FLOW BASED BOTNET DETECTION THROUGH SEMI-SUPERVISED ACTIVE LEARNING

Zhicong Qiu, David J. Miller and George Kesidis

School of Electrical Engineering and Computer Science,
The Pennsylvania State University,
University Park, PA 16802

ABSTRACT

In a variety of Network-based Intrusion Detection System (NIDS) applications, one desires to detect groups of unknown attack (e.g., botnet) packet-flows, with a group potentially manifesting its atypicality (relative to a known reference "normal"/null model) on a low-dimensional subset of the full measured set of features used by the IDS. What makes this anomaly detection problem quite challenging is that it is *a priori* unknown which (possibly sparse) subset of features jointly characterizes a particular application, especially one that has not been seen before, which thus represents an unknown behavioral class (zero-day threat). Moreover, nowadays botnets have become evasive, evolving their behavior to avoid signature-based IDSes. In this work, we apply a novel active learning (AL) framework for botnet detection, facilitating detection of unknown botnets (assuming no ground truth examples of same). We propose a new anomaly-based feature set that captures the informative features and exploits the sequence of packet directions in a given flow. Experiments on real world network traffic data, including several common Zeus botnet instances, demonstrate the advantage of our proposed features and AL system.

***Index Terms*—** active learning, anomaly detection, botnet, maximum entropy, p-value, semisupervised learning

1. INTRODUCTION

Detecting botnet communication presents a major challenge for current IDSes. Most coordinated bot malware is used to carry out malicious activities such as DDoS, spamming, and phishing, with huge cost to the victims. One recent survey [1] claimed that around 16% of host computers connected to the Internet today are compromised, becoming either active or passive bots, waiting to follow the bot master's commands. One of the most difficult challenges associated with detecting botnet communication is that both bot masters and slaves constantly modify their behavior to evade popular signature-based IDSes. For example, in order to avoid deep packet inspection based IDSes, botnet applications use secure transmission protocol (STP) to encrypt their command and control (C&C) messages. Recent botnets also use fast flux to randomize both their port numbers and/or domain names, thus avoiding anomaly detectors that are based on usage of certain port numbers or domain names, such as [2]. To avoid timing based signatures, botnets try to randomly delay their transmissions or make their traffic round-trip-times (RTTs) similar to perceived "normal" sessions subject to very low implementation complexity. On the other hand, the ever-evolving internetwork context requires an adaptive IDS to quickly respond to the ever changing states of distributed botnets, which are sometimes driven by context information not immediately available to IDSes.

In this work, we propose an AL framework to detect unknown botnet behavior, using only bidirectional packet sizes of a given flow to derive application-discriminating features. In our AL approach, the network administrator is leveraged as an "oracle" to inform the IDS which AL-forwarded network flows are suspicious (botnets) and which are known normal (e.g., web). Oracle decisions may be based on payload patterns, similarity to previously detected botnet flows, information from honeynets, etc. We propose a novel, anomaly-based, time-independent derived feature set that captures anomalies both in the packet direction sequence and in the sequence of packet sizes in a given flow. We improve upon a recent novel AL approach to learn the (potentially sparse) *informative* feature subset, starting from no ground truth about the botnet traffic. We compare our approach with the baseline flow-based feature representation from [3] and [4] as well as with the feature representation and approach in [5], and show our approach gives the best performance.

2. METHODOLOGY

2.1. Feature Space Representation

To capture the intrinsic behavior of botnet traffic, we note that in the C&C phase (communication between Bot masters and slaves for coordination or attacks), most botnet traffic involves master(s) periodically giving command (control) messages, with the slaves executing the given commands. Normal/background web traffic, on the other hand, tends to involve server-to-client communication. In the attack phase,

This research is supported by a Cisco Systems URP gift.

botnet malware carries out malicious activity, periodically sending out beacon signals to the bot master via the previously established C&C channel, most of the time without packet transmissions from the bot master to bot slaves.

Hence, we seek to preserve the bidirectional packet size sequence information as feature representation for different traffic flows. Such a feature representation was previously considered in [6] and [7]. The authors used the first N (we set $N = 10$ in our experiments) packets after the three-way hand shake of each TCP flow. Then a feature vector of dimension $D = 2N$ was defined, specified by the sizes and directionalities of these N packets. Traffic is assumed to be alternating between client-to-server (CS) and server-to-client (SC). A zero packet size is thus inserted between two consecutive packets in the same direction to indicate an absence of a packet in the other direction. Here, we improve on the representations used in [6] and [7].

2.2. Anomaly Based Derived Features

As we discussed previously, both the presence of packets in given directions and the sizes of packets should be informative in identifying botnet traffic. We accordingly define a set of anomalous scores to quantify such. Considering the previously defined D-dimensional feature vector $\underline{x} = (x_1, x_2, \ldots, x_D)^T$, we use $\mathcal{I}(\underline{x}) = (I(x_1), \ldots, I(x_D))^T$, with $I(x) = 1$ if $x > 0$ and 0 otherwise, a binary vector, to specify the packet direction sequence. To reduce the number of parameters needed to model the joint distribution for $\mathcal{I}(\underline{x})$, we propose to model $\mathcal{I}(\underline{x})$ based on the Chow-Liu Bayesian Network variant [8]. Here, the joint distribution for a vector of discrete-valued random variables is the model which maximizes the likelihood over the training data under the constraint that the distribution factors as a product of first and second-order probabilities. Hence, based on this special Bayesian Network structure, $P[\mathcal{I}(\underline{x})]$ factorizes as:

$$P[I(x_{j_1})]P[I(x_{j_2})|I(x_{j_1})] \ldots P[I(x_{j_D})|I(x_{j_{D-1}})],$$

where j_1 denotes the root node index of the learned Bayesian Network. To simplify notation in the sequel, we will use I_j to denote $I(x_j)$.

The maximum a posteriori estimates of the probabilities are obtained from frequency counts. For all estimates, we added 1 in the numerator to avoid assigning 0 probabilities. That is, $P[I_j = 1] = \frac{N_j^+ + 1}{T_l + 2}$, with N_j^+ representing the number of web-flows belonging to the flow training set $\mathcal{X}_l$ with positive packet size in the j^{th} position, and $T_l = |\mathcal{X}_l|$. Similarly, $P[I_j|I_m] = \frac{P[I_j, I_m]}{P[I_m]}$, with $P[I_j = 1, I_m = 1] = \frac{N_{jm}^{++} + 1}{T_l + 4}$ and N_{jm}^{++} representing the number of training web flows that have positive packet size in the $\{j, m\}$ position pair. Similarly, $P[I_j = 0, I_m = 1] = \frac{N_{jm}^{0+} + 1}{T_l + 4}$, $P[I_j = 1, I_m = 0] = \frac{N_{jm}^{+0} + 1}{T_l + 4}$, and $P[I_j = 0, I_m = 0] = \frac{N_{jm}^{00} + 1}{T_l + 4}$. Note that $P[\mathcal{I}(\underline{x})]$ is a product of D *unweighted* probabilities, giving an (unweighted) aggregate anomaly score over the D dimensions, for the packet direction sequence. We will exploit the low-order *constituent* probabilities of $P[\mathcal{I}(\underline{x})]$ to obtain derived features for input to our classifier.

Next, for all single features and all pairs of features, considering only the *positive* entries (non-zero packet sizes), we propose to model these continuous distributions using Gaussian Mixture Models (GMMs), and use both first and second order *mixture p-values* [5] to quantify the flow anomalies with respect to independently learned GMMs for each individual and pairwise feature pair. Following the development in [5], given a second order mixture null, the p-value – the probability that a two-dimensional feature vector will be more extreme than the given observed vector $\underline{y} = (x_i, x_j)$ – is $p_{ij}^+(\underline{y}) = \sum_{k=1}^{K_{ij}} P[M = k|\underline{y}] e^{-r_k^2(\underline{y})/2}$. Here, the mixture posterior is $P[M = k|\underline{y}] = \frac{\alpha_k f_{\underline{Y}|k}(\underline{y}|\theta_k)}{\sum_{m=1}^{K_{ij}} \alpha_m f_{\underline{Y}|m}(\underline{y}|\theta_m)}$ and $r_k^2(\underline{y})$ is the squared Mahalanobis distance between $\underline{y}$ and $\underline{\mu}_k$, with $f_{\underline{Y}|k}$ denoting the *pdf* of the k^{th} GM component.

Note that $p_{ij}^+(\underline{y})$ is the expected p-value, with the expectation taken with respect to the mixture posterior pmf. In a similar fashion, one can also calculate a set of mixture-based p-values for single (univariate) features, denoted $\{p_i^+(x_i), i = 1 \ldots, D\}$. In this case, complementary error functions are used to measure the p-value conditioned on each mixture component, with the mixture-based p-value again the expected p-value. Based on the Bayesian Network probabilities and the collection of GMM null distributions, we can compute the vector of derived p-value based features for each flow from the raw features $(\underline{x}, \mathcal{I}(\underline{x}))$.

Let us define $p_i(x_i)$ and $p_{ij}(\underline{y})$ the following way:

$$p_i(x_i) = \begin{cases} p_i^+(x_i) & I_i = 1 \\ 1 & else \end{cases}, p_{ij}(\underline{y}) = \begin{cases} p_{ij}^+(\underline{y}) & I_i = 1, I_j = 1 \\ 1 & else \end{cases}.$$

We then have the derived feature vector $\underline{z}$,

$$\underline{z} = (P[I_{l_1}], P[I_{l_k}|I_{l_{k-1}}], p_i(x_i), p_{ij}(\underline{y}) : \forall i, j, k,\ 1 < k \le D,\ 1 \le i < j \le D) \in (0, 1]^{2D + \binom{D}{2}}.$$

2.3. Classification Model, Learning Objective, and Active Learning Strategy

The classifier model is a variant of a logistic regression model, which uses logs of the entries of the derived p-value based feature vector $\underline{z}$ and with *non-negative* constraints on the weights on these features. For the c^{th} class, let

$$f(\underline{z}; \underline{\beta}^{(c)}) = \exp(\beta_0^{(c)} - \sum_{i=1}^{2D + \binom{D}{2}} \beta_i^{(c)} \log z(i)),$$

where the model parameters for the c^{th} class are $\{\beta_i^{(c)}, i = 0, \ldots, 2D + \binom{D}{2}\}$. Using ω_2, ω_1, and ω_0 to respectively denote the known botnet, the known normal, and the unknown class (which represents the union of all "unknown unknown" classes, i.e., those that have not yet been discovered or labeled), we then have:

$$P(\Omega = \omega_c|\underline{z}) = \frac{f(\underline{z}; \underline{\beta}^{(c)})}{\sum_{c'} f(\underline{z}; \underline{\beta}^{(c')})} \quad \text{with } \beta_i^{(c)} \geq 0, \forall i > 0, c = 0, 1, 2. \tag{1}$$

The inclusion of ω_0 allows for the possibility that there are *unknown* classes in a test data batch, beyond a botnet class (ω_2) that has already been discovered (for which there are labeled flow examples). Note also that (1) is for the case of one normal class and one known botnet class. This can of course be generalized if there are *multiple* known botnet classes. Moreover, initially in our scenario, there are *no* known botnet classes, i.e., ω_2 is only instantiated once a sample from a botnet class is selected and actively labeled.

The non-negatively constrained logistic regression model has been shown to produce a highly *sparse* solution, with only "informative features" having non-zero weights [5].

Let us assume at the t^{th} oracle labeling we have a set of labeled samples $\mathcal{Z}_l^{(t)} \in \mathbb{R}^{T_l^{(t)} \times (2D + \binom{D}{2})}$, with associated labels $C_l^{(t)}$ and unlabeled samples $\mathcal{Z}_u^{(t)} \in \mathbb{R}^{T_u^{(t)} \times (2D + \binom{D}{2})}$, with no ground truth. Using Q to denote the uniform distribution on $\{\omega_0, \omega_1, \omega_2\}$ if a sample from class ω_2 has already been labeled, and to denote the uniform distribution $\{\omega_0, \omega_1\}$ otherwise, i.e.,

$$\begin{cases} Q = \{q_{\omega_0}, q_{\omega_1}, q_{\omega_2}\} = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\} & if\ \omega_2 \in C_l^{(t)} \\ Q = \{q_{\omega_0}, q_{\omega_1}\} = \{\frac{1}{2}, \frac{1}{2}\} & otherwise. \end{cases} \tag{2}$$

Our AL-based, semisupervised, regularized negative posterior log-likelihood learning objective, where we use novel *maximum entropy* regularization on the unlabeled sample subset [5], is:

$$\mathcal{J}_{\text{maxEnt}}^{(t)} = -\sum_{(\underline{z},c) \in (\mathcal{Z}_l^{(t)}, C_l^{(t)})} \alpha_c^{(t)} \log P[\Omega = \omega_c|\underline{z}] + \gamma \sum_{\underline{z} \in \mathcal{Z}_u^{(t)}} d((Q||P[\Omega|\underline{z}]). \tag{3}$$

Here, in minimizing (3), we aim to maximize the class posterior log-likelihood on the labeled samples, but *also* to maximize the class *uncertainty* of the posterior on the unlabeled samples, where $d(Q||P) = \sum_c q_c log(q_c/p_c)$, the Kullback-Leibler distance [9] (cross entropy) between probability mass functions $Q = \{q_c\}$ and $P = \{p_c\}$. Compared to the previously proposed *minimum entropy* regularization approach [10], which minimizes decision uncertainty on unlabeled samples, (3) avoids over-training, especially when the rare category (botnet) is underrepresented, i.e., during the early stages of AL [5]. Moreover, by maximizing class entropy on unlabeled samples, unknown clusters (with no labeled samples) will have high class uncertainty, hence facilitating AL identification of unknown unknowns (zero-day threats).

The proposed learning objective (3) is a *convex* objective function, with a unique global minimum, unlike [10] and [5]. Also, unlike [5], we distinguish rare and unknown-unknown in this paper, primarily to preserve convexity. We optimize (3) via projected gradient descent, which is guaranteed to reach the global minimum due to the objective's convexity. $\alpha_c^{(t)}$ is chosen to balance the effective sample size between the two classes, whereas γ is chosen via cross validation (CV). When there is not enough botnet traffic for CV, γ is set to $\frac{T_l^{(t)}}{T_u^{(t)}}$.

Finally, we use pool-based AL, wherein the oracle ground-truth labels the informative samples from the unlabeled batch sequentially forwarded by the learner. We use the best AL strategy proposed in [5] – *most likely unknown (MLU) sampling* – to pick the unlabeled sample that has the highest probability of belonging to the unknown class, as evaluated by $P(\Omega = \omega_0|\underline{z})$. Alternatively, mixed strategies may be considered in future work (that balance unknown class discovery with classification accuracy).

3. EXPERIMENTAL SETUP AND RESULTS

The overall AL system is illustrated in Fig. 1. We obtained

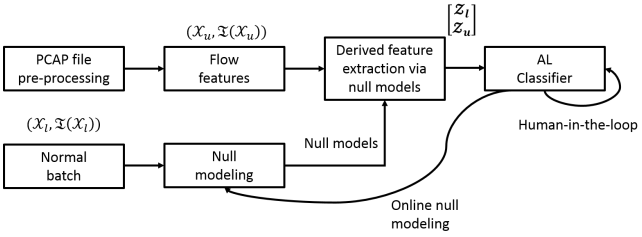


Fig. 1: The overall AL system architecture.

normal traffic from LBNL traces [11], which were based on monitoring a medium-size enterprise network with more than 100 hours of web activities, covering 22 subnets. Specifically, the experiments in this paper are based on the same three PCAP files as used in previous work [7]. Zeus bots are well-known for their detection evasion techniques such as randomization of proxy servers and/or port numbers, which make them very difficult to detect; Zeus variants have become the most popular botnet application on the Internet today, especially for cybercrime activities [12]. We obtained Zeus PCAP files from [13] and [14]. Both C&C and non C&C traffic are combined and used in our experimentation. In Table 1, the sample sizes of these web and botnet traffic traces are shown. All traffic uses TCP as the transport protocol. We would like to use as much flow information as possible in making clas-

sification decisions. Since more than 90% of the TCP flows have no less than 10 packets after the three-way handshake, we use the first 10 packets from each flow after the three-way handshake.

Table 1: Normal web and botnet flow sizes.

Application	Number of Flows Used
LBNL Web [11]	9972
VRT Zeus [13]	64
ISOT Zeus [14]	23

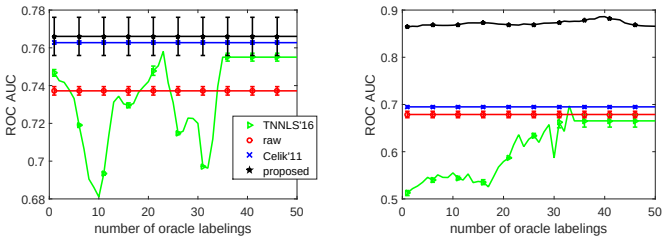
3.1. Performance Metrics

For botnet anomaly detection, we are interested in the following generalization performance criteria. One is sensitivity: the ratio of botnet flows that are classified as truly botnet; the other is specificity: the ratio of web flows that are falsely classified as botnet flows. To give a comprehensive trade-off between the sensitivity and specificity, we use ROC AUC as the generalization measure on the test batch, in all of our experiments. The ROC AUC is calculated as a function of the number of active labelings.

3.2. Experimental Results

One third of the normal batch is first randomly subsampled without replacement from the whole normal batch and used to train the Bayesian Network and all the marginal and pairwise packet size GMMs. The derived feature vector $\underline{z}$ is then obtained for each flow sample. Half of the remaining two thirds of the normal traffic are treated as unlabeled and combined with half of the unlabeled botnet traffic for active learning (semisuperived AL training), and the remaining samples are used for testing (measuring generalization performance). Generalization performance is averaged over 5 random training-test splits.

We are interested in the effectiveness of the proposed p-value based feature vector, compared with alternative feature representations. Besides performance using the raw features $(\underline{x}, \mathcal{I}(\underline{x}))$, we also compare with popular derived flow-based features for network traffic classification. Through a correlation-based filtering process, [3] identified 8 among the 248 flow based features described in [15] as highly discriminative for different network flows. [4] additionally used IP-ratio and goodput in their experiment. We denote this feature set as CSET'11. Since the botnet applications are from different domains than the web traffic, we only use the time-independent features from [4], i.e., RTT-samples and goodput are not used. Also compared is the feature representation proposed in [5], i.e., without the use of a Bayesian Network to capture presence/absence features for packets, but using GMM based p-values. We denote this feature set as TNNLS'16.



(a) ISOT Zeus (b) VRT Zeus

Fig. 2: Comparison of different feature representations, with bars indicating one standard deviation.

In Fig. 2, we compare different feature representations, using MLU as the AL sample selection strategy. As expected, the proposed feature set greatly outperforms the CSET'11 feature set, as well as greatly outperforming the feature set proposed in [5] for general-purpose anomaly detection.This is because in [4], the absence of a data packet in a given direction is ignored; however, this information is seen to be highly discriminative between web and Zeus. Moreover, while the ROC AUC curves are flat for some methods, this does not mean the classifier is not improving with more oracle labelings – e.g., the test set error rate curves tend to be decreasing.

The proposed maxEnt model (3) produces a highly sparse solution, e.g., for VRT Zeus, 95.2% of the $\{\beta_i^{(2)}, i = 1, \ldots, 2D + \binom{D}{2}\}$ are zero by the fiftieth AL iteration, essentially eliminating the effects of these (uninformative) features.

Additional experimental results are reported in [16].

4. DISCUSSION OF DETECTION EVASION

By using the proposed feature representation, several common botnet evasion schemes become ineffective. For example, the random back-off presented in [17] can be completely overcome simply by ignoring (as above) timing-based features such as goodput or RTT, see [4]. Also, encryption of data packets will not affect performance using the proposed feature representation as it only requires packet-size information of the bidirectional TCP flows. Obviously, randomization of port numbers and domain names (fast flux) would also be ineffective since these features are not used. However, flow "perturbation" or "noise injection" [17], can significantly impact the performance of our system. For example, if the intruder has detailed knowledge of the normal traffic patterns at the detection point, then they can adaptively groom their bidirectional packet-size sequence to defeat a detection scheme predominantly based on such features. But such evasion techniques have significant overhead and require packet-traffic telemetry that may not be available to the bot slaves or master end-hosts. Hence, evasion is possible but may have very high implementation complexity and require adaptation overhead leading to less covert malware [17].

5. REFERENCES

[1] Sérgio SC Silva, Rodrigo MP Silva, Raquel CG Pinto, and Ronaldo M Salles, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. 378–403, Feb. 2013.

[2] Guofei Gu, Roberto Perdisci, Junjie Zhang, Wenke Lee, et al., "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection," in *Proc. The 17th USENIX Security Symposium*, San Jose, CA, Jul. 2008, pp. 139–154.

[3] Wei Li, Marco Canini, Andrew W Moore, and Raffaele Bolla, "Efficient application identification and the temporal and spatial stability of classification schema," *Computer Networks*, vol. 53, no. 6, pp. 790–809, Mar. 2009.

[4] Z Berkay Celik, Jayaram Raghuram, George Kesidis, and David J Miller, "Salting public traces with attack traffic to test flow classifiers," in *Proc. USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, San Francisco, CA, Aug. 2011.

[5] Zhicong Qiu, David J Miller, and George Kesidis, "A maximum entropy framework for semisupervised and active learning with unknown and label-scarce classes," *IEEE Trans. on Neural Network and Learning System*, Jan. 2016.

[6] Fatih Kocak, David J Miller, and George Kesidis, "Detecting anomalous latent classes in a batch of network traffic flows," in *Proc. The 48th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, Mar 2014, IEEE, pp. 1–6.

[7] Zhicong Qiu, David J Miller, and George Kesidis, "Detecting clusters of anomalies on low-dimensional feature subsets with application to network traffic flow data," in *Proc. 25th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Boston, MA, Sep. 2015, pp. 1–6, IEEE.

[8] C Chow and C Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. on Information Theory*, vol. 14, no. 3, pp. 462–467, May. 1968.

[9] Solomon Kullback and Richard A Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, pp. 79–86, Mar. 1951.

[10] Yves Grandvalet and Yoshua Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Advances in Neural Information Processing Systems 17*, Vancouver, B.C., Canada, Dec. 2004, pp. 529–536.

[11] "LBNL/ICSI Enterprise Tracing Project," `http://www.icir.org/enterprise-tracing`.

[12] Dune Lawrence, "The hunt for the financial industry's most-wanted hacker," `http://www.bloomberg.com/news/features/2015-06-18`, Jun. 2015.

[13] "VRT Labs - Zeus Trojan Analysis," `https://labs.snort.org/papers/zeus.html`.

[14] Sherif Saad, Issa Traore, Ali Ghorbani, Bassam Sayed, David Zhao, Wei Lu, John Felix, and Payman Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," in *The 9th Annual International Conference on Privacy, Security and Trust (PST)*, Montreal, QC, Canada, Jul. 2011, IEEE, pp. 174–180.

[15] Andrew Moore, Denis Zuev, and Michael Crogan, "Discriminators for use in flow-based classification," Tech. Rep. RR-05-13, Queen Mary and Westfield College, Department of Computer Science, Aug. 2005.

[16] Zhicong Qiu, David J Miller, and George Kesidis, "Flow based Botnet Detection through Semi-Supervised Active Learning," Tech. Rep. CSE-16-010, CSE Dept, PSU, Sept. 12, 2016, http://www.cse.psu.edu/research/publications/tech-reports/2016/CSE-16-010.pdf.pdf/view.

[17] Elizabeth Stinson and John C Mitchell, "Towards systematic evaluation of the evadability of bot/botnet detection methods.," in *Proc. USENIX Workshop on Offensive Technologies (WOOT)*, Boston, MA, Jul. 2008, vol. 8, pp. 1–9.