Phish-Net: Investigating Phish Clusters Using Drop Email Addresses

Shams Zawoad, Amit Kumar Dutta, Alan Sprague, Ragib Hasan, Jason Britt, and Gary Warner
{zawoad, adutta, sprague, ragib, jrbritt, gar}@cis.uab.edu

Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL 35294-1170

Abstract—The most common approach to collect users’ secret credentials from phishing websites is to email the credentials to criminals’ email addresses which we call drop email addresses. We propose a clustering algorithm, which is based on the assumption that if there is a common drop email address found in the phishing kits from two different phishing websites, then these two websites are directly related. Based on obfuscated and plain-text drop email addresses, we produce two types of clusters: one is called phishing kit creator cluster and another is kit user cluster. Clustering related phishing websites using our proposed approach will allow phishing investigators to focus their investigative efforts on important phishing attacks rather than random attacks. For example, in January 2013, 1475 phishing websites are hosted by only 317 groups of phishers (who we will call kit users). Our scheme will thus help investigators to narrow investigation to pervasive phishing criminals. By analyzing the clusters generated using our clustering approach, we can determine the strongest and most pervasive phishers, and phishing kit creators, relationships between phishing kit creators and phishing kit users, and the most dominant phisher of one group. These findings have real-life implication in phishing investigation paradigm.

Index Terms—Phishing, Cluster, Investigation

I. INTRODUCTION

Phishing is a social engineering attack that is a pervasive and ongoing problem over the Internet. Phishing websites look like legitimate websites, such as banks, product vendors, and service providers. Using these specially crafted fake websites, the phishers have deceived many users to provide private credentials, such as usernames and passwords, bank account numbers, and credit card numbers with pin codes. Phishing sites are advertised by spam emails to draw the users to phishing websites. The information collected through this process is used to withdraw money from the bank accounts and perform identity theft [1]. According to the recent RSA fraud report, global losses from phishing estimated at $1.5 billion in 2012, which is a 22% increase from 2011 [2].

Although more users are currently aware of phishing threats, unfortunately, the phishing attacks have increased and continue to exist. As mentioned in the Anti-Phishing Working Group (APWG) survey for the first half of 2012, there were at least 93,000 unique phishing attacks using approximately 64,000 unique domain names worldwide [3], which is an increase compared to the second half of 2011. Another report says that total number of phishing attacks launched in 2012 was 59% higher than 2011 [2]. One of the reasons behind the escalation of these attacks is that the criminals behind phishing attacks are not afraid of being prosecuted. One way to deter criminals is to increase successful prosecution, which has not increased partially because of a lack of capable techniques and tools [4]. Therefore, there is a need for automated solutions to correlate phishing evidence from different sources, which will help investigators to understand their data and to prioritize investigations. Important evidence or data can be found within ‘phishing kits’, such as the drop email addresses that receive the private credentials [5].

Phishing kits are zip files that are used to store all the files and directory structures necessary to create a phishing website. We assume that a phishing kit found in the phishing domain is used to create the phishing website. Generally, these kits are built in such a way that whenever a victim provides his secret credentials to phishing websites, that information will be emailed to some email addresses (drop email address), which can be found in the phishing kits. These kits are developed by technically adept kit creators. A kit user just need to unzip the kit and provide his drop email addresses to successfully deploy a phishing site. Kit creators also want to harvest the same credentials, so they hide their drop email addresses to receive the secret credentials covertly [6]. Hence, the drop email addresses found in kit appear either in obfuscated or plain-text form. Usually, criminals use multiple email addresses, so that shutting down one email address will not affect their phishing operation. These email addresses play a vital role in investigation process and can lead to two important steps during phishing investigations: First, the email account provider (e.g., Google, Yahoo, Microsoft) can provide login history of an email account, potentially revealing the criminal’s IP address. Second, while a phished organization realizes it has lost some money due to phishing attacks, that organization cannot associate a given phishing website to a precise volume of financial loss [7]. Therefore, by reviewing the email records of the criminal, the names and bank account numbers of victims can be linked to the phishing websites where the victimization occurred. This allows the phished institution to quantify its financial loss due to phishing attacks. However, focusing on all the email addresses of all the phishing websites decelerates the investigation process. Moreover, there can be groups of phishers or single individual perpetrating the phishing attacks [8]. Hence, finding clusters
of related phishing websites, and from the clusters, finding predominant email addresses of phishing kit creators and kit users are crucial for forensics investigation.

In this paper, we propose a scheme for clustering phishing websites, based upon the assumption that if there is a common recipient/drop email address between two websites, then those two websites are related and created by same group of phishers or kit creators. Using this approach, some websites can be directly related, while some websites can be transitively related. We argue that all the phishing websites, whether they are directly or transitively related, form a cluster of related phishing websites. Based on obfuscated and plain-text drop email addresses, we produce two types of clusters. A cluster of phishing websites generated using obfuscated email addresses means that the phishing kits behind those websites are created by same group of kit creators. Similarly, a cluster of phishing websites generated using plain-text email addresses means that those phishing websites are deployed by same group of phishing actors. While generating these clusters, we maintain a trace of cluster generation. From these traces, we can identify the predominant email addresses of kit creators and kit users. Our method has been evaluated against a collection of phishing sites in the UAB Phishing Data Mine [9].

Contribution. The contributions of this paper are as follows:

1) We explore a scheme to cluster phishing websites based on common recipient/drop email addresses.
2) We introduce the notion of a phish-net graph. We keep the trace used to generate this graph, using which, we can identify the most dominant phishing campaign in terms of number of phishing websites, time span, and cluster strength factor.
3) From the phish-net, we identify the most active phishing kit creator and kit users behind a cluster of phishing websites.
4) We identify the relationship between phishing kit creator and kit user clusters to show patterns in kit creator and kit user interactions.

Organization. The rest of the paper is organized as follows. Section II provides some background and motivation for our work. In Section III, we provide statistics about the data set on which we run our clustering process. Section IV presents the clustering algorithm, visualization of phish-net, and evaluation of the algorithm. In Section V, we present our important findings by analyzing the generated clusters. Section VI provides the contemporary studies on clustering phishing websites. Section VII provides concluding thoughts and future work.

II. BACKGROUND AND MOTIVATION

This section describes some important terms in the phishing domain to better understand our work. We also present the motivation behind this work.

A. Background

- Phishing kit: A Phishing kit is an archive file, which contains all the necessary files to create a phishing page, such as HTML files, graphics files, style sheet files, or JavaScript files. Kit creators create a phishing kit targeted for a particular brand and distribute their kits to phishing actors, who are commonly known as phishers. Phishers use these kits to create phishing websites. A phishing kit can contain files such as PHP or other scripting files to implement the functionality of emailing the stolen personal information to phishers. Kit creators can also launch a phishing attack by using their own kits.

- Drop email address: Phishers use the phishing websites to collect users private credentials, which are usually sent to one or more email addresses of phishers. These email addresses are called drop email addresses. In this research, we use this drop email address to build a relationship between phishing websites. Only the phishing websites for which at least one drop email address is found can be clustered using our clustering method.

- Obfuscated drop email address: Phishing kit creators do not provide their kits to phishers benevolently [8]. Besides selling their kits to phishers, they also collect users’ credentials through drop email. However, most often they keep their email addresses obfuscated to hide their trace in phishing kits. Some common obfuscation techniques are Base64, Hex, and NUXI. These obfuscated email addresses are hidden inside kits in different forms, sometimes even inside image files. Besides emailing users’ secret credentials to the drop email addresses inserted by the phishers, the kit also sends email to these hidden email addresses.

- Plain-text drop email address: After obtaining phishing kits from creators, kit users place their email addresses in scripting files to acquire users’ credentials. These drop email addresses are mostly found in plain-text format.

- Kit creator cluster: As obfuscation is highly coupled with the kit creation process and obfuscation is an unnecessary step for kit users, obfuscated emails addresses are assumed to be tied to kit creators, who should have sufficient knowledge to apply a obfuscation technique. Hence, clustering phishing websites on the basis of obfuscated drop email addresses gives us the phishing websites created by same group of creators. We denote such cluster of phishing websites as kit creator clusters. By analyzing these clusters, we can identity a predominant creator.

- Kit user cluster: As the general kit users do not have sufficient knowledge to hide their email addresses by using an obfuscation technique, the plain-text email addresses usually refers to a kit user. Hence, clustering phishing websites by using plain-text drop email addresses creates clusters of phishing websites deployed by same group of kit users. We call these clusters as kit user clusters. By analyzing these clusters, we can identity a predominant phish kit user.

B. Motivation

Researchers have been improving their ability to identify phishing websites [10]–[12], so these websites can be blocked
by Internet Service Providers, or web browsers. There are other defensive mechanisms, such as preventing a phishing attack from reaching its intended recipient [13], [14]. However, it has been shown that there is a need to better protect victims from phishing attacks, as blacklist are not working [15]. Phishing attacks continue to exist since victims are not appropriately warned or blocked from the attacks [16], webmasters are not informed of vulnerable applications [17], and criminals (phish kit creators & users) are not prosecuted partially due to lack of capable techniques and tools [4]. Sheng et al. interviewed several phishing investigators and experts on the current problems in phishing. Two of the interviewees mentioned that there is plenty of data for investigations, but they do not have the tools and resources available to analyze the data [4]. Hence, providing algorithms and tools that can help investigators in their phishing investigation process is a necessity.

Investigators can receive benefits from a clustering algorithm, which provides them the ability to prioritize their phishing investigations. Each month, UAB Phishing Lab finds more than 20,000 phishing websites. From January 2013 to March 2013, the number of drop email addresses found in 3,764 phishing websites is 20,295. Focusing on large numbers of phishing websites and drop email addresses is tedious and requires significant amount of human resources. A well-defined clustering algorithm can help the investigators to narrow investigations and can help them to identify the most active kit creators and kit users to mitigate phishing attacks. Creation of a clustering algorithm using the drop email addresses may supply knowledge of cross-brand attacks showing the same phisher attacking multiple brands.

III. DATA SET

An anti-phishing framework was developed at UAB to detect phishing websites rapidly and gather phishing evidence. The data set for this research is from the UAB Phishing Data Mine [9] from the first quarter of 2013. This data set consists of 203,979 potential phishing websites collected from a large spam-based URL provider, a large anti-phishing company, and a number of other feeds including private companies, security companies, and financial institutions. The source of the suspected phishing website URLs is either URLs contained in spam or URLs reported by the public to fraud alert email addresses.

UAB Phishing Data Mine lab applies three automatic ways to confirm phishing websites: applying Deep MD5 technique [18], applying Abstract Syntax Tree [19], and comparing main page hashes of suspected phishing websites with confirmed phishing websites. If the automated processes do not confirm a suspected phish, it is manually reviewed and determine to be a phish or a legitimate website. A suspicious phishing URL that cannot pass the above confirmation procedures will be treated as unconfirmed.

Phishing kits are fetched using an automated web crawler that uses GNUs wget [20]. After confirming a suspected phishing website as a phish (automated or human confirmation), the phish is scanned for phishing kits using known phishing kit names. A select number of phish are manually scanned for phishing kits.

Table I shows some statistics about the data set for January 1st 2013 to March 31st 2013.

<table>
<thead>
<tr>
<th>Category</th>
<th>January</th>
<th>February</th>
<th>March</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of suspected phish</td>
<td>107,408</td>
<td>66,784</td>
<td>29,797</td>
</tr>
<tr>
<td>Number of confirmed phish</td>
<td>28,422</td>
<td>22,564</td>
<td>20,364</td>
</tr>
<tr>
<td>Number of phish having kit</td>
<td>2,939</td>
<td>1,687</td>
<td>493</td>
</tr>
<tr>
<td>Number of phish with drop email addresses</td>
<td>2,113</td>
<td>1,229</td>
<td>422</td>
</tr>
<tr>
<td>Number of drop email addresses before data cleansing</td>
<td>26,735</td>
<td>11,865</td>
<td>146,961</td>
</tr>
</tbody>
</table>

TABLE I: Data Statistics

From the data set, we notice that the percentage of phish for which we find drop email address is not high. As seen from table I, the first reason is number of phishing websites having phishing kit is low. There are several reasons behind this: kit users can delete the kit after using that kit, and while fetching the phishing website’s content we cannot browse all the directories due to the security settings of the host and thus fail to collect the kits. However, on 77% of phishing websites having a phishing kit we found drop email addresses. Another reason for getting low number of phish with drop email addresses is that there are some other ways of collecting users’ credentials, where there is no need of drop email addresses. Phishers may use FTP to send users’ credentials or a kit can even save credentials to a database. Our scheme can cluster only those phishing websites for which we can find drop email addresses.

A. Data Cleansing

As we notice from Table I, number of email addresses is quite large compared to number of phish with drop email addresses. After further investigation, we found that not all of the email addresses extracted from phishing kits are being used as drop email address. Hence we apply the following data cleansing techniques to reduce the number of non-drop email addresses:

- **Removing email addresses for unconfirmed phishing websites:** Not all the suspected phishing sites collected from various sources are genuine phishing websites. We do not consider the email addresses from unconfirmed phishing websites. We can remove 0.7% of the email addresses from the 3-month duration using this criterion.
- **Removing white listed email addresses:** There are some email addresses that are common in almost all the phishing sites of a particular brand. For example, new@paypal.com appears in almost all the phishing websites of PayPal. Executing our clustering process using this email address will give us a fat cluster for PayPal brand centered on new@paypal.com. However, this email address is not used as a drop email address. On the contrary, this email address is used as a sender’s email address to email user’s credentials to the drop email addresses. These types of email addresses are marked as...
white listed emails. Another type of white listed emails is found from different JavaScript and CSS library files. These libraries are made for lawful uses, but sometimes kit creators use these to create phishing kits. Library creators often list their contact email addresses in their library files. When kit creators use these libraries to build their phishing kits, the library creators’ email addresses are also collected by the email address extractor. We remove all the email addresses, which fall into white listed category. There are 3.44% email addresses that fall into this category.

- **Removing Spam email addresses**: Email addresses are extracted from various files of a phishing kit. The drop email addresses are mostly found in different scripting files and sometimes they can be even embedded with images. However, there are other email addresses that are found in text files containing hundreds of thousands of email addresses. These email addresses are spam lists that are not used for collecting users’ private credentials. Identifying clusters using these email addresses will not direct us to find the phishing actors and creators. Hence, we remove all the suspected spam email addresses by excluding email addresses found in text files. With this criterion, 77.29% of total email addresses are excluded.

Table II summarizes the number and percentage of email addresses found in these different categories. After removing all the unwanted email addresses, 10.95% of the original email addresses (i.e. 20,295 email addresses) remain in our working set. However, as we notice from Table III, the number of email addresses is not uniformly distributed in different months

<table>
<thead>
<tr>
<th>Category</th>
<th>No of Email</th>
<th>% of Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unconfirmed</td>
<td>1,291</td>
<td>0.7</td>
</tr>
<tr>
<td>White listed</td>
<td>6,376</td>
<td>3.44</td>
</tr>
<tr>
<td>Spam</td>
<td>143,423</td>
<td>77.29</td>
</tr>
<tr>
<td>Working set</td>
<td>20,295</td>
<td>10.95</td>
</tr>
</tbody>
</table>

**TABLE II: Statistics of Email Addresses**

<table>
<thead>
<tr>
<th>Category</th>
<th>January</th>
<th>February</th>
<th>March</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unconfirmed</td>
<td>1,055</td>
<td>186</td>
<td>0.93</td>
</tr>
<tr>
<td>White listed</td>
<td>4,744</td>
<td>1,001</td>
<td>631</td>
</tr>
<tr>
<td>Spam</td>
<td>82</td>
<td>23</td>
<td>143,318</td>
</tr>
<tr>
<td>Working set</td>
<td>11,642</td>
<td>5,640</td>
<td>3,013</td>
</tr>
</tbody>
</table>

**TABLE III: Statistics of Email Addresses By Month**

Finally, we split the data set according to obfuscation type. The obfuscated email addresses are used to find clusters of phishing websites, which are created by same group of phishing kit creators. The plain-text email addresses are used to find clusters of phishing websites, which are deployed by same group of kit users. Table IV provides statistics of phishing websites for plain-text and obfuscated email addresses.

**IV. BUILDING THE PHISH-NET**

To create a cluster of related phishing websites, we rely on the direct and transitive relationship between two phishing websites, which is defined based on drop email addresses. The direct relationship between two phishing websites is defined as follows:

**Definition 1**: A phishing website X is directly related with a phishing website Y, if there is a common drop email address found in the phishing kits of these two websites.

Using the direct relationship between two phishing websites, we define the transitive relationship as follows:

**Definition 2**: A phishing website X is transitively related with another phishing website Z, if X is directly related with a phishing website Y and Y is directly related with Z.

The transitive relationship defined above can be used to define multi-level of transitivity.

**Definition 3**: A phishing website X is directly/transitively related with a phishing website Y and Y is directly/transitively related with a phishing website Z, then X and Z are transitively related.

All the phishing websites that are directly or transitively related form a cluster of related phishing websites. In the phish-net graph, we represent the phish as nodes on the graph and connect two directly related phish nodes via a link to a common drop email address node. In a cluster, any node is reachable from all other nodes.

<table>
<thead>
<tr>
<th>Category</th>
<th>January</th>
<th>February</th>
<th>March</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct.</td>
<td>930</td>
<td>196</td>
<td>1.93</td>
</tr>
<tr>
<td>White</td>
<td>4,744</td>
<td>1,001</td>
<td>631</td>
</tr>
<tr>
<td>Spam</td>
<td>82</td>
<td>23</td>
<td>143,318</td>
</tr>
<tr>
<td>Working</td>
<td>11,642</td>
<td>5,640</td>
<td>3,013</td>
</tr>
</tbody>
</table>

**TABLE IV: Phish and Drop Email Addresses Distribution by Obfuscation type**

**A. Clustering Process**

Based on common drop email addresses, the clustering process merges directly/transitively related phishing websites hierarchically. Each cluster consists of a list of phish, a list of drop email addresses, a list of brands, and a list of traces.
Algorithm 1 fetches all the related drop email addresses corresponding to a phish, and merges phish into a cluster, if they share at least one drop email address. This is an Agglomerative Hierarchical Clustering algorithm [21], where at the beginning, each phish itself is a cluster and merges into another cluster based on the common drop email address. Therefore, two clusters cannot share a common drop email address. We call the combination of an email address and a phish pair a Trace, if that drop email address is shared between the two phish. Algorithm 1 keeps track of traces while merging multiple clusters. The trace list provides interesting insights about the clusters, which are discussed in section V.

Algorithm 1 Cluster & Trace Generation

1: \( \text{emailsmap} \leftarrow \text{HashMap} << \text{String}, \text{Tuple} << \text{Integer} >\), List < Phish >>
2: \( \text{clustersmap} \leftarrow \text{HashMap} < \text{Integer}, \text{Group} >\)
3: for all phish in Phish Table do
4: \( \text{previousclusters} \leftarrow \text{List} < \text{Integer} >\)
5: \( \text{traces} \leftarrow \text{List} < \text{Trace} >\)
6: for all email related to phish in Phish Table do
7: if email in emailsmap then
8: get previous cluster ID from emailsmap and add this ID to previousclusters
9: add current phish to the phish list of email
10: generate new trace and add them to traces
11: end if
12: end for
13: if previous clusters found then
14: get all the previous clusters from previousclusters and clustersmap
15: merge all previous clusters into one cluster
16: add traces to the newly generated cluster
17: update emailsmap with new cluster ID
18: add new cluster to clustersmap
19: else
20: crate a new cluster
21: add traces to the newly generated cluster
22: add new cluster to clustersmap
23: end if
24: end for
25: return clustersmap

The algorithm maintains two maps for faster computation. First one is \( \text{emailsmap} \), which is a HashMap to map a drop email address with its cluster ID and related phish. Second one is \( \text{clustersmap} \), another HashMap that maps a cluster ID with the corresponding cluster. Algorithm 1 scans the set of drop email addresses related to a phish and detects the existence of these email addresses in previous clusters. If some of these drop email addresses appeared in other clusters before, the algorithm finds those clusters using \( \text{emailsmap} \) and \( \text{clustersmap} \). Then, it breaks the previous clusters and creates a new one with all the new and previous email addresses and phish. This process is called merging of multiple clusters. As showed in Figure 1, the merging process can be one or multi-way. For one way merging, it is possible to keep the cluster ID unchanged because only new phish and email addresses are added to the cluster. However, for multi-way merging, we need to break a number of previous clusters and provide a new ID to the new cluster. As multi-way merging is more general case, algorithm 1 always perform multi-way merging. After the merging finishes, algorithm 1 updates the \( \text{emailsmap} \) and clustersmap with the information of new cluster. It also creates the trace of merges during the detection of previous existence of email addresses. These new traces of merge, along with the old traces from previous clusters are added to the newly generated cluster.

Algorithm 1 uses a hierarchical approach to define the basic scheme for merging related clusters based on phish. Performance of this algorithm is discussed at Section IV-C. This algorithm deals with high volumes of data and requires database access to fetch email addresses corresponding to a phish. We employ a simple technique to minimize database access for faster cluster generation. We call this technique Cache-based Clustering. In the cache-based approach, the algorithm serializes the generated clusters into xml files for future use. If the algorithm is run again for \( n \) days, it first loads all the clusters for \( k \) days \((n > k)\) from xml files. Clusters for \( k \) days are already available from previous computation. Algorithm 2 takes two parameters, \( n \) and \( k \). \( n \) denotes the overall time span for cluster generation and \( k \) specifies the previous \( k \) days’ cluster is available in file. This technique is very useful when the algorithm is run for consecutive days and reduces running time significantly.

Algorithm 2 Cache Based Clustering (n,k)

1: load previous \( k \) days cluster from file
2: run Algorithm 1 for \( n - k \) days

Fig. 2: Visualizing Phish-Net of Kit Creator Cluster for January 2013

B. Visualizing the Phish-Net

The clustering algorithm defined in section IV-A merges two phishing websites into a cluster when a common drop email address between the phish is found. The merging trace
is preserved and used to produce the phish-net. An example of Trace is represented in Table V

```xml
<Trace>
  <email>brain00@hotmail.fr</email>
  <Phish1>
    <Phish>
      <phishId>2931550</phishId>
      <phishName>http://ermitaj.com.ua/pp2/Confirm.htm</phishName>
      <brand>PayPal</brand>
      <firstDate>2013-02-09 06:00:00.0 UTC</firstDate>
    </Phish>
    <Phish2>
      <phishId>2922089</phishId>
      <phishName>http://remoil.ru/tmp/pp2/</phishName>
      <brand>PayPal</brand>
      <firstDate>2013-02-06 06:00:00.0 UTC</firstDate>
    </Phish2>
  </Phish1>
</Trace>
```

**TABLE V: A Sample Trace**

In this structure, *Phish1* and *Phish2* represent two phish, and the *email* field contains common drop email addresses between two directly related phish. Algorithm 1 describes the algorithm for preserving trace while merging.

Lists of traces can be plotted in a graph, where nodes represent phish and drop email addresses. One trace entry connects the two phish with one drop email address. Thus each trace is represented by 2 undirected edges in the graph. The resulting graph is a Phish-Net graph. Figure 2 illustrates the phish-net graph for January 2013 for obfuscated email addresses.

By cluster visualization, we can identify the largest cluster, cross-brand clusters, and dominant phishers and creators. For example, Figure 3 represents a cross-brand creator cluster of January 2013 with Bank of America, Chase Bank, and PayPal brands.

![Fig. 3: Multi-brand Cluster](image)

**C. Evaluation**

We clustered the phishing websites in monthly time windows, since a longer time span would require a large amount of compute time. We evaluated the performance of our algorithm using a laptop computer with Intel Core I7 quad core CPU, 16 GB ram, and 750 GB hard drive. The clustering algorithm was developed in Java. To evaluate the feasibility of our clustering algorithm, we ran the algorithm on January 2013 data by 5 day intervals. Figure 4 shows the performance evaluation of the clustering algorithm, with the increase in number of drop email addresses, number of phish, and number of clusters. From this figure, we notice that the number of phish and number of drop email addresses grow almost linearly over days. We gain significant performance improvement while using the cache based clustering algorithm.

![Fig. 4: Performance Evaluation](image)

**Table VI** represents some of the important statistics about clusters after running the algorithm on several months' phishing data. For every month, we notice that the average number of brand per cluster is greater than 1 but the cross-band cluster is not very high in number. Hence, it is clear that if there is a cross-brand cluster, then most likely it contains a significant number of brand's phish. Also maximum number of drop email addresses in one cluster indicates the need of finding predominant drop email address of a cluster.

**V. ANALYZING THE CLUSTERS**

From the clustering scheme, several important pieces of information are gathered for a given time duration:

- The most dominant phish cluster.
- The most active kit creator and kit user.
- Relationship between kit creator, kit user, and brand.

In this section, we describe each of these findings, which have real-life implications for phishing investigations.
<table>
<thead>
<tr>
<th>Information</th>
<th>January</th>
<th>February</th>
<th>March</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obfuscated</td>
<td>Plain-text</td>
<td>Obfuscated</td>
</tr>
<tr>
<td>Total no. of cluster</td>
<td>31</td>
<td>317</td>
<td>39</td>
</tr>
<tr>
<td>Maximum no. of phish</td>
<td>51</td>
<td>135</td>
<td>52</td>
</tr>
<tr>
<td>Maximum no. of email addresses</td>
<td>22</td>
<td>541</td>
<td>16</td>
</tr>
<tr>
<td>Maximum no. of brand</td>
<td>3</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>Average phish count/cluster</td>
<td>5.94</td>
<td>4.07</td>
<td>6.03</td>
</tr>
<tr>
<td>Average email address count/cluster</td>
<td>3.0</td>
<td>4.60</td>
<td>2.67</td>
</tr>
<tr>
<td>Average brand count/cluster</td>
<td>1.26</td>
<td>1.19</td>
<td>1.38</td>
</tr>
<tr>
<td>Average candidate email count/cluster</td>
<td>1.5</td>
<td>1.56</td>
<td>1.43</td>
</tr>
<tr>
<td>No. of single phish cluster</td>
<td>13</td>
<td>167</td>
<td>16</td>
</tr>
<tr>
<td>No. of cross-brand cluster</td>
<td>7</td>
<td>38</td>
<td>8</td>
</tr>
</tbody>
</table>

**TABLE VI: Cluster Statistics**

(a) Finding Most Dominant Kit Creator Cluster  
(b) Distribution of Phish for Kit User Cluster  
(c) Distribution of Drop Email Address for Kit User Cluster

Fig. 5: Analysis of Clusters by Number of Phish and Drop Email Address

A. Identify the Most Dominant Cluster

We identify the most dominant cluster for a given duration with respect to the number of phish and number of email addresses, time span, and a cluster strength factor.

**Number of phish and email addresses.** Clusters containing larger numbers of phish are the most important clusters to investigate. From the kit creator clusters, we can identify which group is distributing the highest number of kits as the kits are used to build a phishing website. From the clusters generated using plain-text email addresses, we can identify which group is hosting the highest number of phishing websites. The number of drop email addresses in a cluster gives an idea about how quickly drop email addresses change. A lower number means the kit creators or users are not changing their email addresses frequently. On the contrary, a higher number of email addresses means that the kit creators and kit users are changing their email addresses frequently. Figure 5a represents the number of phishing websites and email addresses for the kit creator clusters generated from the obfuscated drop email addresses found in January 2013. Because single phish clusters are not significant, we remove them to better visualize the graphs, and use the same technique in later figures. From figure 5a, we can clearly identify the most dominant cluster. Cluster number 16 is the most dominant cluster in terms of number of phish (51 phish) and cluster 13 is the most dominant cluster in terms of number of drop email addresses (22 drop email addresses). Figure 5b represents the distribution of phish among the kit user clusters. We observe that most of the clusters contain small number of phish while very few clusters contain large number of phish. Hence, we can devote our effort to these few clusters to further analyze. The same behavior exists for drop email address distribution among the kit user clusters, which is illustrated in Figure 5c.

**Time span of clusters.** The Time span of a cluster indicates how long a cluster survives. From the time span of cluster, we can find the most active kit creator or kit user. A cluster with a long time span indicates that the people behind the creation of phishing kits or the phishing attacks remain undetected for a long time period. Hence, finding the clusters with long time span is important for phishing investigation. Figure 6a represents the time line of kit creator clusters for January 2013. Cluster number 18 was active during the whole month of January. In terms of time span, this is definitely the most important kit creator cluster during January 2013 and we need to investigate this cluster further. Figure 6b represents the time line of kit user clusters for January 2013. Cluster number 56 and 57 have the longest time span, from January 1st to January 31st. We also notice that average duration for kit creator clusters is 18.4 days, whereas average duration for kit user clusters is 10 days. One possible reason behind this is that kit creators are able to create a long chain of transitive relation by distributing their kit over longer time period.

However, a cluster spanning 31 days does not mean that there is a phishing website found in every day. After an attack,
the phishers or creators may remain dormant for few days and start with some new drop email addresses along with some of the previous addresses that remain unblocked. Hence, duration of clusters does not always indicate the most active phishers. Figure 7 illustrates the comparison between duration and number of phishing websites for both kit creator and kit user clusters. We notice that some clusters have long time spans but there are few phishing websites found during the lifespan of the clusters. On the other hand, there are some clusters, which have short lifespan but contain large number of phish. The later is a stronger attack scenario than the previous one.

We further investigate the detailed time line of the two most important clusters found using plain-text email addresses. The first cluster is cluster 102 with the highest number of phish and another one is cluster number 57 with the longest time span. Figure 8 represents the number of phishing websites found each day for these two clusters. For cluster 57, the number of phish varies between 1 and 2. On the other hand, for cluster 102, we see some burst of phish on particular days. Hence, cluster no 102 is more important for forensic investigation than cluster no 57 as it represents a stronger phishing attack.

**Strength of a cluster.** Strength of a cluster suggests how closely the phishing websites are related. Figure 9 shows two clusters of different strength. Here, we only show the phish
nodes from the phish-net graph. In this figure, cluster 2 has more strength than cluster 1. We believe that a cluster with higher strength suggests that there is a high probability that the phishing websites are operated by same group of phishers. We denote the strength of a cluster by Strength Factor $SF$. $SF$ of a cluster is defined as the ratio of the number of edges to the number of vertices in the cluster. An edge of Figure 9 is represented by a Trace entry, and vertices are represented by phish entries. Hence, we can formalize the $SF$ as following equation:

$$SF = \frac{NT}{NP}, \tag{1}$$

where $NT$ represents the number of traces in one cluster and $NP$ represents the number of phish in one cluster. Using the above equation, $SF$ of cluster 1 is 0.8, while the $SF$ of cluster 2 is 1.6. Hence, clusters with higher $SF$ provide us the cluster with higher correlation between phishing websites.

Figure 10 represents the $SF$ of each creator clusters found from January 2013. From this figure we can easily determine the most dominant kit creator cluster in terms of $SF$.

B. Identify the Most Active Kit Creator and Kit User by Candidate Email Address

After identifying the most dominant clusters, the next important task is to identify the most active creator and phisher who are behind these predominant clusters. From Figure 5, we notice that the number of drop email addresses in some important clusters is large. Finding the phishers behind all of the email addresses will be tedious for investigators. Hence, finding the most important drop email address of one cluster and tracing back the owner of that email address will expedite the investigation process. We denote the most important email address of a cluster by the term Candidate Email Address.

With the help of phish-net graph, we can identify the candidate email address of one cluster.

Figure 11, presents one kit creator cluster of January 2013. From this figure, it is clear that one email address connects more number of phish than other email address. The candidate email address of this cluster should be the email address that connects highest number of phish. In other words, the email address node with highest branching factor is the candidate email address.

Candidate email addresses for a cluster can be identified from the cluster’s trace list. One trace entry is used to connect two phish by one email. Hence, the number of phish that an email address connects, depends on the number of traces in which the email address has appeared. The more traces containing the email address, the higher the branching factor for an email address. From this assumption, we define the candidate email address of a cluster as follows:
Definition 4: A candidate email address of a cluster is the email address that appears maximum number of times in the trace list of that cluster.

![Fig. 11: Finding Candidate Email Address](image)

In a Trace list, multiple email addresses can appear same number of times. Therefore, for one cluster, there can be more than one candidate email addresses. The algorithm to identify the candidate list from the Trace list is described in Algorithm 3. It creates a temporary map of distinct drop email addresses found in the trace list and their frequency. After that, it sorts the map based on frequency and adds the most appeared email address to the candidates list.

We may have multiple candidates from one cluster. Even though, as noted in Table VI, the average number of candidate email address is less than the average number of email address in one cluster. The maximum number of drop email addresses that we found in a single cluster for kit creator cluster is 22 and for kit user cluster it is 541. However, for these clusters there is only one candidate email address. Hence, without tracing the owner of 542 email addresses, tracing only one email address will expedite the phishing investigation process.

C. Relationship Between Kit Creator and Kit User Cluster

Identifying relationships between clusters found using obfuscated email address, and clusters found using plain-text email address gives the relationship between kit creators and kit users. From this relationship, we can identify, which groups of creators are distributing phishing kits to which groups of kit users. This finding is crucial for phishing investigation. One kit user cluster is related with an creator cluster if there is a common phishing website between these two clusters. On the basis of this assumption, we find relationships between kit creator and kit user clusters for a given time duration. Figure 12 represents the relationship between two types of clusters during January 2013.

![Algorithm 3 Candidate Email Identification](image)

Algorithm 3 Candidate Email Identification

1. candidates ← List < String >
2. map ← HashMap < String, Integer >
3. for all trace in traces do
4. if map contains trace then
5. increment count for the trace
6. else
7. add trace to map with count 1
8. end if
9. sort the map based on count fields
10. if map has at least one entry then
11. max ← count for first entry in the map
12. for all email in map do
13. if email's count is smaller than max then
14. break
15. end if
16. add email to candidates
17. end for
18. end if
19. end for
20. return candidates

In Figure 12, one node represents one cluster. Red boxes represent the clusters found using obfuscated email addresses and blue boxes represent clusters found using plain-text email address. We connect one creator cluster node with one kit user cluster node, if there is at least one common phishing website between these two clusters. As kit users collect kits from the creators, normally, we expect one kit creator cluster to be connected to multiple kit user clusters. However, in our case study, we found 167 kit user clusters (including the single phish clusters), which are not connected with any kit creator clusters. A possible reason behind this is that in some cases, creators do not use any obfuscation technique to hide their email addresses. Another possible reason is that the creators are using some obfuscation techniques that are unknown to UAB’s phishing mining application and it fails to extract the obfuscated email addresses. For these phishing kits, we cannot find any related creator cluster. We also found 14 creator clusters that are not connected with any kit user clusters. For these cases, phishing kit creators may be launching phishing attacks by themselves. Hence in these cases, we cannot find any related kit user cluster associated with these creator clusters. Most importantly, from the relationship between kit user and creator clusters, it is clear that one group of creators can distribute their phishing kits to multiple groups of kit users. Hence, from Figure 12, we can identify the strongest creator group, which is behind the highest number of kit user clusters.

As we notice in Table VI, the average number of brands per cluster is always greater than one, and the number is higher for kit creator cluster compared to kit user cluster. This is obvious, since the creator can create phishing kits of different brands and can distribute among different group of phishing kit users. On the other hand, kit users can also collect phishing kits of different brands from different group of creators. From the relationship between kit user and creator clusters, we can identify how the kit users and creators are linked and over what brands. This relation shows us which group of creators is distributing kits for multiple brands, and which group of kit users are launching multiple brands phishing websites. Figure
VI. RELATED WORK

Researchers are working in phish clustering domain from various perspectives to mitigate phishing attacks. Some of the clustering approaches are used for defensive techniques, such as to prevent the phishing attack from reaching the intended general users. For this purpose, Saberi et al. classify the words in the email body to determine the legitimacy of the email [13]. Other email-based approaches use sender email, sender IP address, and non-matching URLs between the hyperlink and anchor tag as clustering features [14]. Gyawali et al. and Ma et al. proposed solutions to phishing identification by using features that can be derived from a URL [10], [11]. Web browsers can adopt these techniques to block phishing websites.

Moore et al. presented a scheme of identifying drop email addresses from an email provider’s email metadata [22]. They identified the drop email addresses from the URL field of the metadata. In the URL field of the drop email addresses, they found their email address that they entered to some phishing websites. The reason behind this is that when there is an email address inside a email body it is treated as an URL (mailto://victim email address). They also found a common pattern in the subject of drop emails. Using this pattern, they were able to find more drop email addresses. They argued that same subject of the drop emails points to one single kit, which boils down to one single phisher. Whereas, we found that a single phisher or a group of phisher can operate multiple kits. Moreover, we can distinguish between kit creator and kit users.

Clustering of phishing websites is also used for identifying the severity of phishing attacks. Chen et al. [7] proposed a technique to determine the severity of phishing attacks by applying textual classification and data mining techniques on phishing alerts from Millersmiles [23] and financial information provided by the phished organizations. They claimed to classify the risk level of phishing attacks based on the margin of loss from the phished organization and categorized text from the phishing alerts. As duration of phishing attack is important, Moore et al. [24] study the effective lifetime of phishers by examining data from the PhishTank data set [25]. Phishing message producers try to extend the usefulness of a phishing message by reusing the same message. Hence, Irani et al. proposed a clustering technique based on the phishing message [26]. They first classify the phishing messages into two groups: flash attacks and non-flash attacks. In flash attack, phishers send a large volume of phishing messages over a short period of time. While in non-flash attack, they send the same phishing message spread over a relatively longer period of time. Secondly, they classify phishing attacks using transitory features and pervasive features. Features that are present in a few attacks and have a relatively short life span (transitory) are
generally strong indicators of phishing, whereas features that are present in most of the attacks and have a long life span (pervasive) are generally weak selectors of phishing. Weaver et al. developed a clustering algorithm using the target, URL, and address where, target is the institution that the phish imitated, URL is the address of phishing website that were instructed to click on in the phishing mail, and address is the IP address of the site hosting the phishing website [27]. They applied capture-recapture technique on the clusters to report phishing population in order to estimate the extent of phishing present on the Internet.

Researchers who are trying to identify the criminal behind the phishing attack are interested in the aggregation of phishing websites. Most phishing websites are hosted on vulnerable web servers without the knowledge of the servers’ owners. Wardman et al. examined the common vulnerabilities, which allow these phishing sites to be created and suggested that similar attack signatures often indicates the same phisher [17]. The researchers applied the Longest Common Substring algorithm to known phishing URLs, and investigated the results of that string to identify common vulnerabilities, exploits, and attack tools, which may be prevalent among those who hack servers for phishing.

Britt et al. presented a clustering technique for identifying most prevalent phishing groups or individuals [28]. Their method was based upon the assumption that phishing websites, attacking a particular brand are often used many times by a particular group or individual. When the targeted brand changes, a new phishing website is not created from scratch, but rather incremental upgrades are made to the original phishing website. They used a SLINK-style clustering algorithm using local domain file commonality between websites as a distance metric. Their method produced clusters of phishing websites with the same brand.

Though there are quite number of works in phish clustering, none of the works considered drop email address as the clustering criterion. Some of the works identify predominant clusters, but none can identify the predominant phishers and creators.

VII. Conclusion and Future Work

In this paper, we have presented a clustering algorithm based on common drop email address found in the phishing kits. We applied our algorithm on three months of phishing data from UAB phishing data set and revealed several important findings – the most dominant clusters in terms of number of phishing websites, drop email addresses, time span, and strength factor, the most active kit creator and phishers, and relation between kit creator and kit user cluster. Our results provide some new insights into phishing paradigm. However, it also shows the need for further research to capture more phishing websites associated with drop email addresses, and to improve the technique of identifying obfuscated email address, since we notice that for large number of phishing websites there are no obfuscated drop email addresses in phishing database.

Our primary goal at this point is to refine the clustering algorithm so that we can run the clustering process for a long duration of phishing data. We will investigate the performance bottleneck for large data sets, and incorporate probabilistic clustering methods for large data sets. Moreover, as the number of phishing websites for which a drop email address is found is less than the number of confirmed phishing websites, we will further integrate other established phishing clustering schemes with our proposed scheme. Thus, we will be able to cluster phishing websites even if they do not have kits or drop email addresses.

ACKNOWLEDGMENTS

This research was supported by a Google Faculty Research Award, the Office of Naval Research Grant #N000141210217, the Department of Homeland Security Grant #FA8750-12-2-0254, and by the National Science Foundation under Grant #0937060 to the Computing Research Association for the CiFellows Project.

REFERENCES