration. Here we further distinguish between statistics extracted from a longer period of time vs. from a shorter, most recent period of time. In this study we use two such feature vectors, one referred to as *Recent-60* features that are collected over a period of 60 days (typically leading up to the time of incident) and the other *Recent-14* features collected over a period of 14 days (leading up to the time of incident).

To give a sense of why these features may be expected to hold predictive power, we similarly compare the distribution of these feature values among the victim and non-victim populations. Fig. 4 shows this comparison for four examples: un-normalized magnitude in a bad period, normalized magnitude in a good period, average duration during bad periods, and the frequency of entering a bad period. We see that in each case there is a clear difference between the two populations in how these feature values are distributed, e.g., victim organizations tend to have longer bad periods, indicative of slow response time, and also higher bad/good magnitudes, etc. As we discuss further in Section 4.4, these features have varying degrees of influence over the prediction outcome.

3.2 Training and Testing Procedure

We now describe the construction of the predictor using the set of features defined above. This consists of a training step and a testing step. The training step uses the following two sets of subjects.

A subset of incident or victim organizations. This will be referred to as Group(1) or the incident group. Depending on the experiments, this subset may be selected from one of the three incident datasets (if we train the classifier and conduct testing based solely on one incident dataset), or from the union of all three. This subset is selected based on the time stamps of the reported incidents, and its size is determined by a training-testing ratio, e.g., 70-30 split or 50-50 split of the given dataset. If we use a 50-50 split, it means that we select the first half (in terms of time of occurrence) of the incidents as Group(1); a 70-30 split means using the first 70% of incidents as Group(1). The remaining victim organizations are used in the testing step.

A randomly selected set of non-victim organizations (with size comparable to that of Group(1) in any given experiment). These are taken from the global table described in Section 2.3.2. This will be referred to as

Group(0), or the non-incident group. As mentioned earlier, since there are close to three million non-victim organizations compared to less than a thousand victim organizations, the random sub-sampling is necessary to avoid the common problem of imbalance in the machine learning literature[2]; this issue has also been discussed in [51]. This random selection of non-victim organizations is repeated numerous times, each time training a different classifier. The reported testing results are averages over all these versions.

For a victim organization i in Group(1), its complete feature set $\mathbf{x}_i$ includes the mismanagement symptoms $\mathbf{m}_i$, the three time series $\mathbf{r}_i^{SP}, \mathbf{r}_i^{PH}, \mathbf{r}_i^{SC}$ over the two months prior to the month in which the incident in i occurred[3], secondary features F_i collected over the same time period as the time series, namely Recent-60, and that collected over the two weeks prior to the month of the incident occurrence, namely Recent-14. Each such feature set is associated with the label (or ground-truth or group information in machine learning) $L_i = 1$ for incident. For a non-victim organization j in Group(0), its complete feature set $\mathbf{x}_j$ consists of exactly the same components listed above, with the only difference that the time series and the secondary features are for the two months prior to the month of the first incident in Group(1). It is also associated with the label $L_j = 0$ for non-incident.

The collections of $\{[\mathbf{x}_i, L_i]\}$ and $\{[\mathbf{x}_j, L_j]\}$ constitute the training data used to train the classifier. The testing step then uses the following two inputs: (1) The subset of victim organizations not included in Group(1); denote this group by Group(1^c). (2) A randomly selected set of non-victim organizations not used in training. Unlike in training where we try to keep a balance between the victim and non-victim sets, during testing we use a much larger set of non-victim organizations to better characterize the classifier performance.

For these two subjects their complete feature sets $\mathbf{x}_i$ are obtained in exactly the same way as for those used in training. For the non-victim organizations selected for testing, the features are collected over the two months prior to the incident month of the first incident in Group(1^c). For the victim organization used for testing we further consider two scenarios. In the *short-term forecast* scenario, we collect these features over the two months prior to the incident month for an organization in Group(1^c), while in the *long-term forecast* scenario, we collect these features over the two months prior to the incident month of the first incident in Group(1^c). In the short-term forecast scenario, since in each incident test case the incident occurred within a month of collecting the features, the TP rate is essentially for a forecasting window of one month. In the long-term forecast scenario, an incident may occur months after collecting the features (up to 12 months in the case of VCDB), thus the TP rate is for a forecasting window of up to a year. Note that the short-term forecast can be repeatedly done over time to produce prediction for the immediate future. The differences between these two forecast schemes are also illustrated in Fig. 5.

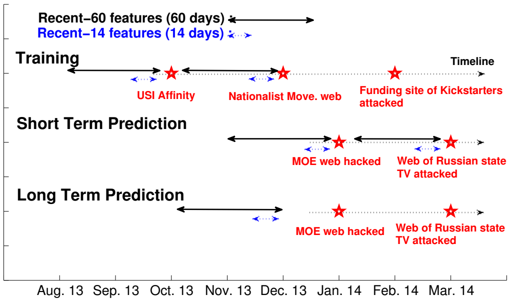


Figure 5: Feature extraction, short-term and long-term forecasting. In training, features are extracted from the most recent period leading up to an incident. In testing, the same is done when we perform short-term forecast. In long-term forecast, features are extracted from periods leading up to the time of the first incident used in testing.

These inputs are then fed into the classifier to produce a label (or prediction). The output of Random Forest is actually a risk probability; a threshold is then imposed to obtain a binary label. For instance, if we set the threshold at 0.5, then all output > 0.5 means a label of 1. By moving this threshold we obtain different prediction performances, which constitute a ROC curve.

4 Incident Prediction

In this section, we present our main prediction results and investigate their various implications.

4.1 Main Results

Using the methodology outlined in the previous section, we performed prediction using the three incident datasets separately, as well as collectively. When used collectively, we removed duplicate reports of the same incident whenever applicable. The separation between training and testing for each dataset is done chronologically, as shown in Table 4. For each dataset, these separations

[2] If we use all three million non-victims in training, the resulting classifier will simply label all of them as non-victims, and achieve performance very close to 100% overall. But clearly this classifier would be of little use, as it will also have 0 true positive probability.

[3] Most incident occurrences in our dataset are timestamped with month and year information.

result in an approximate 50-50 split of the victim set between the training and testing sample sizes. In addition, for each test we randomly sample non-victim test cases from the non-victim organization set.

	Hackmageddon	VCDB	WHID
Training	Oct 13 – Dec 13	Aug 13 – Dec 13	Jan 14 – Mar 14
Testing	Jan 14 – Feb 14	Jan 14 – Dec 14	Apr 14 – Nov 14

Table 4: Chronological separation between training and testing samples for each incident dataset; the split is roughly 50-50 among the victim population.

There is one point worth clarifying. When processing non-sequential data, the split of samples for the purpose of training and testing is often done randomly in the machine learning literature. In our context this would mean to choose a later incident for training and use an earlier incident for testing. Due to the sequential nature of our data, we intentionally and strictly split the data by time: earlier ones are for training and later ones for testing. Because of this, our testing results are indeed “prediction” results; for the same reason, we did not set aside a third, separate dataset for the purpose of “more testing” as is sometimes done in the literature, as this purpose is already served by the second, test dataset.

The prediction results are summarized in the set of ROC (receiver operating characteristic) curves shown in Fig. 6. Recall that the RF classifier outputs a probability of incident for each input sample. To test its accuracy, a threshold is adopted that maps this value into a binary prediction: 1 if it exceeds the threshold and 0 otherwise. This binary prediction is then compared against the ground-truth: a sample from an incident dataset has a true label of 1, while a sample from the non-victim organization set has a true label of 0. Since our non-victim set for training (to balance) is randomly selected from the total non-victim population, the above test is repeated 20 times for a given threshold value, each time for a different random non-victim set. The average TP and FP over these repeated tests form one point on the ROC curve.

We see the prediction performance varies slightly between the datasets, but remain very satisfactory, generally achieving combined (TP, FP) values of $(90\%, 10\%)$ or $(80\%, 5\%)$. In particular, when we combine the three datasets, we can achieve an accuracy level of $(88\%, 4\%)$. A summary of some of the most desirable operating points are given in Table 5.

The above prediction results substantially outperform what has been shown in the literature to date; e.g., the web maliciousness prediction study in [51] reported a combination of $(66\%, 17\%)$ for (TP, FP). It is also worth pointing out that TP and FP values are independent of the sizes of the respective populations of the victim and non-victim organizations (these are conditional probability estimates), whereas the overall accuracy does depend on the two population sizes as it is the unconditioned probability of making correct predictions. Since based on our dataset we have a minuscule victim population (accounting for $\ll 1\%$ of the overall population), the overall accuracy is simply $\sim$ (1-FP). Therefore, if the overall accuracy is of interest, the best classifier would be a naive one that simply labels all inputs as “0”. This would lead to 0% TP, 0% FP, and an overall accuracy of $> 99\%$. However, despite achieving maximum overall accuracy, such a classifier is clearly useless. This point is also emphasized in [51] for similar reasons. Additionally, in the context of forecasting, where the goal is to facilitate preventative measures at an organizational level, having a high TP is perhaps more relevant than having a low FP; this is in contrast to spam detection, where the cost of FP is much higher than a missed detection. Therefore, the three measures in Table 5 should be taken as a whole.

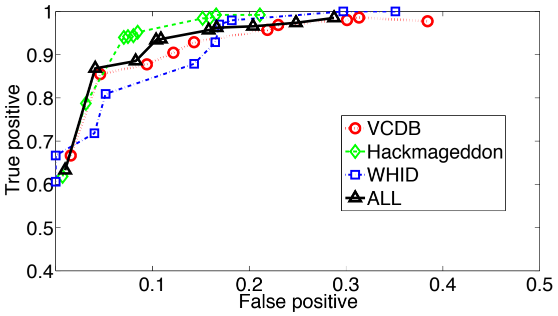


Figure 6: Prediction results. There are variations between the datasets, but an operating point – combined (TP, FP) values – of $(90\%, 10\%)$ or $(80\%, 5\%)$ is achievable. In particular, when we use all three datasets together, we can achieve an accuracy level of $(88\%, 4\%)$.

Accuracy	Hackmageddon	VCDB	WHID	All
True Positive (TP)	96%	88%	80%	90%
False Positive (FP)	10%	10%	5%	10%
False Negative (FN)	4%	12%	20%	10%
Overall Accuracy	90%	90%	95%	90%

Table 5: Best operating points of the classifier for the best combinations of (TP, FP) values.

4.2 Impact of Training:Testing Ratio

The results in Fig. 6 are obtained under a 50-50 split of the victim set into training and testing samples, based on the incident time. Furthermore, they are obtained using the short-term forecasting method described in Section 3.2. In general, one can improve the prediction performance by increasing the training sample size. There is no exception in our study, as shown in Fig. 7 where we compare results from a 70-30 training and testing sample split of the victim set to that from the 50-50 split, for the VCDB data. A best operating point is now around $(94\%, 10\%)$, indicating a clear improvement. Note that, a 70-30 split is not generally regarded high in the machine learning literatures, see e.g., in [57] a 90-10 split was used. We however believe a 50-50 split gives a more objective measure of the prediction performance.

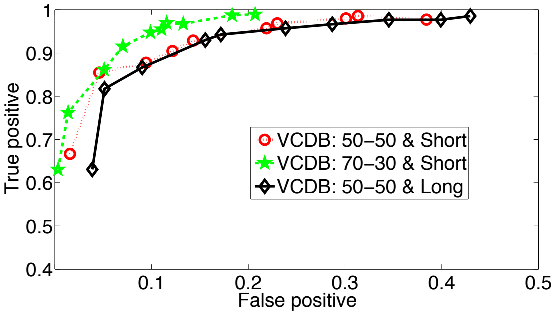


Figure 7: The impact of larger training set and size of forecasting window; all results are obtained using VCDB. The three curves: (1) using a 50-50 split of the victim set between training and testing under the short-term forecasting scenario (this curve is identical to the one in Fig 6); (2) using a 70-30 split of the victim set between training and testing under the short-term forecasting scenario; (3) using a 50-50 split of the victim set between training and testing under the long-term forecasting scenario.

4.3 Short-term vs. Long-term Forecast

Also shown in Fig. 7 are our long-term forecasting results under a 50-50 training and testing sample split of the victim set, again for VCDB. As seen, the prediction performance holds even when we move from a one-month to a 12-month forecasting window. The use of mismanagement symptoms and long-term malicious behaviors in the features contributes to this: they generally remain stable over time and have relatively high importance in the prediction, discussed in greater detail in the next section.

4.4 Relative Importance of the Features

In addition to the prediction output, the RF classifier also outputs a normalized relevance score for each feature used in training [17]; the higher the value, the more important the feature in the prediction. In this section, we examine these scores more closely. This study will further help us understand the extent to which different features determine the chance of an organization becoming breached in the near future. For brevity, the experiments presented in this section are based on a combination of all three datasets.

The importance of each category of features is summarized in Table 6. We make a number of interesting observations. First, note that the mismanagement features stand out as the most important category in prediction. Second, the Recent-60 secondary features are almost as important as the time series data, despite the fact that the former are derived from the latter. This is because the use of time series data has the effect of capturing synchronized behavior between malicious activities of different organizations, while the secondary features are aimed at capturing the dynamic behavior over time within an organization itself. That the latter adds value to the predictor is thus validated by the above importance comparison. Last but not least, the Recent-60 features appear much more important than Recent-14 features.

Feature category	Normalized importance
Mismanagement	0.3229
Time series data	0.2994
Recent-60 secondary features	0.2602
Organization size	0.0976
Recent-14 secondary features	0.02

Table 6: Feature importance by category. The mismanagement features are the most important category in prediction. Secondly, the Recent-60 secondary features are almost as important as the time series data; the former capture dynamic behavior over time within an organization whereas the latter capture synchronized behavior between malicious activities of different organizations.

A closer look into each category reveals that among the mismanagement features, untrusted HTTPS is by far the most important (0.1531), followed by Openresolver (0.0928), DNS random port (0.0469), Mail relay (0.0169), and BGP misconfig. (0.0132). The more significant role of untrusted HTTPS in prediction as compared to Openresolver is consistent with the bigger difference in distributions seen earlier in Fig. 2; that is, a victim organization tends to have a higher percentage of mis-configured HTTPS for their network assets. A possible explanation is that a majority of the incidents in our dataset are web-page breaches; these correlate with the untrusted HTTPS symptom, which reflect poorly managed web systems.

Similarly, a closer look at the secondary features (both Recent-60 and Recent-14) suggests that the dynamic features (duration and frequency together, totaling 0.1769) are far more important than static features (magnitude, totaling 0.0834). This suggests that dynamic changes over time, or in other words, organizations' response time in terms of cleaning up the origin of their malicious activities, is more indicative of security risks.

4.5 The Power of Dataset Diversity

A question that naturally arises is what if only a single feature category is used to train the classifier. For instance, given the prominent score of mismanagement features in prediction, would it be sufficient to only use these in prediction? The answer, as shown in Fig. 8, turns out to be negative. In this figure, we compare the prediction performance by using the following four categories of features separately to build the classifier: mismanagement, time series data, organization size, and the entire set of secondary features. While it is expected that using only one feature set leads to worse prediction performance, it is somewhat surprising that the secondary features are more powerful than mismanagement features or the time series when used separately. Recall that the secondary features were designed specifically to capture the organizational behavior, including their responsiveness and effectiveness in dealing with malicious acti-vities. One explanation of this result is that the human and process element of an organization is the most slow-changing compared to the change in the threat landscape, and thus holds the most predictive power.

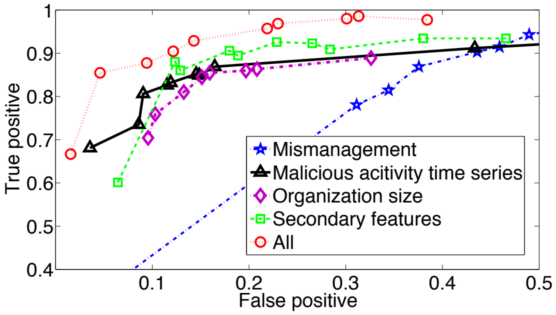


Figure 8: Independent prediction performance using only one set of features. The secondary features are shown to be the most powerful in prediction when used alone. Mismanagement features perform the worst, even though they have the highest importance factor. This is because the factors reflect conditional importance, given the presence of other features. This means that mismanagement features alone are poor predictors but they add valuable information to the other features.

Note that this is not inconsistent with the relative importance given in Table 6, as the latter is a measure of conditional importance of one feature given the presence of other features. In other words, the relative importance suggests how much we lose in performance if we *leave out* a feature, whereas Fig. 8 shows how well we do when using *only* that feature. What's seen here is that mismanagement features add very significant (orthogonal) information to the other features, but they are poor predictors in and by themselves. Perhaps most importantly, the results in Fig. 8 validate the idea of using a diverse set of measurement data that collectively form predictive descriptions of an organization's security risks.

4.6 Comparison with SVM

As a reference, we also trained classifiers using SVM; the prediction results are much poorer compared to using RF. For instance, using the VCDB data, the best operating point under SVM (with a 50-50 training-testing split of the victim population and short-term forecasting) is around (70%, 25%). This observation is consistent with existing literature, see e.g., [57].

5 Discussion

5.1 Top Data Breaches of 2014

In Fig. 9, we plot the distribution (CDF) of the predictor output values for the VCDB victim set and a randomly selected non-victim set used in testing. We use an example threshold of 0.85 for illustration. All points to the right of a threshold is labeled "1", indicating positive prediction, and all to its left "0". Three incident examples are also shown, falling into the categories of true-positive (ACME), false-positive (AXTEL), and false-negative (BJP Junagadh).

Also highlighted in Fig. 9 are the top five data breaches of 2014 [43], namely JP Morgan Chase, Sony pictures, Ebay, Home Depot, and Target. Using the suggested threshold value, our prediction method would have correctly labeled four of these incidents, and only narrowly missed the Target incident. It is worth noting that the Target incident was brought on by one of its contractors; however, the fact that Target did not have a more secure vendor policy in place is indicative of something else amiss (e.g., lack of consistent procedure between IT and procurement) that could also have manifested itself in the data and features we examined.

These examples highlight that, in addition to enabling proactive measures by an organization, there are potential business uses of the prediction method presented in this study. The first is in vendor or third party evaluation. Consider Online Tech, the hosting service used by JP Morgan Chase, as an example. As shown in Fig. 9, Online Tech posed very high security risks; this information could have been used in determining whether to use this vendor. Furthermore, information provided by our prediction method can help underwriters better customize terms of a cyber-insurance policy. The insurance

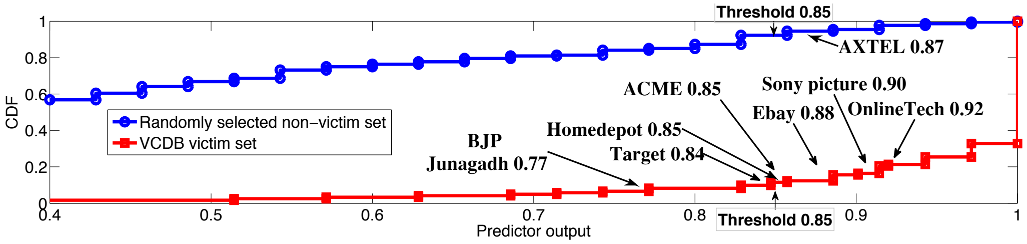


Figure 9: Distribution of predictor outputs with an example threshold value 0.85 (with 91% TP, 10% FP). On the curve with circles (non-victim) to the right of the threshold are FPs; on the curve with squares (victim) to the right of the threshold are TPs. Three types of incidents are shown, presenting true-positive (ACME), false-positive (AXTEL), and false-negative (BJP Junagadh). Also highlighted are top data breach events in 2014.

payout in the case of Target was reported to be around $60M; with this much at stake, it is highly desirable to be able to accurately predict the risk of an insured and adjust terms of the contracts.

5.2 Prediction by Incident Type

For a majority of the incident types, we do not have enough sample points to serve both training and testing purposes, except for the 368 reports of the type "web applications incident" in VCDB. This allows us to train a classifier to predict the probability of an organization being hit with a "web app incident". The corresponding results are similar in accuracy to those obtained earlier (e.g., at (92%, 11%)). This suggests that our methodology has the potential to make more precise predictions as we accumulate more incident data.

Similarly, the current forecasting methodology is not aimed at predicting highly targeted attacks motivated by geo-political reasons (e.g., the Sony Picture breach). Nor does it use explicit business sector information (e.g., a bank may be a bigger target than a public library system). In this sense, our current results represent more the likelihood of an organization falling victim provided it is being targeted. However, an ever increasing swath of the Internet is rapidly under cyber threats to the point that all major organizations should simply assume that they are someone's target. The use of explicit business sector information does allow us to make more fine-grained predictions. In a more recent study [48], we leverage a broad array of publicly available business details on victim organizations reported in VCDB, including business sector, employee count, region of operation and web statistics information from Alexa Web Information Service (AWIS), to generate risk profiles, the conditional probabilities of an organization suffering from specific types of incident, given that an incident occurs.

5.3 Robustness against adversarial data manipulation and other design choices

One design choice we made in the feature extraction process is the parameter δ which determines how a time series is quantized to obtain secondary features. Below we summarize the impact of of having different δ values. In the results shown so far, a value of $\delta = 0.2$ is used. In Fig. 10 we test the cases with $\delta = 0.1$ and $\delta = 0.3$. We see that this parameter choice has relatively minor effect: with $\delta = 0.3$ a desirable TP/FP combination is around $(91\%, 9\%)$, and for $\delta = 0.1$, we have $(86\%, 6\%)$. It appears that having a higher value of δ leads to slightly better performance; a possible explanation is that quantizing using $\delta = 0.2$ retained more noise and fluctuation in the time series, while quantizing using $\delta = 0.3$ may be more consistent with the actual onset of events.

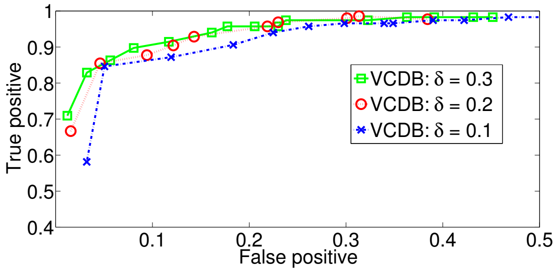


Figure 10: Experiment results under different δ.

Throughout the paper we have assumed the data are truthfully reported (though with noise/error). It is thus reasonable to question how robust is the prediction against possible (malicious) manipulation of the data used for training, a subject of increasing interest and commonly referred to as *adversarial* machine learning. For instance, an entity may attempt to set up fake networks with clean data (no malicious activities) but with fake reported incidents, and vice versa, to mislead the classifier. Without presenting a complete solution, which remains a direction of future research, below we test the robustness of our current prediction technique using two scenarios: (1) In the first we randomly flip the labels of victim organizations from 1 to 0; those flipped to 0 are now part of the non-victim group, thus contaminating the training data. (2) In the second scenario we do the opposite: randomly flip the labels of non-victim organizations, effectively adding them to the victim group. Incidentally, the former scenario is akin to under-reporting by randomly selected organizations.

Experimental results suggest no performance difference for case (1). The reason lies in the imbalance between the victim and non-victim population sizes. Recall that because of this, in our experiment we randomly select a subset of non-victim organizations with size comparable to the victim organizations (on the order of $O(1,000)$). Then in each training instance, the expected number of victims selected as part of the non-victim set is no more than $N_v \cdot O(1,000)/N$, with N_v denoting the number of fake non-victims and N the total number of non-victims. Since $N \sim O(1,000,000)$, even if one is able to inject $N_v \sim O(100)$ victims into the non-victim population, on average no more than one fake non-victim will actually be selected for training, resulting in negligible contamination effect unless such alterations can be done on a scale larger than the actual victim population.

For case (2), we indeed observe performance degradation, albeit slight, in the true positive, as shown in Fig. 11 at the 20% contamination level (20% of non-victim organization labels are flipped).

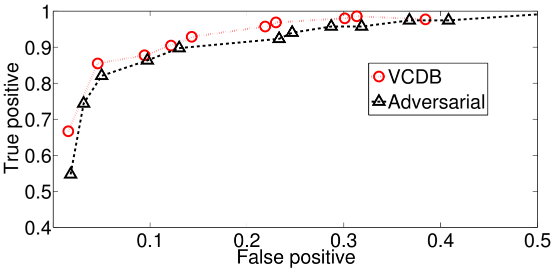


Figure 11: Adversarial case (2) with 20% contamination.

5.4 Incident Reporting

One of the main obstacles in studies of this nature is the acquisition of high quality incident data, without which we can neither train nor verify with confidence. Our result here demonstrates that machine learning techniques have the power to make accurate incident forecasts, but data collection is lagging by comparison. The research community would benefit enormously from more systematic and uniform incident reporting.

6 Related Work

As mentioned in the introduction, a large part of the literature focuses on detection rather than prediction. The work in [44] is one such example. Among others, Lee et al. [36] built sophisticated Hidden Markov Model techniques to detect spam deobfuscation, and in [57] Wang et al. applied (adversarial) machine learning techniques to the detection of malicious accounts on Weibo.

Relatively fewer studies have focused on prediction; even fewer are on the type of prediction presented in this paper where the predicted variable (classifier output) is of a different type from the input variables (feature input). For instance, the predictive IP-based blacklist works in [50, 30] have the same input and output variables (content of the blacklist). Similarly, in [54] the evolution of spam of a certain prefix is predicted using past spam activities as input. Predictive studies similar to ours include the aforementioned [51] that predicts whether a website will turn malicious by using textual and structural analysis of a webpage. The performance comparison has been given earlier. It is worth pointing out that the intended applications are also different: whereas webpage maliciousness prediction can help point to websites needing improvement or maintenance, our prediction on the organizational level can help point to networks facing heightened probability of a broader class of security problems. Also as mentioned earlier, our study [48] examines the prediction of incident types, conditional on an incident occurring, by using an array of industry, business and web visibility/population information. Other predictive studies include [28], where it is shown that by analyzing user browsing behavior one can predict whether a user will encounter a malicious page (attaining a 87% accuracy), [52], where risk factors are identified at the organization level (industry sector and number of employees) and the individual level (job type, location) that are positively or negatively correlated with experiencing spear phishing targeted attacks, and [53], where risk factors for web server compromise are identified through analyzing features from sampled web servers.

Also related are studies on reputation systems and profiling of networks. These include e.g., [26], a reputation assigning system trained using DNS features, reputation systems [8, 9] based on monitoring Internet traffic data, and those studied in [29, 46].

7 Conclusion

In this study, we characterize the extent to which cyber security incidences can be predicted based on externally observable properties of an organization's network. Our method is based on 258 externally measurable features collected from a network's mismanagement symptoms and malicious activity time series. Using these to train a Random Forest classifier, it is shown that we can achieve fairly high accuracy, such as a combination of 90% true positive rate and 10% false positive rate. We further analyzed the relative importance of the features sets in the prediction performance, and showed our prediction outcome for the top data breaches in 2014.

Acknowledgement

This work is partially supported by the NSF under grant CNS 1422211, CNS 1409758, CNS 1111699 and by the DHS under contract number HSHQDC-13-C-B0015.

References

[1] AFRINIC whois database. http://www.afrinic.net/services/whois-query.

[2] APNIC whois database. http://wq.apnic.net/apnic-bin/whois.pl.

[3] ARIN whois database. https://www.arin.net/resources/request/bulkwhois.html.

[4] Composite Blocking List. http://cbl.abuseat.org/.

[5] DShield. http://www.dshield.org/.

[6] Evernote resets passwords after major security breach. http://www.digitalspy.co.uk/tech/news/a462959/evernote-resets-passwords-after-major-security-breach.html.

[7] Global Reputation System. http://grs.eecs.umich.edu//.

[8] Global Security Reports. http://globalsecuritymap.com/.

[9] Global Spamming Rank. http://www.spamrankings.net/.

[10] Google Kenya and Google Burundi hacked by 1337. http://thehackersmedia.blogspot.com/2013/09/google-kenya-google-burundi-hacked-by.html.

[11] hpHosts for your pretection. http://hosts-file.net/.

[12] LACNIC whois database. http://lacnic.net/cgi-bin/lacnic/whois.

[13] Multiple DNS implementations vulnerable to cache poisoning. http://www.kb.cert.org/vuls/id/800113.

[14] Open Resolver Project. http://openresolverproject.org/.

[15] OpenBL. http://www.openbl.org/.

[16] PhishTank. http://www.phishtank.com/.

[17] Rf classifier. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[18] RIPE whois database. https://apps.db.ripe.net/search/query.html.

[19] SpamCop Blocking List. http://www.spamcop.net/.

[20] SURBL: URL REPUTATION DATA. http://www.surbl.org/.

[21] Syrian hacker Dr.SHA6H hacks and defaces City of Mansfield, OH website. http://hackread.com/syrian-hacker-dr-sha6h-hacks-and-defaces-city-of-mansfield-oh-website-for-free-syria.

[22] The SPAMHAUS project: SBL, XBL, PBL, ZEN Lists. http://www.spamhaus.org/.

[23] UCEPROTECTOR Network. http://www.uceprotect.net/.

[24] WPBL: Weighted Private Block List. http://www.wpbl.info/.

[25] Agrawal, T., Henry, D., and Finkle, J. JPMorgan hack exposed data of 83 million, among biggest breaches in history. http://www.reuters.com/article/2014/10/03/us-jpmorgan-cybersecurity-idUSKCN0HR23T20141003, October 2014.

[26] Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., and Feamster, N. Building a Dynamic Reputation System for DNS. In *Proceedings of the 19th USENIX Security Symposium* (Berkeley, CA, USA, August 2010).

[27] Bishop, C. M., et al. *Pattern Recognition and Machine Learning*, vol. 1. Springer New York.

[28] Canali, D., Bilge, L., and Balzarotti, D. On the Effectiveness of Risk Prediction Based on Users Browsing Behavior. In *ASIA CCS '14* (New York, NY, USA, June 2014), ACM, pp. 171–182.

[29] Chang, J., Venkatasubramanian, K. K., West, A. G., Kannan, S., Lee, I., Loo, B. T., and Sokolsky, O. AS-CRED: Reputation and Alert Service for Interdomain Routing. vol. 7, pp. 396–409.

[30] Collins, M. P., Shimeall, T. J., Faber, S., Janies, J., Weaver, R., De Shon, M., and Kadane, J. Using Uncleanliness to Predict Future Botnet Addresses. In *Proceedings of ACM IMC* (San Diego, California, USA, October 2007), pp. 93–104.

[31] Consortium, T. W. A. S. Web-Hacking-Incident-Database. http://projects.webappsec.org/w/page/13246995/Web-Hacking-Incident-Database.

[32] Durumeric, Z., Kasten, J., Bailey, M., and Halderman, J. A. Analysis of the HTTPS Certificate Ecosystem. In *Proceedings of ACM IMC* (Barcelona, Spain, October 2013), pp. 291–304.

[33] Durumeric, Z., Wustrow, E., and Halderman, J. A. ZMap: Fast Internet-Wide Scanning and its Security Applications. In *Proceedings of the 22nd USENIX Security Symposium* (Washington, D.C., August 2013), pp. 605–620.

[34] Hubert, A., and Mook, R. V. Measures for Making DNS More Resilient against Forged Answers. RFC 5452, January 2009.

[35] KREBS, B. The target breach, by the numbers. http://krebsonsecurity.com/2014/05/the-target-breach-by-the-numbers/, May 2014.

[36] LEE, H., AND NG, A. Y. Spam Deobfuscation using a Hidden Markov Model. In *In Conference on Email and Anti-Spam* (July 2005).

[37] LIAW, A., AND WIENER, M. Classification and Regression by randomForest. http://CRAN.R-project.org/doc/Rnews/, 2002.

[38] LINDBERG, G. Anti-Spam recommendations for SMTP MTAs. BCP 30/RFC 2505, 1999.

[39] MA, J., SAUL, L. K., SAVAGE, S., AND VOELKER, G. M. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, June 2009), KDD ’09, ACM, pp. 1245–1254.

[40] MAHAJAN, R., WETHERALL, D., AND ANDERSON, T. Understanding BGP misconfiguration. In *Proceedings of SIGCOMM ’02* (August 2002), vol. 32, ACM, pp. 3–16.

[41] OF OREGON, U. Route Views Project. http://www.routeviews.org/.

[42] PASSERI, P. Hackmageddon.com. http://hackmageddon.com/.

[43] PRINCE, B. Top data breaches of 2014. http://www.securityweek.com/top-data-breaches-2014, December 2014.

[44] QIAN, Z., MAO, Z. M., XIE, Y., AND YU, F. On Network-level Clusters for Spam Detection. In *Proceedings of the Network and Distributed System Security Symposium (NDSS ’14)* (San Diego, CA, March 2010).

[45] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the Network-level Behavior of Spammers. In *Proceedings of SIGCOMM ’06* (August 2006), vol. 36, ACM, pp. 291–302.

[46] RESNICK, P., KUWABARA, K., ZECKHAUSER, R., AND FRIEDMAN, E. Reputation Systems. *Commun. ACM 43*, 12 (December 2000), 45–48.

[47] ROMANOSKY, S. Comments on incentives to adopt improved cybersecurity practices noi. http://www.ntia.doc.gov/federal-register-notice/2013/comments-incentives-adopt-improved-cybersecurity-practices-noi, April 2013.

[48] SARABI, A., NAGHIZADEH, P., LIU, Y., AND LIU, M. Prioritizing Security Spending: A Quantitative Analysis of Risk Distributions for Different Business Profiles. In *the Annual Workshop on the Economics of Information Security (WEIS)* (June 2015).

[49] SIDEL, R. Home depot’s 56 million card breach bigger than target’s. http://www.wsj.com/articles/home-depot-breach-bigger-than-targets-1411073571, September 2014.

[50] SOLDO, F., A., L., AND MARKOPOULOU, A. Predictive Blacklisting as an Implicit Recommendation System. In *INFOCOM, IEEE* (March 2010), pp. 1–9.

[51] SOSKA, K., AND CHRISTIN, N. Automatically Detecting Vulnerable Websites Before They Turn Malicious. In *Proceedings of the 23rd USENIX Security Symposium* (San Diego, CA, August 2014).

[52] THONNARD, O., BILGE, L., KASHYAP, A., AND LEE, M. Are You at Risk? Profiling Organizations and Individuals Subject to Targeted Attacks. In *Financial Cryptography and Data Security* (January 2015).

[53] VASEK, M., AND MOORE, T. Identifying Risk Factors for Webserver Compromise. In *Financial Cryptography and Data Security*. Springer, March 2014, pp. 326–345.

[54] VENKATARAMAN, S., BRUMLEY, D., SEN, S., AND SPATSCHECK, O. Automatically Inferring the Evolution of Malicious Activity on the Internet. In *Proceedings of the Network and Distributed System Security Symposium (NDSS ’14)* (San Diego, CA, February 2013).

[55] VERIS. VERIS Community Database (VCDB). http://veriscommunity.net/index.html.

[56] VERIZON. Data Breach Investigations Reports (DBIR) 2014. http://www.verizonenterprise.com/DBIR/.

[57] WANG, G., WANG, T., ZHENG, H., AND ZHAO, B. Y. Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers. In *Proceedings of the 23rd USENIX Security Symposium* (San Diego, CA, August 2014), pp. 239–254.

[58] ZHANG, J., DURUMERIC, Z., BAILEY, M., KARIR, M., AND LIU, M. On the Mismanagement and Maliciousness of Networks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS ’14)* (San Diego, CA, February 2014).

APPENDIX

Incident Dataset

A snapshot of sample incident reports from VCDB dataset (Table 7).

Incident type	Time	Report summary
Web site defacement	May 2014	”ybs-bank.com” a Malaysian imitation of the real Yorkshire Bank website
Hacking	Apr. 2014	4chan hacked by person targeting information about users posting habits.
Web site defacement	N/A 2013	AR Argentina Military website hacked.
Server breach	N/A 2013	The systems of AdNet Telecom, a major Romania-based telecommunications services provider, have been breached.
Web site hacked	May 2013	Albany International Airport website hacked.
Private key stolen	Mar. 2014	Amazon Web Services, Inc.
Phishing	N/A 2013	Bolivian tourist site was compromised and a fraudulent secret shopper site was installed.

Table 7: Incidents from the VCDB Community Database