The Benefits of a Functional Approach to Detecting and Mitigating a DDoS Attack

Robert McAndrew, Stephen Hayne, and Haonan Wang
Colorado State University
Fort Collins, Colorado, USA
email: robert.mcandrew@colostate.edu, stephen.hayne@colostate.edu, wanghn@stat.colostate.edu

***Abstract*—Distributed Denial of Service (DDoS) attacks have received significant global attention because they are increasing in frequency and severity. We analyze all flows surrounding the Network Time Protocol (NTP) amplification attack that occurred during January of 2014 at a large mountain-range university. We present an unsupervised machine learning data-driven approach that can detect and mitigate attacks in near real-time. Our method is based on thresholding, Functional Principal Component Analysis, and K-means clustering (with tuning parameters for flexibility), which dissects the dataset to reveal several categories of outliers. Using eigenfunction scores, clustering, and individual IP behavior summary statistics, we assign risk probabilities to the outliers, which enables creating dynamic firewall rules. We demonstrate the speed and capabilities of our technique in a forensic replay of the NTP attack. We show that we can detect and attenuate the DDoS within two minutes with significantly reduced volume throughout the six waves of the attack.**

Keywords-anomaly detection; clustering; DDoS; Functional Principal Component Analysis; network monitoring.

I. INTRODUCTION

There is an abundance of network security events, and one of the most impactful is the Distributed Denial of Service (DDoS) attack, in which attackers attempt to flood the target systems with huge amounts of traffic from many compromised systems leading to an interruption of the victim's services. Often, victims are high profile networks in companies, banks, or governments, and sometimes entire Internet Service Providers (ISPs) are targeted. Adversaries want to not only steal data (for later use or sale), but also disrupt operations of those targeted and impact their reputation. Hackers also increasingly use DDoS attacks as a smokescreen or distraction for more covert operations that allow them to carry out data breaches [1].

DDoS have been reported in the 1Tb/s range, driven by more than 150,000 compromised Internet of Things (IoT) devices, such as DVRs and security cameras [2]. Just a few months before that attack, the same botnet launched another in the range of 600Mbps - the volume trend is upward. Also, in Q4 of 2017, 67% of DDoS targets were blasted with more than one attack - an increase of 10% from Q3 [3]. In Q4, 32% of targets had between two and five assaults aimed at them, 6.5% that attracted between six and nine attack attempts, and a truly unfortunate 29% that were targeted over ten times (mean is 8.7 attack attempts per target over the course of the quarter). Perhaps a silver lining is that the attack duration has significantly decreased from an average of five days in 2016, to 1.3 hours at the end of 2017. However, direct costs to large organizations range from $50,000 to $100,000 per hour [4].

One variant of DDoS is the amplified reflection attack. There are several services that are vulnerable, and the one we will focus on here uses the Network Time Protocol (NTP). In this type of attack, adversaries send relatively small queries spoofing victim's Internet protocol (IP) address(es) to public servers (e.g., an NTP server), requesting a response - usually a large amount of data. As a result, this floods both the server's and the victim's network bandwidth. In 2014, 85% of all DDoS attacks larger than 100Gbps were using NTP amplification [5], and the bandwidth consumed peaked at 1% of all global Internet traffic (in late 2013 and early 2014) [6]. NTP had grown from .001% in early 2013; a dramatic three order of magnitude rise in both absolute and relative terms. This translated into organizational and financial impact [7] [8]; in fact, our own university suffered significantly. After peaking globally on February 11th (2014), NTP traffic declined back to .1% by May, still two orders of magnitude higher than at the start, as attacks continued on unmitigated systems.

In a Department of Homeland Security (DHS) funded project called "*NetBrane*" (see Figure 1), we model and characterize traffic both prior to and during DDoS attacks in order to quickly detect them and mitigate their impact. While hosted cloud-based security services offer some protection from DDoS, current solutions cannot benefit everyone. Many institutions, such as government, military, and financial organizations, need to tightly control their data, which is incompatible with cloud services. To bridge this gap, [9] is designed to be a defense service that takes advantage of the desirable properties of cloud technologies but allows customers to keep their data local. In this system, anomaly detection analytics using machine learning occurs on pre-attack network flows (inside the red box in Figure 1).

At our large university, we have installed optical taps to capture network flows at line rate (40gbps or more, top left of Figure 1) and push those flows into Hadoop Distributed File System (HDFS). Our analytics engine reads those flows in one-minute intervals, and searches for anomalies that should be investigated further. We use multi-core (parallel R packages) techniques.

In this paper, we study NTP traffic flows captured at our organization during the real 2014 main reflection attack. We conduct forensic re-analysis using our methodology to detect outliers in the flow data and apply the result to mitigate the effects of the actual DDoS in near real-time. Specifically, we detect unusual behaviors in two steps: (1) Functional Principal

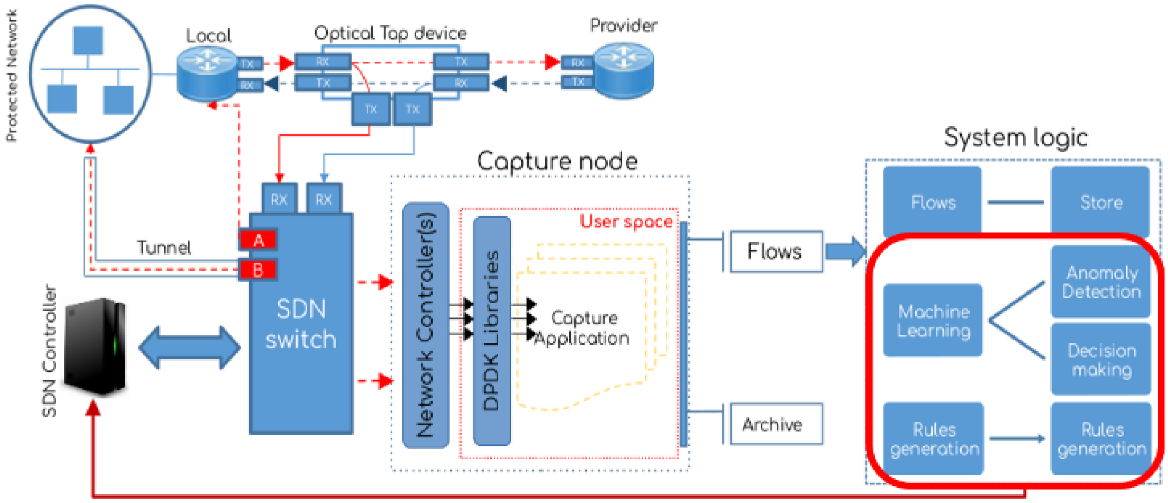

Figure 1. System Architecture

Component Analysis (FPCA) combined with (2) k-means clustering.

The remainder of this paper is structured as follows: Section 2 describes previous work related to Internet anomaly detection and Section 3 describes the dataset we use. In Section 4, the methodology of our technique is detailed, and Section 5 shows and discusses results of the analysis. Lastly, Section 6 discusses results and limitations, and Section 7 provides a conclusion.

II. Related Work

Anomaly detection methods can be classified into (1) signature based and (2) profile-based [10]. Signature-based methods use prior knowledge about characteristics of the anomaly of interest to identify suspects, and have several concerns, such as the need for labeled data, an external supervisor, and prior results from anomalies. Many machine learning classification techniques are supervised, meaning that they need to be trained on a set of labeled data prior to use. Examples of popular approaches are the Support Vector Machine, Bayesian Networks, Neural Networks, and Discriminant Analysis (surveyed in [9][10]). While these have been shown to perform well in certain situations, the reliance on labeled data can be a difficult hurdle to overcome. For the case of network traffic classification, ground truth knowledge may not be available. These supervised techniques can then only be applied when the true labels are approximated. Training on an incorrectly labeled dataset can greatly skew results [11].

In the case of a real-world DDoS attack, knowledge of which behaviors are malicious is not known; we do not have labels. Thus, supervised techniques cannot be applied. Profile-based methods create representative normal traffic behavior, and anomalies are detected by deviations from this profile. While there may be higher false alarm rates, profile-based methods are more promising due to their data-driven flexibility and they may also detect previously unknown anomalies [11]. Principal Component Analysis (PCA) is a widely used profile-based method which has been applied to detect traffic anomalies in DDoS data by decomposing network traffic into two components [12]. The anomalous subspace, which is noisier and contains the significant traffic spikes, is separated from the normal, which is dominated by predictable traffic. An individual observation is deemed an anomaly if its projection to the anomalous subspace is large. A two-stage approach was proposed, using (1) PCA to identify potential anomalies, and (2) a meta-heuristic to group them [13].

However, the use of PCA has been criticized due to issues pertaining to (i) false positive rates, (ii) traffic measurement aggregation, (iii) normal subspace pollution, and (iv) correct anomaly identification [14]. The third is important, as it highlights the need to choose which principal components represent normal behavior, and which ones represent the abnormal. It has been demonstrated that some traffic captures do not lend themselves to this partition/selection; that is, all principal components contain abnormal behaviors, and thus this approach is not usable.

Clustering is another example of a profile-based method. Clustering has been applied to all traffic, comparing the centers of known normal traffic clusters to the centers of actual traffic, to try and determine if the actual traffic is not normal [15]. Unfortunately, this approach has only been applied to Simple Network Management Protocol (SNMP) objects, not network flows, and requires known normal traffic data. Clustering techniques have been used to characterize DDoS attack traffic (k-means, CLARA, and Self Organizing Maps) [16]. K-means was found to be the most accurate for attack detection because attack traffic displays strong similarity as opposed to the heterogeneity of normal traffic. Note their attack cluster still mixed legitimate traffic in with malicious (between .4% and 2.04%). We believe this phenomenon can be eliminated by clustering only demonstrated outliers, not all traffic.

To avoid the concerns with PCA and clustering when applied separately, we will use FPCA (instead of PCA) and

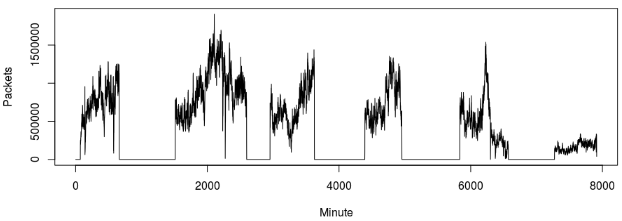

Figure 2. NTP DDoS Attack - All Waves

apply clustering to the resulting outliers [17] (that paper examined scanner behavior, where here we analyze a DDoS attack). We perform classification only using the data that is given as input, making this technique well-suited for dealing with an attack in real-time. We suggest this is more appropriate than using a supervised approach trained on data from a previous attack, as there are a wide variety of different perpetrators with attack methods, and what was previously learned may not apply.

III. Dataset Description

The raw data we consider in this paper is a collection of bi-directional flow records to and from our mountain-west university, relating to the NTP service. We focus on traffic between January 12 and January 25 of 2014, during the second half of which a true real-world reflection DDoS was carried out (starting in the early morning of January 18). This attack impacted the university in six waves (see Figure 2 for a plot of packet counts), with a wave defined by significantly decreased packet volume, or the monitoring system becoming unavailable. Discussion of our analytics will begin with the first wave because detection and mitigation are most crucial at the start of the attack but will generalize to all waves.

The flow records contain a plethora of useful information, such as timestamp, source and destination IP (SIP & DIP), source and destination port, packets, and bytes. We group information into one-minute bins, and the full dataset covers roughly twenty-thousand minutes. As this is a real-world dataset, we do not have ground truth knowledge of which SIPs are the victims (spoofed by attackers). However, we suggest that ground truth is not necessary as we know that a reflection DDoS indeed occurred, and we are only seeking ways to alleviate damage.

IV. Methodology

We conduct our analytics in near real-time by replaying the actual NTP attack with a sliding-window mechanism; to illustrate: the initial round of analytics is carried out on the data that appears only in the first thirty minutes of the dataset, the next round of analyses on the second to thirty-first minute of the dataset, and so on.

In each thirty-minute window, we construct a time series of contacts made for each SIP that appeared on the network, where a contact is defined to be a SIP sending at least one packet to a DIP. So, every external IP has a corresponding series with each value counting the number of DIPs they contacted in that minute. These series are used as input for Functional Principal Component Analysis (FPCA), which we use to identify outlier IPs - these are SIPs that interact with our network in an unusual way. These outliers are then clustered with the K-means algorithm to facilitate understanding of the outliers and streamline creation of firewall rules if an attack is detected. When not under attack, these outliers are displayed for operators to monitor or investigate further. Brief details of FPCA and K-means are included in Section IV.B, with full discussion found in [17].

We also monitor the time series of aggregated (SIP and DIP) packets and bytes in each thirty-minute window to volumetrically detect when the attack begins. In each window, a threshold is calculated, and when a new minute's data exceeds the pervious window's threshold, it identifies a potential start of a DDoS attack (next section).

If our volumetric threshold(s) are exceeded, we suspect that we may be under attack. At this point, the sliding-window becomes a growing window, fixed at the current time, and subsequent minutes are appended to the previous data. For example, if a significant volumetric change is detected while we are considering a window from minute 30 to 59, the next window we analyze will contain data from minute 30 to 60. This growing-window is used so we do not skew outlier detection as the attack proceeds; that is, if more behavior enters the network that is similar to that which caused the volumetric trigger, we do not want it to become representative of normal traffic.

When we are under attack, the outliers gathered and clustered by FPCA+K-means are remembered in what we refer to as a total recall strategy. We keep a running list of outliers that are detected and assign a threat level to each based on the individual IP's activity on the network (details in Section IV.B). From the threat levels, we construct a list of suspected attackers, as well as a list of those that are believed to have acceptable behaviors. It may seem counter-intuitive that acceptable traffic can be flagged as unusual, and more discussion on this is given in Section V. These two groups can then be blocked from and allowed into the network, respectively, to mitigate the attack and yet allow some known good systems to continue access.

As a final note, when the analytics have detected an attack, we implement a two-pass procedure where we repeat FPCA with outlier collection on the subset of the data that were classified as non-outliers from the first pass. The outliers from the first and second passes are added to the running list. This two-pass is carried out only during attack mitigation, and later we will compare the effects of one and two passes to demonstrate the marginal gain from each round of outlier collection. (Discussed in Section V).

All data management and analytics are carried out in version 3.4.4 of the R programming language. The analytics in each sliding-window iteration takes approximately four seconds, while the attack analytics (growing-window) take no more than ten seconds (on a 10Gb set of flows) using eight cores.

A. Volumetric Attack Detection

As a first warning for a DDoS event, we seek to identify a drastic increase in packets or bytes sent and received by the network in any given minute. We define this drastic increase to be an instance when a new minute's data exceeds a threshold from the previous thirty-minute window. Specifically, we calculate separate thresholds for packets and bytes in each window and compare the new minute's aggregated packet and byte counts to these thresholds. The threshold is defined by (1),

$$Threshold = \max_{t \in H} X_t + cv \cdot SE\left[\max_{t \in H} X_t\right]. \qquad (1)$$

In (1), X_t for $t \in H$ is the time series of packets or bytes in the given window of history. $SE\left[\max_{t \in H} X_t\right]$ is the standard error of the maximum packet or byte count from a LOESS fit of the packet/byte time series in the window of history. Lastly, cv is a critical value determined from investigation of long-term (months) packet and byte distributions. This takes the largest value in the given window of history and sets a threshold greater than it by adding a scaled measure of this maximum's variability. That is, we start with the largest packet or byte count in local history which is considered acceptable because it previously did not indicate a potential DDoS, then raise this to calculate the threshold. This is motivated by the idea that we may see normal network activity that is larger than the previously accepted amount, but we only expect a DDoS if new packet or byte counts exceed what we expect from historical variability of our data.

If a new minute of data is collected and it does not exceed the previous window's threshold, the time series of flows for the new thirty minutes are stored, and the threshold is recalculated. This creates a dynamic threshold for volumetric detection that takes usual network activity into account.

If a new minute of data is collected and either our packet or byte threshold is exceeded, we consider an attack to be starting and we initiate the growing window and two-pass FPCA outlier collection.

B. Outlier Detection and Risk Assessment

FPCA takes as input a collection of series that can be treated as realizations of a function over time, and then models these series as a mean curve plus a linear combination of eigenfunctions. These eigenfunctions are orthogonal curves created by finding the largest dimensions of variability in the data. That is, the first eigenfunction can be thought of as the direction of highest variance, the second captures the next most variance, and so on. When the original series are projected onto the eigenfunctions it produces scores, which are the locations of the observations on each new dimension. These scores are used to identify data points as outliers. For each eigenfunction, we calculate the bounds $\bar{x} \pm ks$, where $\bar{x}$ is the average score, s is the standard deviation of the scores, and $k > 0$ is a constant. If a score is outside of the bounds on any eigenfunction, it is flagged as an outlier. As we use the time series of contact counts for input, these outliers are the IP addresses that are interacting with the network in an unusual way. With the feature of descending variance in the eigenfunctions, SIPs that are outliers based on our definition are also extreme in the sense of the original dataset. For our analytics, we use the Principal Analysis by Conditional Expectation (PACE) implementation of FPCA [18].

The number of eigenfunctions to use in the FPCA model is a parameter that must be selected so we use the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) [19]. Should the results of these not match, factors specific to the situation should be considered to determine which is better suited [20]. As this analysis is carried out in each iteration of the sliding window, disagreement between AIC and BIC at some point is likely. For the purposes of detecting unusual behavior in network traffic, the true model for which can be highly complex, we focus on AIC. It can also be the case that AIC and BIC are not applicable, depending on the amount of data available and its sparseness. In these situations, the number of eigenfunctions is chosen using a Fraction-of-Variance-Explained (FVE) cutoff. That is, we select the first n eigenfunctions that capture a certain percent of the variance from the original dataset. This method is always applicable, provided the FPCA model can be used successfully.

Following the gathering of outlier SIPs on the network in the available window, they are clustered using the K-means algorithm of [21]. The number of clusters is also chosen using a FVE cutoff commonly referred to as the elbow method. The cluster amount chosen is such that adding one additional cluster will not significantly increase how much variance of the original dataset is explained by our clustering; i.e., the point of diminishing marginal returns. The outlier SIPs are clustered based on their proportion of successful contacts, where a success is defined to be at least one packet sent back to the SIP by the DIP being contacted. With this completed, the result is a set of SIPs that are interacting with the network in an unusual way, stratified by their success. This facilitates easier understanding of the traffic that is detected.

After the clustering, we assign a threat level (TL) to the outlier IPs. This is calculated using (2).

$$TL = \alpha_1 v + \alpha_2 d + \alpha_3 (1 - c). \qquad (2)$$

In (2), the α_j's are constants that satisfy $0 < \alpha_j < 1$ and $\sum \alpha_j = 1$. Further, v is the proportion of volume sent and received by the given IP relative to that of the entire window, d is the number of destinations contacted by the IP divided by the total number contacted by the SIPs in the given sliding window iteration, and c is the fraction of minutes that the SIP reached out to at least one destination. For purposes of the analytics in this paper, we use $\alpha_j = 1/3$, but other choices can be made based on context-specific factors. With these definitions, the threat level TL exists between 0 and 1 and represents the maliciousness of the outlier - a threat level closer to 1 indicates greater likelihood the IP is malicious. The quantities used in this calculation are chosen based on analysis of NTP behaviors prior to and during the attack. It is likely that different factors must be considered when assigning a threat level to behaviors on other services.

Since we only group the outlier SIPs based on one numerical summary of their behavior, there can be instances

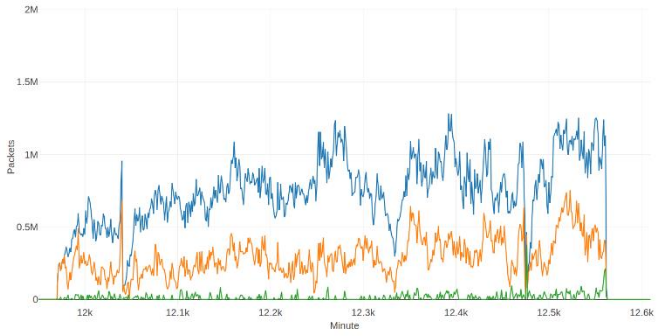

Figure 3. NTP reflection DDoS wave 1: actual packets (blue), one-pass reduced packets (orange), two-pass reduced packets (green)

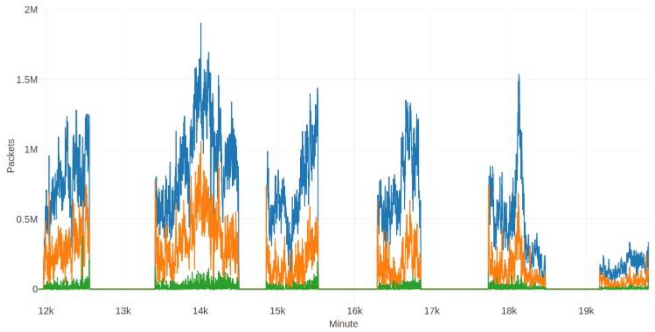

Figure 4. NTP reflection DDoS: actual packets (blue), one-pass reduced packets (orange), two-pass reduced packets (green)

where behaviors are mixed within clusters. The TL's alleviate this issue, as we can search for non-threatening SIPs in lower-centered clusters or threatening ones in the high.

V. Results

We first present results from attack detection, and then turn to mitigation. We begin our sliding-window analysis by working with the section of the dataset prior to the attack. This is roughly a week's worth of data, during which our detection mechanism does not trigger. So, in each iteration of the window, we carry out only one pass of outlier detection (FPCA) and the outliers are clustered with K-means. This results in 95 distinct outliers out of approximately 6 thousand SIPs across the entire pre-attack period. When assigning threat levels (TL's), the vast majority of these non-attack TL's are close to 0, indicating that the behavior is non-threatening. In fact, the outliers we identify here have well-known and acceptable NTP behaviors - they are consistently checking the time with the network's published NTP servers. We think of such behaviors as active peers because they are working in a way we expect for this service [22]. These are identified as outliers because they are contacting the network in a way that is unusual with respect to the rest of the dataset considered; that is, reaching out to one DIP uniformly over time.

In each iteration of our thirty-minute window prior to the attack, our threshold is calculated as in (1), with cv calculated from long-term historical distributions of maximum packet and byte counts. Specifically, we recreated our sliding window procedure on approximately 5 weeks of data directly prior to the dataset described in Section III, saving the maximum packet and byte counts in each window. We then find the 99th percentile for both distributions and standardize them to calculate cv. For example, we find the 99th percentile of our maximum packet counts, then subtract off the mean maximum packet count and divide this by the standard deviation of the distribution. The same is done for byte counts. In the case of packets, cv is 19.94, and in the case of bytes, it is 21.2.

As the sliding-window marches forward, we eventually reach the minute at which the attack begins (12000 minutes into the dataset). In the iteration that captures the start of the event, it is as if the attack has been going on for one minute, and we are observing that first minute along with the previous thirty (of pre-attack traffic). With a new minute of data, we compare the packet and byte counts to previous thresholds and find that both are exceeded. Our attack flag is triggered, and we begin to grow our window of history, which now contains the 30 minutes prior to the attack, as well as its first minute. As an approximation of real-world monitoring, our method detects the DDoS attack during its second minute of activity.

Once the attack has been detected, we begin applying our two-pass procedure, using the 3-sd from the mean cutoff to define outliers in each pass ($\bar{x} \pm 3s$ on each eigenfunction). With outliers identified, TL's are assigned and stored before adding the new minute's worth of data to our now growing-window (because we are under attack). In this new minute, we simulate blocking the high-threat outlier IPs activity from the network; that is, the data generated by all malicious outlier SIPs gathered in previous window iterations are removed from the new minute's information. This mimics the creation of firewall rules that would block IP addresses from the network and is the proposed mitigation strategy for DDoS - we remove those outliers which are found to be the malicious actors in the attack. Figure 3 visualizes the packets counts and mitigation during each pass, in order to demonstrate the reduction gain from each round of outlier collection. At the end of the first wave, we have identified about 100 unique SIPs as outliers. The one-pass reduced volume is approximately 60% less than the actual first wave, while the two-pass reduced volume attains a reduction of 95%. This is a significant mitigation result, making the attack look much more like pre-attack traffic than a DDoS.

Later waves of the attack are handled in the same manner, and Figure 4 visualizes the reduced packet volumes. Approximately 3000 SIPs are identified as outliers throughout all six waves. The volume reduction achieved in the first wave extends to the entire attack: 95% of the overall volume is masked by removing the traffic from outliers (detected by two passes of FPCA) that are determined to be threatening.

While we block the traffic from the threatening outliers, we propose allowing activity from the active peers through to the network. The series of packet volumes with and without the non-threatening activity are virtually identical, with the active peer outlier traffic representing only 3% of the overall data. This gives high mitigation levels of the attack while allowing the known active NTP actors (from clustering) to continue operating. We acknowledge that malicious actors could possibly take advantage of this but will address the issue in future work. For any of the window iterations during the attack, no more than six clusters are used at any point (as set by the elbow method). This stratification consistently collects what we consider as active peer behavior in the higher-centered clusters (those with the largest portion of success), facilitating the creation of firewall rules.

We now turn to alternatives in parameter choices and investigate their effects on mitigation. The standard deviation (sd) threshold can be altered. $k = 3$ was previously used, and we also considered $k = 1$ and $k = 2$. On one-pass of outlier detection, the 2-sd cutoff removes 80% of the traffic while the 1-sd achieves 90% reduction. A second-pass using these cutoffs achieves close to the 95% reduced volume found using the 3-sd threshold. This indicates that we do not gain new outliers in this attack by varying standard deviations.

Another option for mitigation would be to mask the subnets that contain the outliers found by our method. This option would reduce the number of firewall rules that need to be created to block the outliers; instead of creating a rule for each individual IP, a rule for the /24, /16, or /8 subnets could be created - we tested this. For example, if the address 169.229.70.49 is found as an outlier and we are masking traffic at the /16 level, we omit IPs with addresses 169.229.X.X from further iterations of our sliding window. Figure 5 shows the mitigation results on the first wave using one-pass (3sd) with varying levels of subnet masking. Observe that blocking the individual outlier IPs, the /24, and /16 subnets achieve similar reductions in traffic, while the /8 subnet mask diminishes packets by almost 95%. Note this uses one-pass of outlier detection, indicating that this blocking results in mitigation like our complete two-pass procedure without the subnet rules, however we suggest that blocking the subnets can lead to blocking legitimate traffic too.

Avoiding blocking of legitimate traffic is the motivation for the two-pass procedure. We know that a DDoS is happening but have no ground truth about the attackers – thus, we must at least consider the notion of false-positives. Blocking /8 will certainly block more legitimate traffic than blocking outliers from a second pass of FPCA. At each application of FPCA when scores are calculated, we can estimate the shape of the distribution of scores on each eigenfunction using kernel density estimation [23]. We carry this out on the scores from the second pass of FPCA, when the outliers detected in the first pass are removed. Specifically, we test the estimated distributions for normality using the Shapiro-Wilk test [24]. If we find evidence for the scores being approximately normal, it indicates that the IPs detected in the first pass are appropriate outliers. We can extend this idea to an n-Pass procedure, in which we stop the repeat applications of FPCA when normality of the scores is reached, or we can no longer apply our method. The most common reason is that eventually no outliers are found by FPCA.

We recall that we employ a Total Recall strategy, in which our threatening outliers are remembered from previous window iterations and masked from future incoming data. As this list grows over time, there are less outliers to be found and less passes of FPCA are needed. The second reason the procedure stops is because of no data, in which this is the case or the data available is few and sparse. Lastly, normality is reached only a small portion of the time. The average number of passes throughout all waves of the attack is 1.88, with only 1 window iteration needing 11 passes. This occurs near the beginning of the first wave, before a large list of outliers is built up. With this, and the packet reduction achieved by our methodology, we believe two passes is appropriate for outlier detection and attack mitigation.

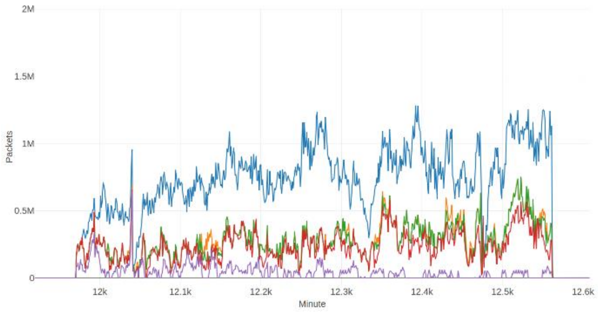

Figure 5. NTP reflection DDoS wave 1: actual packets (blue), one-pass reduced packets (orange), one-pass /24 reduced packets (green), one-pass /16 reduced packets (red), one-pass /8 reduced packets (purple)

VI. Discussion and Limitations

The notion of false-positives arises whenever anomaly detection is discussed. Without ground truth of the attackers in the dataset we analyzed, there is no way to accurately measure the false positive rate of the results in Section V. Our assignment of a threat level attempts to alleviate this issue, as we allow the non-threatening outliers to remain in the network's traffic. Even with this, there may be a few SIPs blocked that are not malicious. We suggest that this is not a major concern when truly under attack, as the security of the network is paramount and only outliers are being blocked.

The formula for calculating threat levels is created based on observing the data during and prior to the attack. Data from different services may require a different calculation of the TL, and this is true of the NTP-port as well - as this attack and more are studied further, the TL computation will be improved. Also, the methodology will benefit if selection of the α_j parameters is made dynamic and data-driven. For example, if network history or other contextual information can help select the weights for factors being considered, threat level assignment is expected be more accurate.

A major part of our analysis relied on the sliding-window approximation of real-time streaming data. We used a fixed thirty-minute window, because it is a near worst-case scenario, in that it is the smallest window of time-series data on which FPCA can still be applied. A larger history could be kept, and different types of behaviors may become apparent. This requires more data in RAM, especially during an attack, but we believe that the attack detection and mitigation would occur in the same way as presented in Section V.

VII. Conclusion

We have demonstrated an approach to detecting and mitigating an actual DDoS attack that occurred in early 2014. Dynamic volumetric thresholding is shown to detect the attack, and the FPCA+K-means approach mitigates the attack volume significantly (by >95%). These unsupervised approaches are best suited for detection and mitigation of unknown attacks. We have proposed multiple options for reducing the packet volumes of the attack, including the alteration of tuning parameters and masking subnets. Assignment of threat levels to the outliers allows for better understanding of the SIPs identified.

ACKNOWLEDGMENT

This material is based on research sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency (HSARPA), Cyber Security Division (DHS S&T/HSARPA CSD) BAA HSHQDC-14-R-B0005, and the Government of United Kingdom of Great Britain and Northern Ireland via contract number D15PC00205. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Department of Homeland Security, the U.S. Government, or the Government of United Kingdom of Great Britain and Northern Ireland. The research of Haonan Wang was partially supported by NSF grants DMS-1521746 and DMS-1737795.

REFERENCES

[1] S. Mansfield-Devine, "The Growth and Evolution of DDoS," *Network Security*, pp. 13 - 20, 2015.

[2] B. Krebs, "KrebsOnSecurity Hit With Record DDoS," 2016. [Online]. Available: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/.

[3] I. Incapsula, "Global DDoS Threat Landscape Q4 2017," 2017. [Online]. Available: https://www.incapsula.com/ddos-report/ddos-report-q4-2017.html.

[4] T. Matthews, "Incapsula Survey: What DDoS Attacks Really Cost Businesses," 2014. [Online]. Available: https://lp.incapsula.com/rs/incapsulainc/images/eBook%20-%20DDoS%20Impact%20Survey.pdf. [Accessed July 2019].

[5] M. Mimoso, "Volume of NTP Amplification Attacks Getting Louder," 2014. [Online]. Available: http://threatpost.com/volume-of-ntp-amplification-attacks-getting-louder/105763.

[6] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey and M. Karir, "Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks," in *Internet Measurement*, Vancouver, BC, Canada, 2014.

[7] K. Arora, K. Kumar and M. Sachdeva, "Impact analysis of recent DDoS attacks," *Intntl. Journal on Computer Science and Engineering*, pp. 877-883, 2011.

[8] J. Armin, B. Thompson and P. Kijewski, "Cybercrime Economic Costs: No Measure No Solution," in *Combatting Cybercrime and Cyberterrorism*, Springer, 2016, pp. 135-155.

[9] "NetBrane, Funded Project, Department of Homeland Security Award D15PC00205," 2015-2019.

[10] M. Ahmed, A. N. Mahmood and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, pp. 19-31, 2016.

[11] M. Soysal and E. G. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Performance Evaluation*, pp. 451-467, 2010.

[12] A. Lakhina, M. Crovella and C. Diot, "Diagnosing Network-Wide Traffic Anomalies," in *Computer Communication Review*, 2004.

[13] G. Fernandes, L. F. Carvalho, J. J. Rodrigues and M. L. Proenca, "Network anomaly detection using IP flows with principal component analysis and ant colony optimization," *Journal of Network and Computer Applications*, pp. 1-11, 2016.

[14] H. Ringberg, A. Soule, J. Rexford and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *ACM SIGMETRICS Performance Evaluation Review*, pp. 109-120, 2007.

[15] W. Cerroni, G. Monti, G. Moro and M. Ramilli, "Network attack detection based on peer-to-peer clustering of SNMP data," in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2009.

[16] B. Hammi, M. C. Rahal and R. Khatoun, "Clustering methods comparison: Application to source based detection of botclouds," in *Security of Smart Cities, Industrial Control System and Communications, 2016 Intntl. Conf. on*, 2016.

[17] R. McAndrew, M. Gharaibeh, H. Wang, S. Hayne and C. Papadopoulos, "A Functional Approach to Scanner Detection," in *Proceedings of the Asian Internet Engineering Conference*, Bangkok, Thailand, 2017.

[18] H.-G. Muller and J.-L. Wang, 2015. [Online]. Available: www.stat.ucdavis.edu/PACE/.

[19] S. I. Vrieze, "Model selection and psychological theory: a discussion of the differences between the AIC and the BIC," *Psychological methods*, p. 228, 2012.

[20] Y. Li, N. Wang and R. J. Carroll, "Selecting the number of principal components in functional data," *Journal of the American Statistical Association*, pp. 1284-1294, 2013.

[21] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C*, pp. 100-108, 1979.

[22] T. N. Distribution, "Association Management," 2014. [Online]. Available: http://doc.ntp.org/4.1.0/assoc.htm.

[23] J. S. Simonoff, Smoothing methods in statistics, 2012: Springer Science & Business Media.

[24] J. Royston, "Algorithm AS 181: the W test for normality," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, pp. 176-180, 1982.

[25] K. Labs, 2018. [Online]. Available: https://usa.kaspersky.com/.

[26] A. Singh, N. Thakur and A. Sharma, "A review of supervised machine learning algorithms," in *Computing for Sustainable Global Development (INDIACom)*, 2016.