infoblox

Threat Intel

RESEARCH REPORT

TOOLS OF THE TRADE
DISTILLING MALICIOUS CAMPAIGNS IN SPAM

A graph-based sieve for email spam

Author:
Dr. Renée Burton

Notice

Infoblox publications and research are made available solely for general information purposes. The information contained in this publication is provided on an “as is” basis. Infoblox accepts no liability for the use of this data. Any additional developments or research since the date of publication will not be reflected in this report.

Table of Contents

Introduction

This paper introduces a technique used by the Infoblox Cyber Intelligence Unit to identify malicious campaigns from email spam. The methods described here allow us to automatically process large volumes of data to focus our resources for manual analysis. In this sense, the techniques act as a sieve for email spam. Specifically, we use bipartite graphs constructed from email metadata and compute the set of connected components within them to identify likely individual campaigns. We studied the results using these graph algorithms over an 18 month period and have developed a set of best practices for their use. We have not seen a similar practice published elsewhere, so will show the results of our research and describe the methods we used.

Infoblox security products leverage block lists to protect our customers and their network users from Internet threats at the Domain Name System (DNS) level. We create original content for these products from several types of source data, using a range of algorithms and techniques. One of those data types is email spam. While spam identification techniques have become much more effective and widely integrated into end-user mail systems, this type of email remains prevalent as spammers continually adjust to evade the protective measures. As a result, customers continue to receive spam in their inbox.

Cybercriminals leverage spam as a high volume, low cost means to infect victims with malware. They are then able to steal or hold for ransom personal and proprietary information, gain access to and control of system processes, as well as spread to other connected individuals or networks. Their strategy is similar to Internet advertising, in which the small likelihood that a user will click on a displayed ad created a $124.6B yearly industry.[1] Malicious actors target individuals, major corporations, organizations and governments alike. As a result, locating spam-based malware campaigns as quickly and as accurately as possible to prevent further damage is a critical capability for the cybersecurity community in our effort to protect others.

This paper will begin with background on malicious spam campaigns and the challenges of using spam data as a source for block lists. Our approach to addressing these challenges leverages the field of graph theory. We first describe the necessary technique and terminology. Then we will walk through our results, and conclude with comments on other applications of graphs to Infoblox Threat Intel derived from spam collection.

1 PwC, Internet Advertising Revenue Report, May 2020, https://www.iab.com/wp-content/uploads/2020/05/FY19-IAB Internet-Ad-Revenue-Report_Final.pdf

Background

Malicious spam, often referred to as malspam, utilizes file attachments or embedded hyperlinks (URLs) to infect victims. The email recipient must either open the file or click on the URL, and often may need to enable macros or editing on their machine for the attack to continue. Threat actors use various types of lures, including spoofed documentation, promises of financial gain or threats of blackmail to trick victims into taking these steps. They gain access to the user's machine and often their private information by using lures that prey on people's hopes and fears, as well as inexperience with computer security. The consequences can be quite significant. One such example is the December 2019 Emotet attack that brought city services of Frankfurt, Germany to a halt.[2] Organized thieves also leverage crises like the Coronavirus pandemic[3] or Black Lives Matter protests[4] as a means to lure victims and steal their financial information.

An effective lure is only the first step in the attack chain, which may involve several stages and can occur quickly or over a longer period of time. In modern malspam, the attachments themselves generally serve to download further malware. This multi-stage process reduces the likelihood of stopping the malware infection through automated detection, and also allows the malware distributor to conduct checks on the victim's location or system configuration before proceeding. Malware delivered via email ultimately reaches out through the victim's network to its command and control (C&C) endpoint(s). The C&C domain names, IP addresses and URLs are referred to as indicators of compromise (IOCs). Recovering these IOCs for use in block lists is the ultimate goal of this research.

However, the massive volume of email spam, as well as the staged approach of the threat actors and their constant adaptation to avoid detection make it difficult to isolate IOCs. Traditional approaches leverage algorithms, both heuristic and machine learning, to identify suspicious code or content in websites. In some cases, automation is able to definitively determine whether a given attachment or URL is malicious, but more often it will lead to large quantities of generically suspicious emails requiring manual review. There are not enough human resources to manually evaluate all of these results.

As a result, we are left with large volumes of complex data and limited resources to locate the malicious behavior within it. By applying methods taken from the long-established scientific fields of graph theory and social network analysis, we created a workflow that allows us to automatically group together emails that are likely part of the same spam campaign. We use this as an initial filter and then apply more traditional methods of Infoblox Threat Intel to the results. This multi-step process allows us to focus our resources and harvest IOCs more efficiently. In the next two sections, we define terminology and detail our technique.

2 Kaspersky ICS-CERT, German cities under attack by Emotet botnet, 24 December 2019, https://ics-cert.kaspersky.com/news/2019/12/24/emotet-attacks-german-cities/
3 US Center for Disease Control, COVID-19-Related Phone Scams and Phishing Techniques, 3 April 2020, https://www.cdc.gov/media/phishing.html
4 E. Patterson, BLM Themed Malspam Delivers Trickbot Trojan, 1 July 2020, https://insights.infoblox.com/threat intelligence-reports/threat-intelligence--77

Terminology

A **graph** is a mathematical representation of connections between different items called nodes within a set. Two nodes are connected by an **edge** if they share some attribute or feature. Graphs abound in our everyday life, and we intuitively incorporate them into our decision processes. Some familiar examples include:

- Social media leverages connections between individuals, creating a network in which nodes are representations of people, and edges represent relationships such as friendship, readership, or common interests.
- Contact tracing in viral outbreaks creates a graph in which nodes are individuals and edges represent contact.

Graphs in one form or another have long been used in investigations. The oldest, and among the most famous is the work by John Snow[5] in identifying the source of London's cholera outbreak from 1853 to 1854. This study influenced not only epidemiology, but the fields of network analysis and graph theory. Evolutions of Snow's original results have been produced over 150 years later including work by Shiode[6] to further visualize the distribution of victims. The social sciences and epidemiology originally dominated the use cases for graphs, which were most often hand constructed and easily interpreted. The advent of computers and the capability to conduct large-scale processing opened the door for areas of mathematics and computer science to process and visualize extremely complex networks.

There are numerous types of graphs. The techniques described in this paper leverage undirected bipartite graphs. A graph is **undirected** when there is no inferred directional relationship between the nodes. Undirected graphs are most easily understood as the opposite of directed graphs, in which an edge between two nodes has some specific relationship. For example, a graph consisting of email addresses representing email sent between parties is directed if the edges are interpreted as node A sent email to node B, but node B may not have necessarily sent email to node A. Each edge in such a case represents a directional relationship.

If a graph is constructed with nodes that split into two distinct sets, and edges only exist between the sets, the graph is considered a **bipartite** graph. This is sometimes referred to as a **bigraph**. For example, a graph constructed from email in which the nodes are the set of sender addresses and the set of subject lines, and edges represent an email from the sender with that subject, is a bipartite graph. Edges in a graph can be assigned a weight to indicate frequency of a relationship or importance. In our example, the **weight** of an edge might be the number of emails from the sender with a given subject line.

5 The John Snow Archive and Research Companion, https://johnsnow.matrix.msu.edu/index.php
6 S. Shiode, Revisiting John Snow's Map: network-spatial demarcation of cholera area, February 2012, https://www.tandfonline.com/doi/full/10.1080/13658816.2011.577433

Two nodes are considered **connected** if there is an edge between them. A **connected component**, often shortened to component or **cluster**, is a subset of the graph in which all of the nodes are connected by edges. Within a connected component, any node can be reached from any other node by traversing edges. The **size** of a component here is defined as the sum of the weights of all of its edges.

Email metadata includes the headers and envelope associated with an email and its transmission across the Internet. This metadata contains structured and unstructured data related to the original email, along with a log of the steps taken for it to reach the destination. The actual communication within the email is considered the body. The headers and envelopes are themselves quite complex, and for the purposes of paper, we will restrict the discussion to commonly recognized fields, e.g., the subject, the sender's IP address, attached filenames, etc.[7, 8]

We define a malspam **campaign** to be a set of emails sent by a threat actor, through either the use of a spambot or a directly controlled infrastructure. We classify malspam campaigns as being limited in both time and content. The emails in a campaign may contain several topics, but share other features such as malicious attachments, or focus on a single theme, such as shipping notifications or current events with variations in other features.

Technique

To isolate spam campaigns, we create an undirected bipartite graph from email metadata. Each connected component within the resulting graph represents a set of emails that are likely all part of a single campaign. Treating each component as related allows us to focus our subsequent Infoblox Threat Intel processes onto representatives of each component, as well as prioritize our resources based on the size of campaign or some other feature of the graph.

These graphs can be computed over varying time intervals. We have found that a window of three to five days is very effective in identifying complete and accurate campaigns. We use longer timeframes to study the threat landscape, and shorter intervals to quickly isolate campaigns for the purpose of extracting threat indicators.

Additionally, there are a large number of combinations within email metadata to use for nodes within the graph. We found the optimal choice to be somewhat dependent on the exact nature of the email collection. For example, the use of subject lines and filenames works well in cases where the email contains file attachments.

Illustration of Results

We studied the effectiveness of these techniques over an 18 month period. To demonstrate the results, we will use a set of over 21,000 emails containing attachments from 18 to 24 December 2019. We constructed an undirected bipartite graph with nodes drawn from the

7 wikipedia.org, Simple Mail Transfer Protocol, https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol, https://tools.ietf.org/html/rfc5321

8 wikipedia.org, Internet Message Format, https://en.wikipedia.org/wiki/Email#Internet_Message_Format, https://tools.ietf.org/html/rfc5322

email subject lines and the attachment filenames. In this dataset, unique attachments are in direct correlation with filenames, meaning every filename represents a distinct file attached to the email. While not always the case, this is a dominant feature we have consistently observed in spam over time.

The initial graph has 866 components, many of which may contain a single email. To reduce noise, we remove components with a size less than five. This reduces the total number of nodes, thereby reducing the number of components to 101. As an immediate result, assuming the components capture campaigns well, we have reduced the number of items needing review by 99.5 percent, from the original 21,000 emails to a single representative email from each cluster. As shown in Figure 1 below, by coloring the nodes consistent with their type, we can also gain an overall understanding of the emails in this dataset. In particular, notice two large clusters dominated by the color green, indicating that they have a very large number of filenames and a smaller number of distinct subjects. In contrast, three large sets are loosely connected and are characterized by a large number of subject lines.

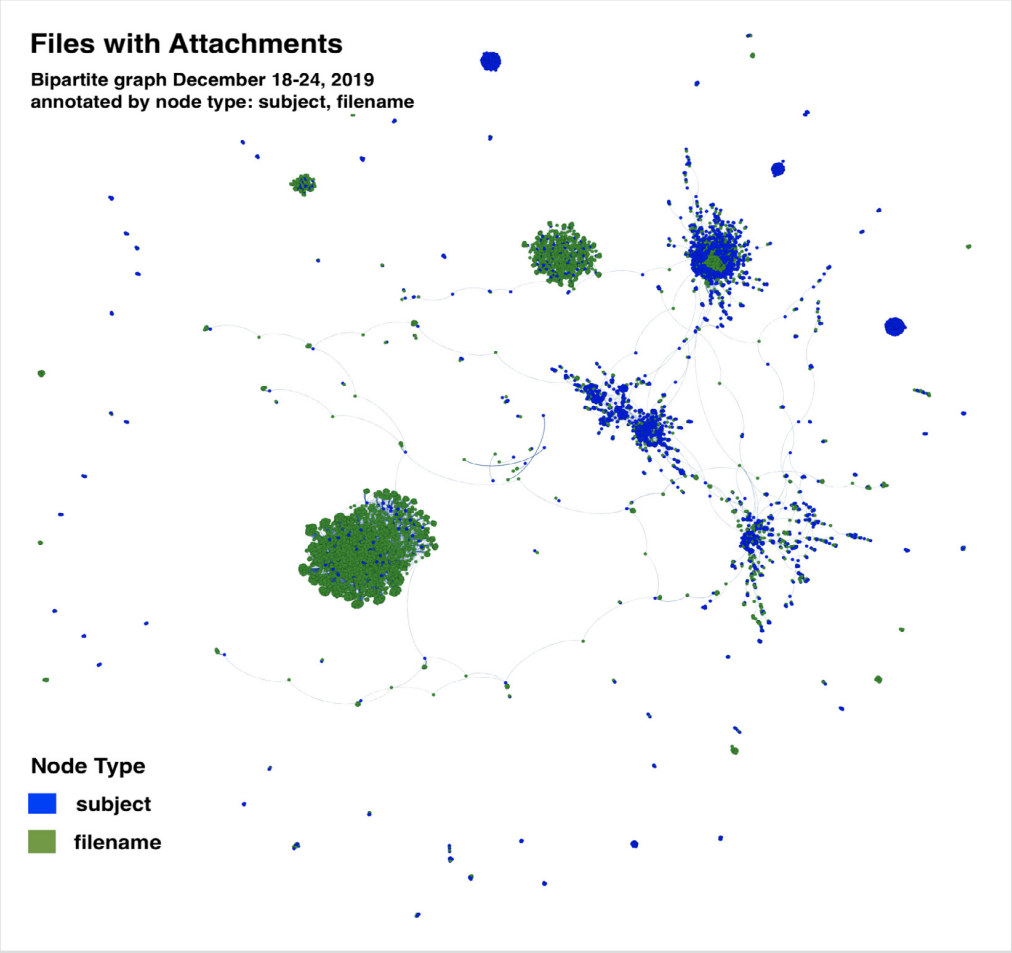

Figure 1. A bipartite graph generated from spam December 18-24, 2019

If we color each node in the graph instead by the associated connected component, we see that the vast majority of emails are found in a handful of components. This allows us to focus our analytic resources on campaigns with larger impacts. In particular, as shown in Figure 2 below, we find that the loosely connected emails above are all part of an Emotet campaign that lasted for much of the week.

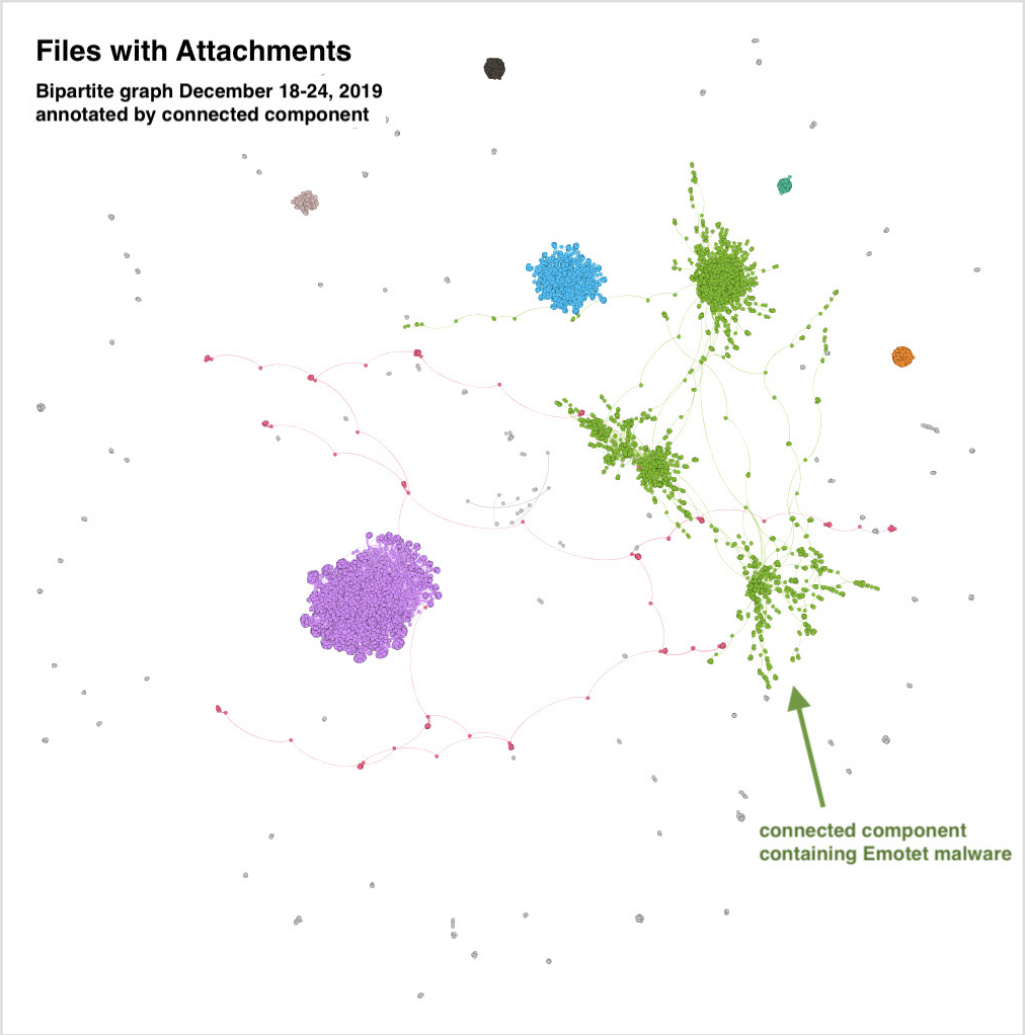

Figure 2. The same bipartite graph as in Figure 1 colored by connected components.

If we isolate our review to the large set of malicious emails sent by the threat actor behind Emotet, shown in green in the Figure 2, we can demonstrate a number of other features of our technique. First we compare the difference between the graph resulting from one, three and five days of data, as shown in Figures 3 through 5. This illustrates two advantages of increasing the time frame used; disparate components are drawn together over time, and the larger data set includes a more complete set of the actor's activity.

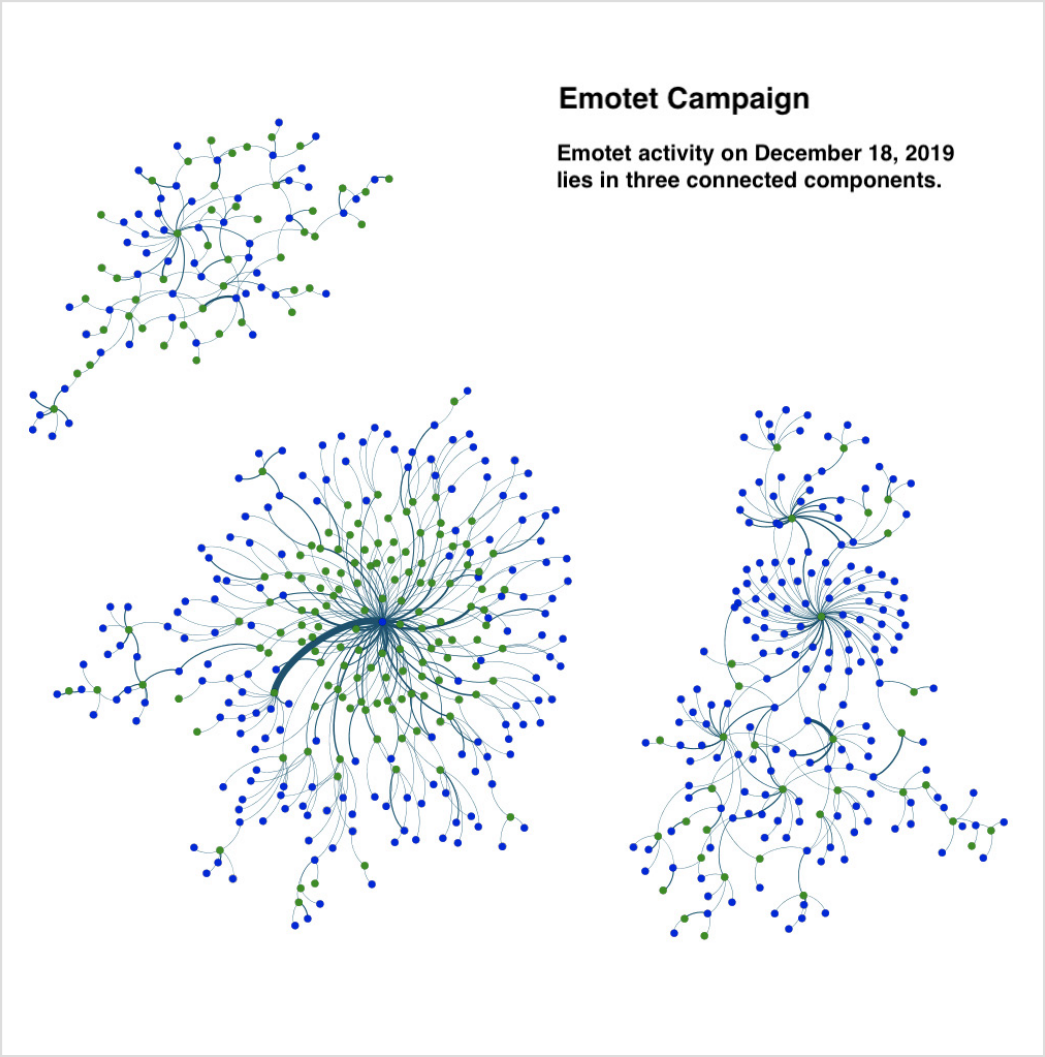

Figure 3. An Emotet campaign from December 18, 2019. This is a subgraph of the graph shown in Figure 1.

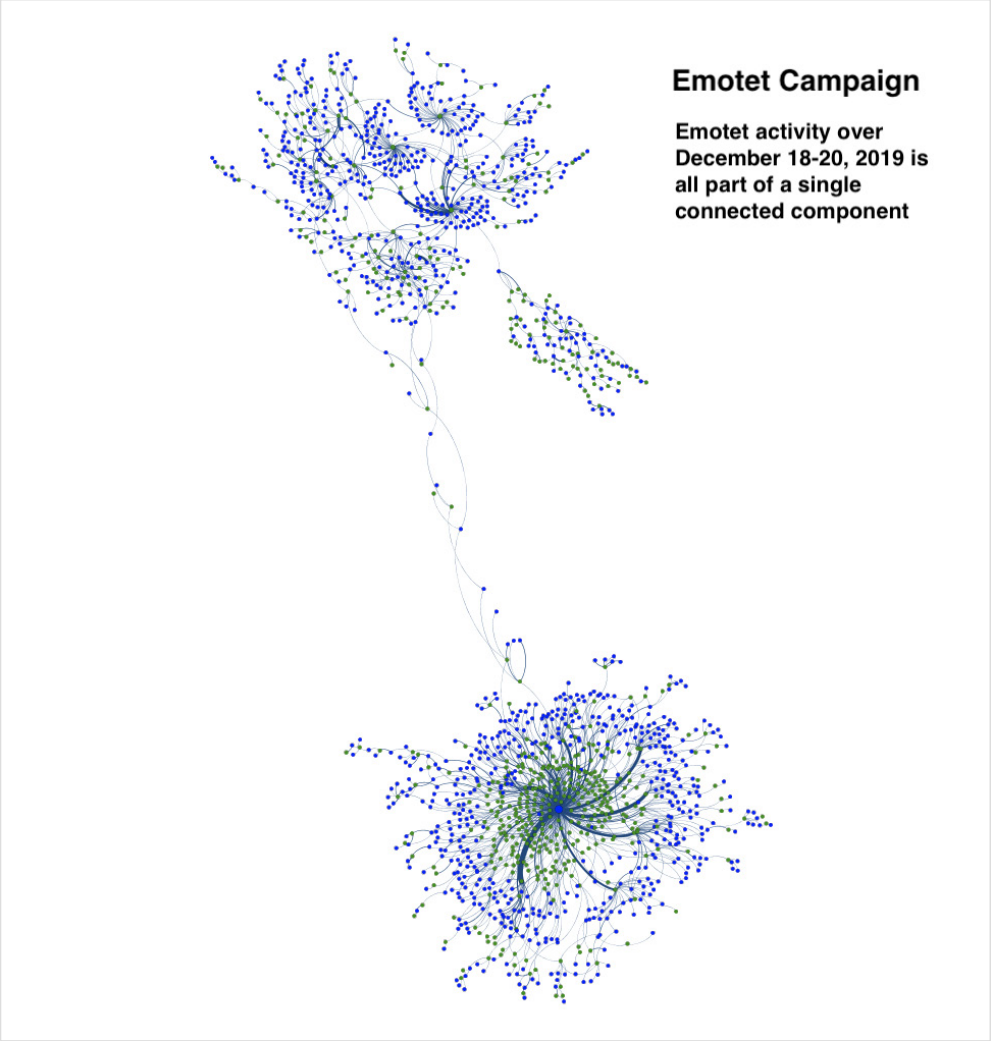

Figure 4. An Emotet campaign as observed over December 18-20, 2019. This is a subgraph of the graph shown in Figure 1.

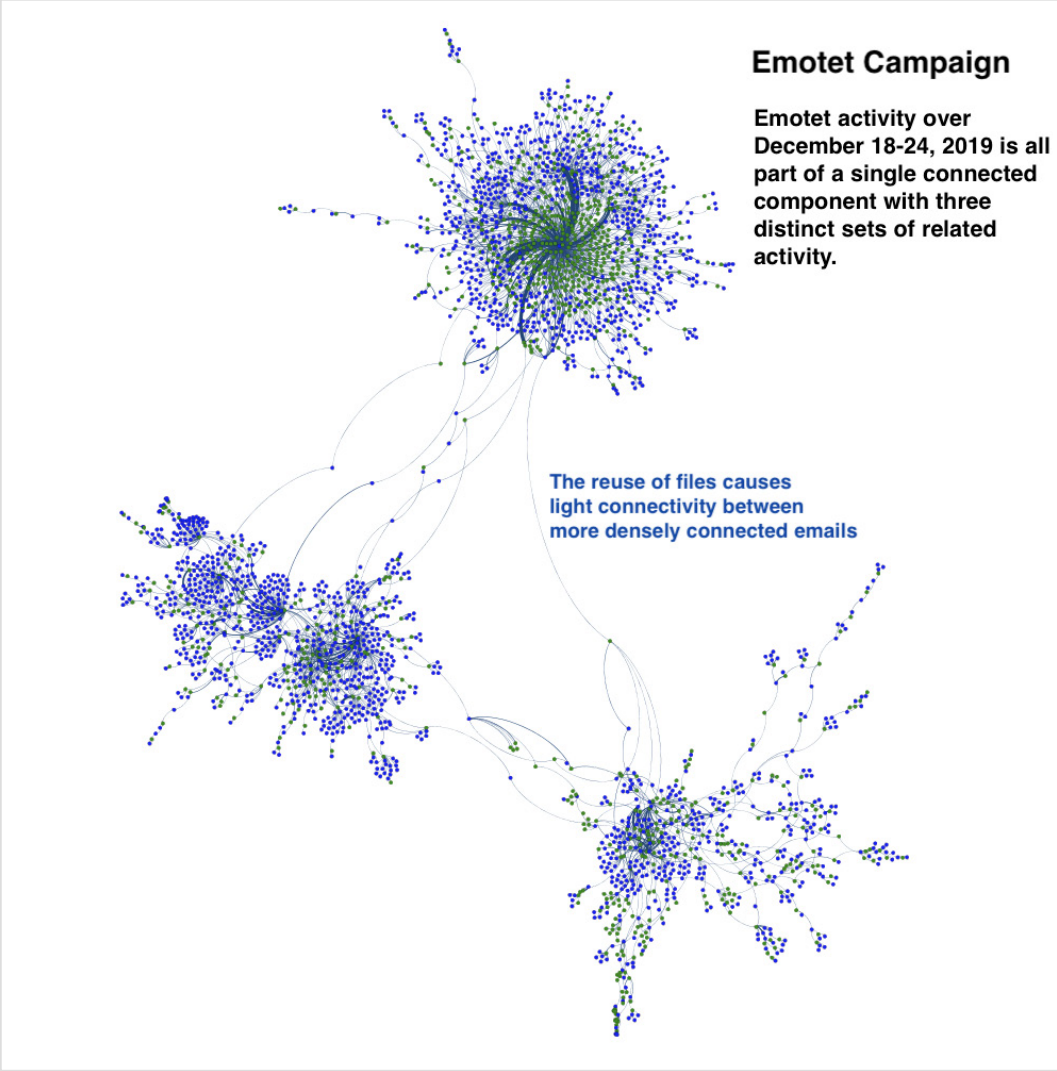

Figure 5. An Emotet campaign as observed during December 18-24, 2019. This is a subgraph of the graph in Figure 1.

Another advantage of the graph technique is that it correlates activity not fully visible in open source intelligence (OSINT) sources. In Figure 6, we can see a subset of the Emotet campaign and the relative proportions of inclusion in VirusTotal, a popular repository of file classifications by various anti-virus vendors. The use of the connected component as an initial filter allows us to associate many more files with Emotet's activity than through OSINT alone.

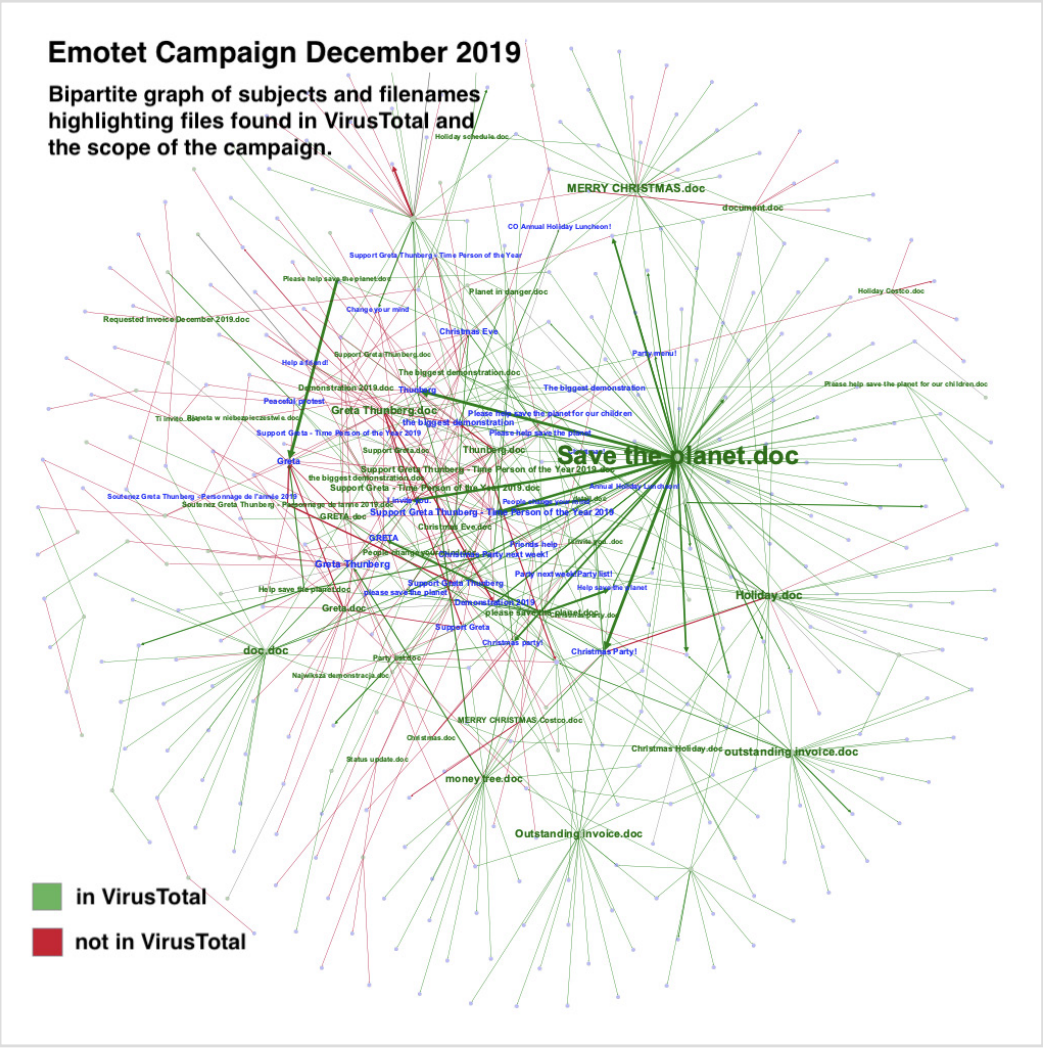

Figure 6. An Emotet campaign annotated by the presence of the file in the public repository VirusTotal.

Conclusion

Our primary use case for graph techniques is to easily identify malspam campaigns for more reliable, efficient indicator extraction. To this end, the graph must be constructed such that the components are, as much as possible:

- pure, or correct, campaigns, and
- complete campaigns.

In general terms, the former goal is easier than the latter. While we evaluated a wide variety of graph constructions for this purpose, we have found the use of bipartite graphs to be the most effective. In addition to isolating campaigns, we have used these graphs for other purposes, such as providing insight into the overall threat landscape. These techniques led to the discovery of the malicious spam actor, WordyThief,[9] who distributes malware that steals personal data from victims.

We recommend the following practices based on our results:

- Separate data into categories that can be analyzed independently. For example, these categories could be based on whether attachments exist, and of what type, or whether the emails contain embedded URLs.
- Within a category, perform statistical analysis on the primary fields identified by the subject matter experts as relevant to campaigns. In particular, two fields that are of one-to-one correspondence will not add value to your graphs, and only one should be used. Fields that contain the most diversity are more likely to be helpful in campaign isolation.
- Create graphs over multiple days to capture accidental transitions made by the threat actor and to visualize the full scope of malicious activity today.

9 Burton, Tymchenko, Sundvall, Hoang, Mozley, Josten; Wordy Thief: A Malicious Spammer, eCrime2020 conference proceedings, to appear November 2020

Corporate Headquarters
2390 Mission College Blvd, Ste. 501
Santa Clara, CA 95054

+1.408.986.4000
www.infoblox.com

Version: 20240614v1