A Hybrid Learning System to Mitigate Botnet Concept Drift Attacks

Zhi Wang[1], Meiqi Tian[1], Xiao Zhang[2], Junnan Wang[3], Zheli Liu[1], Chunfu Jia[1], Ilsun You[4]
[1]College of Computer and Control Engineering, Nankai University, China
[2]R&D, Department of Engineering, Palo Alto Networks, USA
[3]Institute of Information Engineering, Chinese Academy of Sciences, China
[4]Department of Information Security Engineering, Soonchunhyang University, Korea
zwang@nankai.edu.cn, 2120160398@email.nankai.edu.cn, xizhang@paloaltonetworks.com, wangjn@mail.nankai.edu.cn, {liuzheli, cfjia}@nankai.edu.cn, ilsunu@gmail.com

Abstract

Botnet is one of the most significant threats for Internet security. Machine learning has been widely deployed in botnet detection systems as a core component. The assumption of machine learning algorithm is that the underlying data distribution of botnet is stable for training and testing, however which is vulnerable to well-crafted concept drift attacks, such as mimicry attacks, gradient descent attacks, poisoning attacks and so on. So, machine learning itself could be the weakest link in a botnet detection system. This paper proposes a hybrid learning system that combines vertical and horizontal correlation models based on statistical p-values. The significant diversity between vertical and horizontal correlation models increases the difficulty of concept drift attacks. Moreover, average p-value assessment is applied to fortify the system to be more sensitive to hidden concept drift attacks. SIM and DIFF assessments are further introduced to locate the affected features when concept drift attacks are recognized, then active feature reweighting is used to mitigate model aging. The experiment results show that the hybrid system could recognize the concept drift among different Miuref variants, and reweight affected features to avoid model aging.

Keywords: Malware detection, Machine learning, Concept drift, Vertical correlation, Horizontal correlation.

1 Introduction

Botnet is a network of compromised hosts, known as bots or zombies, that could be remotely controlled by an attacker, i.e., botmaster, in the Internet [1-2]. Botnet is one of the most significant threats to the Internet [3]. With enormous cumulative bandwidth and computing capability, botnet becomes the most important and powerful tool available for cheaper and faster malware deployment across the Internet. Botnet is also favored by non-professional attackers with relatively little experience due to its easy adoption and broader scale. As reported by ZingBox [4], more than 400,000 IoT devices infected with the Mirai [5] are for sale. Anyone who is willing to pay the price can quickly hire and assemble a Botnet-as-a-Service (BaaS) and launch large scale cyber attacks. The Mirai botnet controls various IoT devices, including IP cameras, home routers, printers, DVRs and TV receivers, by exploiting the insecure default passwords on IoT products. The Mirai botnet launched extremely huge Distributed Denial of Service (DDoS) attacks against Krebs on Security [6], OVH [7], Dyn [8] and Lonestar Cell [9] with 600Gbps in volume at its peak.

AV-Test [10] reports that on average over 390,000 new malicious programs are detected every day. The enormous volume of new malware variants renders manual malware analysis inefficient and time-consuming. Nowadays, machine learning has been widely used in botnet detection system as a core component [3][11-14]. However, with financial motivation, attackers keep learning new malware detection methods and evolving their evasion techniques. Nowadays, over 70% of the advanced malware uses one or more evasion techniques to bypass detection [15].

The assumption of machine learning algorithm is that the underlying malicious data distribution is stable for training and testing. However, such assumption is vulnerable to well-crafted concept drift attacks, such as new communication channels [16-21], mimicry attacks [22-23], gradient descent attacks [22-23], poison attack [24], and so on. To build secure and sustainable detection system against evasive botnet, recognizing the concept drift of underlying malicious data is very important for machine learning models. In this paper, we combine diverse vertical and horizontal correlation learning models based on p-values, which increase the prediction quality and sensitivity to hidden concept drift attacks.

Vertical correlation model focuses on the life cycle of a single botnet sample, such as BotHunter [25]; while horizontal correlation learning approach builds detection model based on the behavior similarity among a large number of botnet variants, such as BotFinder [26]. There are significant differences between vertical and horizontal correlation learning models. From the prediction results and p-values provided by these two diverse models, we could

*Corresponding author: Ilsun You; E-mail: ilsunu@gmail.com
DOI: 10.6138/JIT.2017.18.6.20171003

obtain more insights into the hidden botnet concept drifting. In a nutshell, this paper makes the following contributions:

- We are the first to combine vertical and horizontal learning models in botnet detection. The proposed system leverages on statistical p-values to mine internal patterns of botnet traffic from diverse perspectives.
- To prevent the threat of concept drift attacks against machine learning systems, we extend the traditional decision methods of vertical and horizontal learning models, which are based on commonly-used fixed and empirical threshold. In contrast, we introduce a system based on fine-grained statistical p-values which could be used to recognize concept drift attacks before the performance starts to degrade.
- Upon identification of concept drift, we apply SIM and DIFF algorithms to assess its effect on the predictive features, and reweight affected features to mitigate model aging.

The remainder of this paper is outlined as follows: In Section 2, we review the related works. Section 3 presents the architecture of our hybrid botnet detection system, and describes each component. Section 4 shows our experiments performed to assess the recognition of underlying concept drift. In Section 5, we discuss the limitations and future work, and in Section 6 we summarize our results.

2 Related Works

Nowadays, machine learning has been widely used in botnet detection system as a core component. However, Arce [27] pointed out that machine learning itself could be the weakest link in the security chain. By exploiting the knowledge of the machine learning (ML) algorithm, many well-crafted evasion approaches have been proposed to evade or mislead ML models [28]. Figure 1 extends the graph in Srndic and Laskov [22] and shows the levels of attacker's knowledge required to launch attacks against machine learning detection systems.

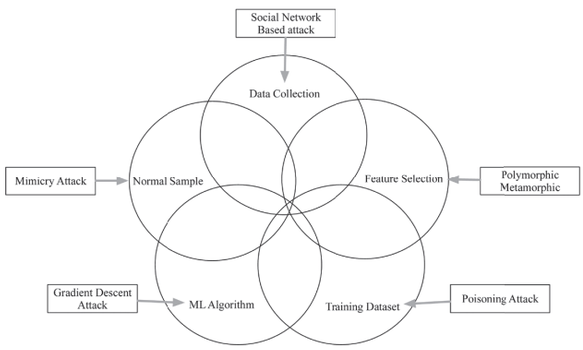

Figure 1 Attack Methods against Machine Learning Detection Approaches at Different Knowledge Levels

Botnet attackers have begun to exploit many stealthy C&C channels, such as social network [19-20], email protocol [16][29], SMS [16] and Bluetooth [17]. Kartaltepe et al. [20] proposed social network based botnet which abuses trusted popular websites, such as twitter.com, as C&C servers. Singh et al. [29] evaluated the viability of using harmless-looking emails to delivery botnet C&C message. Social network traffic and email traffic are beyond the data collection scope of machine learning based detection models. What makes these proposals difficult to be monitored is the usage of trusted and popular websites or email servers as C&C servers. First of all, trusted websites or email servers have very good reputation and usually are whitelisted from botnet detection systems. Secondly, the trusted websites or email services are very popular and have very large usage volume. As a result, light-weight occasional C&C traffic is unlikely to be noticed. However, the new botnets use the centralized architecture that all bots communicate with C&C server directly. The central C&C server is a potential single point of failure. If the C&C server is exposed to the defender, the botnet is easy to be dismantled.

Mimicry attack refers to the techniques that mimic benign behaviors to reduce the differentiation between the malicious events and benign events. Wagner and Soto [30] demonstrated the mimicry attack against a host-based IDS via mimicking the legitimate sequence of system calls. Srndic and Laskov [22] presented a mimicry attack against PDFRate [31], a system to detect malicious pdf files based on random forest classifier.

The gradient descent is an optimization process to iteratively minimize the distance between malicious points and benign points. Srndic and Laskov [22] applied a gradient descent kernel density estimation attack against the PDFRate system that uses SVM and random forest classifier. Biggio et al. [23] demonstrated a gradient descent component against the SVM classifier and a neural network.

Poisoning attacks work by introducing carefully crafted noise into the training data. Biggio et al. [24] proposed poisoning attacks to merge the benign and malicious clusters that makes learning model unusable.

Therefore, malware sample characteristics are not stable but change with time. For ML based malware detectors, they are designed under the assumption that the training and testing data follow the same distribution. Such assumption is vulnerable to concept drift attacks. One of the concept drift mitigation approaches is to recognize and react to recent concept changes. Demontis et al. [11] proposed an adversary-aware approach to proactively anticipates the attackers. Deo et al. [32] presented a probabilistic predictor to assess the underlying classifier

and retraining model when it recognizes concept drift. Transcend [33] is a framework to identify model aging in vivo during deployment, before the performance starts to degrade. In this paper, we combine diverse horizontal and vertical correlation learning algorithms together, and apply conformal evaluation to understand the changes of botnet underlying data distribution.

3 Hybrid Botnet Detection System

Driven by financial motivation, attackers keep evolving botnet perpetually using evasion tricks to avoid detection, especially to bypass widely deployed ML-based models. Many ML-based detection models calculate a score to a new approaching sample describing the relationship between the known botnet model and the new one. Then detectors compare the score with a fixed and empirical threshold to make final verdict decision. The threshold usually fits, and sometimes even overfits, the old training dataset very well. However, the performance degenerates to the new ever-changing malicious dataset with time. In this paper, we propose a hybrid botnet detection system (HBDS) that combines vertical life-cycle algorithm and horizontal traffic similarity algorithm based on statistical p-values. HBDS is robust to botnet concept drift attacks. The average p-value is introduced into HBDS to recognize the hidden concept drift.

Figure 2 depicts the architecture of HBDS that includes five components: non-conformity measure (NCM), conformal learning, concept drift recognition, feature assessment, and active reweighting. The HBDS is an open framework that any machine learning model based on fixed empirical threshold can be integrated into HBDS as an underlying independent NCM. Diverse NCM models provide insights of the botnet concept drift from different perspectives. In this paper, we select vertical correlation classifier BotHunter and horizontal correlation classifier BotFinder as the underlying NCMs. The conformal learning component uses p-values to carry out further statistical analysis based on NCM scores. The p-value is more fine-grained than threshold that can be used to observe the gradual decay of detection model. And p-value is comparable between different models while the NCM scores are not comparable among different models. The concept drift recognition component uses the average p-value (APV) algorithm to detect the concept drift of botnet data distribution between two different time windows. The feature assessment component applies SIM and DIFF algorithms to locate the features that are affected by identified concept drift. The active reweighting component dynamically adjust the weight of affected features to mitigate model aging.

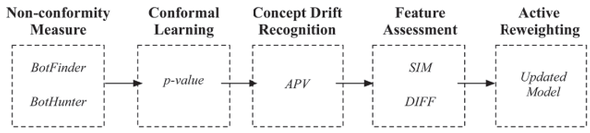

Figure 2 The Framework of Hybrid Detection System

3.1 Non-Conformity Measure

Many machine learning algorithms are in fact scoring classifiers: when trained on a set of observations and fed with a test object x, they could calculate a prediction score $s(x)$ called scoring function. Any scoring classifiers using a fixed and empirical threshold can be introduced into our system as an underlying NCM. NCMs describe botnet concept from different perspectives. Currently, we select BotHunter and BotFinder as the NCMs. BotHunter models the life cycle of botnet from the vertical perspective, while BotFinder selects time related features and traffic volume features to build detection model from the horizontal perspective. The diversity of selected NCMs increases the complexity of the successful concept drift attack, because attackers need to obtain more knowledge to construct concept drift attacks against HBDS than traditional single model detection systems.

The input of the NCM is a known sample set and an unknown sample, and the output is a score that describes the similarity or dissimilarity of the unknown sample to the known sample set.

This paper hybrids two different machine learning models: BotHunter and BotFinder, as shown in Table 1.

Table 1 The Non-Conformity Measures of Hybrid Botnet Detection System

	Object	Features	ML type	Algorithm
BotHunter	Dialog	5	Classification	Life-cycle model
BotFinder	Trace	7	Clustering	CLUES

BotHunter is a multi-dialog-based vertical correlation algorithm. First, BotHunter establish botnet life-cycle model according to the behavior sequence pattern of botnets; Then it maps a set of host dialogues to a pre-learned life-cycle model and calculate a score to describe how close between the dialog and the model. When the dialog correlation algorithm shows that a host dialog pattern maps sufficiently close to the life-cycle model based on a threshold, the host is declared infected. According to the introduction of BotHunter, we construct a botnet life-cycle model in 4 layers and 7 states as shown in Figure 3.

BotFinder is a detection method that does not require deep packet inspection. First, BotFinder groups netflows

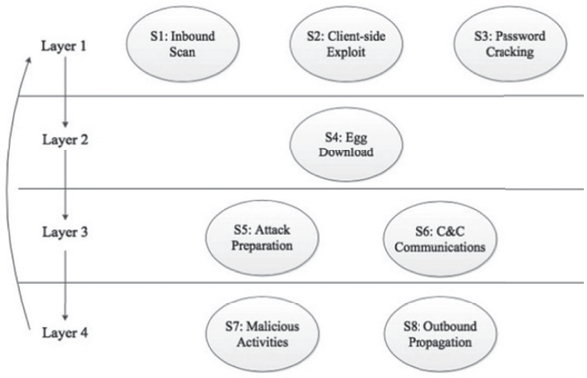

Figure 3 The Architecture of Life-Cycle Model

with the same source IP, destination IP, destination port number, and communication protocol into trace; Then, it extracts traffic volume features from trace, such as the average number of sent bytes, the average number of received bytes, and time related features of trace, such as the average time interval, the average duration, and frequency calculated by Fast Fourier Transformation (FFT) algorithm. BotFinder uses the CLUES algorithm to cluster the similar traces of a botnet family, and builds detection model for each class of this family. This method can effectively identify the botnet network traffic similarity between different botnet variants, and give a prediction based on the optimal threshold fitting the training dataset.

3.2 Conformal Learning

Once NCMs are selected, conformal learning component computes a p-value P_{z^*}, which in essence for a new object z^*, represents the percentage of objects in $\{x \in C, \forall\ C \in D\}$, (i.e., the whole dataset) that are equally or more estranged to C as z^*, and we will get a number between 0 and 1. The algorithm is shown in Algorithm 1.

Algorithm 1 P-value calculation

Require: Dataset $D = \{z_1, ..., z_n\}$, sequence of objects C, non-conformity measure A, and new object z^*

Ensure: p-value P_z^*

1: Set provisionally $C = C \cup \{z^*\}$
2: **for** $i \leftarrow 1$ to n do
3: $\alpha \leftarrow A(C\backslash z_i, z_i)$
4: **end for**
5: $\text{Pz}^* = \dfrac{|\{j: \alpha \geq \alpha_{z^*}\}|}{n}$

P-value measures the fraction of objects within D, that are at least as different from a class C as the new object z^*. For instance, if C represents the set of malicious activities, a high p-value P_{z^*} means that there is a significant part of the objects in this set is more different than z^* with C, on the other words, z^* is more similar to these malicious activities than the objects that already marked malicious. Therefore, the prediction result based on a high p-value shows a high credibility. P-values are directly involved in our discussion of concept drift.

The p-value is comparable between different learning models, while the NCM scores are not comparable among different models, as shown in Figure 4. The p-value is more fine-grained than threshold, and is more sensitive to concept drift attacks. The concept drift recognition component uses the average p-value algorithm to recognize hidden concept drift.

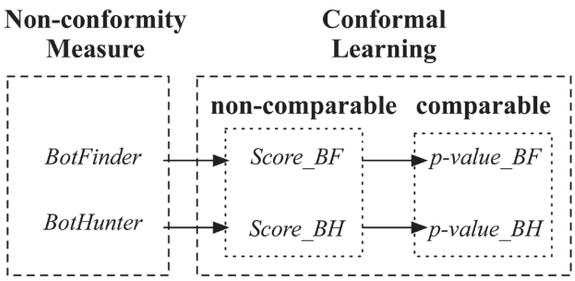

Figure 4 The Comparable Statistical p-values

Note. The Conformal Learning Component Transform Non-Comparable NCM Scores to Comparable Statistical p-values.

3.3 Concept Drift Recognition

We use the average p-value (APV) algorithm based on time windows to recognize concept drift attacks, as shown in Figure 5. We group the botnet samples into different time windows according to their time stamps on the timeline. We calculate p-values for each botnet sample in a time window, and compute the APV value for each time window. Note that, the number of APV for each time window depends on the number of selected NCMs. In this paper, each time window has two APVs for the vertical and horizontal NCMs.

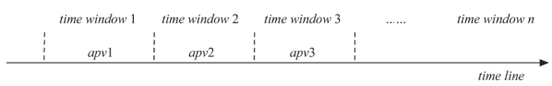

Figure 5 The Conformal Learning Component Calculates APV for Each Time Windows

The p-values are comparable, and the APV scores are also comparable between different time windows and even in the same window with different underlying NCMs. The change of APV value between different time window reflects the change of underlying botnet data distribution. APV could identify botnet *gradual* concept drift between different time windows. In the same time window, the

difference between APV scores calculated from different NCMs reflect the affection of concept drift to different learning models which can detect the botnet *sudden* concept drift.

If the APV score of a certain detection model decreases with time, it shows that the current concept of the botnet data distribution is gradually different from the old concept learnt from previous botnet data, and indicates that the detection model is suffering from concept drift attack. But the decay of threshold based detection performance may not be observed immediately when concept drift is found. The concept drift attack usually is a gradual process. Only when the variation of the underlying data distribution exceeds the boundary of the threshold, the detection model based on fixed threshold starts make poor decisions. If the APV score does not decrease in the new time window, it means in the current time window, the distribution of botnet data does not have concept drift from this observing perspective.

3.4 Feature Assessment

When a detection model finds concept drift attack in current time window, we will use the SIM and DIFF algorithms to locate the features that affected by concept drift. The SIM and DIFF algorithms are used to evaluate the contribution of the features in a data set, as shown in Algorithm 2 and Algorithm 3. $SIM[i]$ represents the effect of the i^{th} feature on the average distance between every two samples with the same label; $DIFF[i]$ represents the effect of the i^{th} feature on the average distance between every two samples with different labels.

If $DIFF[i]$ increases, it means that the concept drift affects the i^{th} feature, leave the sample away from the known sample set in the current time window. The value of $SIM[i]$ indicates the aggregation degree of the sample at the i^{th} feature, the smaller value of $SIM[i]$ indicates that the sample is more stable at this feature; the larger value of $SIM[i]$ means the greater noise from this feature.

Algorithm 2 SIM Algorithm

Require: feature vectors $X = \{x_1, x_2, ..., x_n\}$, labels $Y = \{y_1, y_2, ..., y_n\}$, entries in Y without repetition uY
Ensure: SIM coefficients

```
for i ← 1 to d do
    X_i = i^th column of X
    SIM[i] = 0
    for class in elements of uY do
        formclass = X_i[where Y == class]
        pdist = pairwise_distances(fromclass)
        SIM[i] = SIM[i] + (sum(pdist)/length(pdist))
    end for
SIM[i] = SIM[i]/length(uY)
end for
```

Algorithm 3 DIFF Algorithm

Require: feature vectors $X = \{x_1, x_2, \ldots, x_n\}$, labels $Y = \{y_1, y_2, \ldots, y_n\}$, entries in Y without repetition uY
Ensure: DIFF coefficients

```
for i ← 1 to d do
    X_i = i^th column of X
    DIFF[i] = 0
    for class in elements of uY do
        formclass = X_i[where Y == class]
        notclass = X_i[where Y != class]
        pdist = respective distanced between elements
        in fromclass and elements in notclass
        DIFF[i] = DIFF[i] + (sum(pdist)/length(pdist))
    end for
DIFF[i] = DIFF[i]/length(uY)
end for
```

3.5 Active Reweighting

When concept drift is recognized by a detection model, we will reweight the affected features according to the SIM and DIFF results to actively update the model before model aging. The formula for the feature reweighting is $W_i = 1/SIM[i] + DIFF[i]$. By updating the weight, we can reduce the weight of the feature that is significantly influenced by concept drift attack, and increase the anti-aging ability of the detection model.

4 Experiments

In this paper, we evaluate this novel approach against public dataset provided by Malware Capture Facility project [35]. They capture long-live real botnet traffic, and public labeled netflow files to malware researchers. The CTU traffic dataset has a long-time span, which is useful to observe the degeneration of detection models and understand concept drift.

The recognition of sudden radical concept drift between different botnet families is not the focus of this paper. We plan to recognize the hidden and gradual concept drift between different variants in the same family that is not noticed by traditional models using fixed and empirical threshold. We selected the Miuref family for our experiment, as Miuref has 4 variants and 8 different traffic records, which are more than other families in the public CTU dataset. Miuref redirects web browser to carry out click fraud or download other malware. The four variants of Miuref are listed in the Table 2. According to the time order of traffic records, we use $V1$, $V2$, $V3$ and $V4$ to denote the 4 variants of Miuref.

In the experiments, based on the time span, we created 2 time windows that $V1$ and $V2$ are grouped into the first

Table 2 The Network Traffic of 4 Variants in Miuref Family

Traffic record number	Time span	Variants
127-1	2015.06.01-2015.06.07	*V*1
127-2	2015.06.09-2015.07.08	*V*1
128-1	2015.06.01-2015.06.07	*V*2
128-2	2015.06.09-2015.07.19	*V*2
169-1	2016.08.03-2016.08.04	*V*3
169-2	2016.08.04-2016.08.04	*V*3
169-3	2016.08.03-2016.08.11	*V*3
173-1	2016.08.04-2016.08.11	*V*4

time window whose traffic data were collected in 2015, while *V*3 and *V*4 are grouped into the second time window whose traffic data were captured in 2016.

To identify the concept drift between different time windows and from different perspectives, we use dimension reduction algorithm tSNE [34] and statistical p-values to assess the change of underlying traffic data distribution.

The tSNE is an algorithm to visualize high-dimensional dataset by dimensionality reduction, which maps the high-dimensional points into two or three dimensions and keeps the distance structure that the close points in high-dimensional space remain close to each other on the low dimension.

Figure 6 shows the underlying traffic data distribution and p-value significant levels of Miuref family in two different time windows, which use vertical correlation algorithm as NCM. The left subfigure in Figure 6 shows the data distribution of *V*1 and *V*2 in tSNE space and the p-values for each point. The different colors denote the value of p-values that the darker red means the point has high p-value, while the lighter red means the point has low p-value. The right subfigure in Figure 6 shows the data distribution and p-value significant levels of all Miuref variants in the tSNE space. We can see that from the vertical perspective, the Miuref family has very slight concept drift between the two time windows in 2015 and 2016, because the traffic data distribution and p-value significant levels are almost stable without much change. In addition, the traffic data distribution of Miuref becomes more centralized after absorbing the traffic data of *V*3 and *V*4 variants captured in 2016, because in the middle of the first subfigure there are some points with low p-values, while in the second subfigure such p-values change to be almost zero.

Figure 7 shows the changes of APVs of 4 Miuref variants using vertical correlation algorithm. For Miuref family, all variants have high APV, and the APV of *V*4 is even higher than 0.8, which is consistent with the Miuref data distribution and p-value significant levels in tSNE

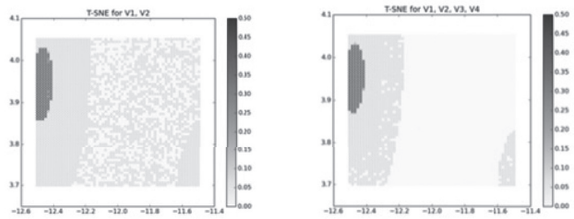

Figure 6 Concept Drift against Vertical Correlation

Note. The data distribution and p-value significant level of Miuref in the tSNE space using vertical correlation algorithm as NCM.

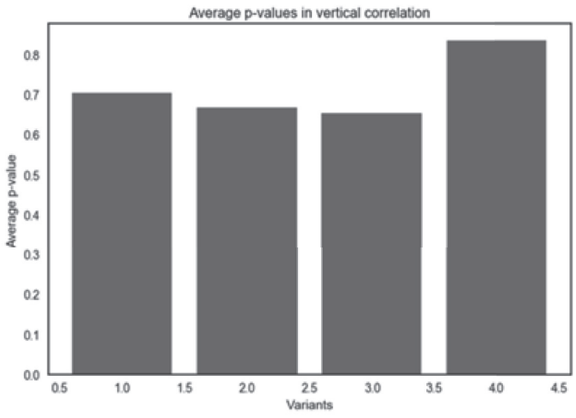

Figure 7 The Change of APVs of 4 Miuref Variants for Vertical Correlation Model

space as shown in Figure 6. So, from the vertical observing perspective, the traffic data of Miuref family have little concept drift, and the vertical correlation model is effective to detect new Miuref variants.

Figure 8 shows the underlying traffic data distribution and p-values of Miuref family using horizontal correlation algorithm as NCM. The left subfigure in Figure 8 shows the data distribution of variant *V*1 and *V*2 in tSNE space and the p-values for each point using horizontal correlation algorithm. The right subfigure in Figure 8 shows the data distribution and p-values of all 4 Miuref variants in the tSNE space using horizontal correlation algorithm. We can see that from the horizontal perspective, the Miuref family has significant concept drift between the two time windows, because between the two subfigures, the data distribution and p-value significant level are obviously changed, especially at the upper left corner in the figure.

Figure 9 shows the changes of APVs of 4 Miuref variants from horizontal perspective. We can see that the APV drops dramatically on *V*4 from 0.7 to 0.4, which means that the data distribution of variant *V*4 changed significantly from horizontal correlation perspective. As shown in Figure 10, most p-values of variant *V*4 are less than 0.4, while the p-values of variant *V*1, *V*2 and *V*3 are

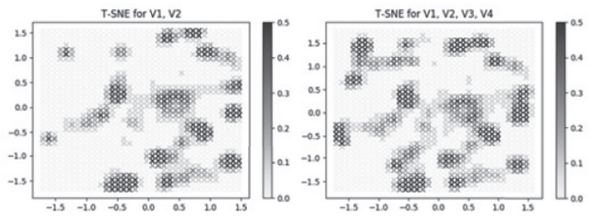

Figure 8 Concept Drift against Horizontal Correlation

Note. The data distribution and p-value significant levels of Miuref in the tSNE space using horizontal correlation algorithm as NCM.

much higher than 0.4. It can be inferred from Figure 8, Figure 9 and Figure 10 that the variant *V*4 is not consistent of family characteristics and *V*4 occurs concept drift.

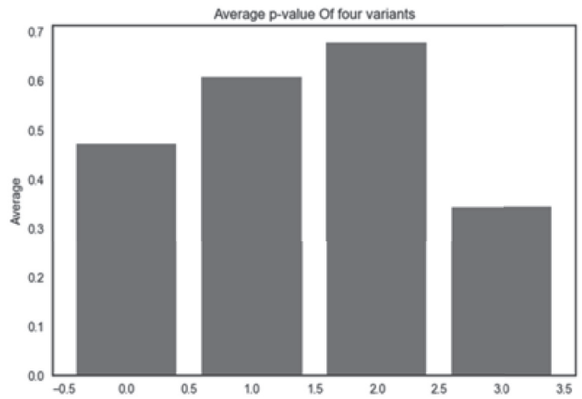

Figure 9 The change of APVs of 4 Miuref variants using horizontal correlation algorithm as NCM

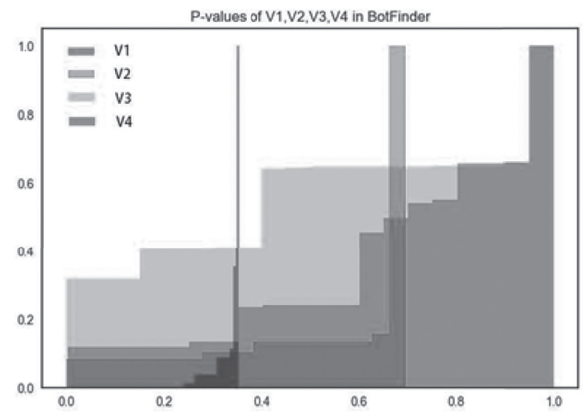

Figure 10 The Cumulative Distribution of p-Values of 4 Miuref Variants Using Horizontal Correlation Algorithm as NCM

After recognized concept drift for horizontal correlation model, we use SIM and DIFF algorithms to assess the effect of concept drift to each predictive feature. SIM score represents the distance between the observed samples that belonged to the same class, while DIFF score represents the distance between the observed samples that belonged to the same class and the samples belonged to all other samples. SIM and DIFF scores reflect the contributions of features in identifying the new variant to its family. Figure 11 shows the SIM and DIFF average scores of five predictive features used by horizontal correlation model. The left subfigure in Figure 11 shows that the average SIM and DIFF scores of features of *V*3 are lower than 0.8. In the right subfigure in Figure 11, the average DIFF scores of the second feature and the forth feature are higher than 1, especially, the average DIFF score of forth feature is up to 1.75. This result confirms that the concept drift happened, and mainly affected the second feature and the forth feature. The average SIM and DIFF scores not only can assess the contribution of features, and provide foundation for feature reweighting in the horizontal correlation model.

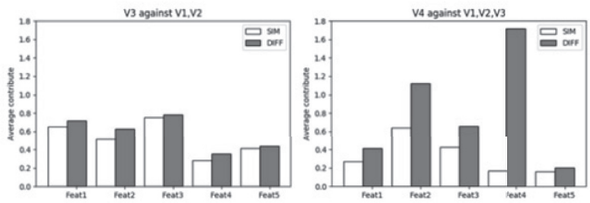

Figure 11 Feature Assessment Results Using SIM and DIFF

Note. The average contribution of 5 predictive features used by horizontal correlation model to assess *V*3 and *V*4 data.

In conclusion, concept drift is the significant factor of causing the model aging problem. We hybrid two much diverse learning models: horizontal and vertical correlation model to analyze malware data from two diverse perspectives. Our hybrid detection system is robust to the concept drift attacks and increases the complexity for evading learning models. We map concept drift of underlying malicious data distribution to tSNE space with p-value as significant levels, so it will be easier for us to understand and recognize concept drift attacks. If concept drift happened, features will be actively reweighting based on the scores of SIM and DIFF to mitigate detection model degeneration.

5 Discussion

Botnets are continuously evolving and new botnets keep showing up. With the increasing prevalence of IP cameras, home routers and smart TVs, more and more smart devices are connected to the Internet every year. However, information security has not always been considered in product design, IoT devices might easily be compromised and abused for cyber attacks. IoT botnets is becoming a significant threat to Internet security.

Building machine learning models of malware behaviors is widely used as a panacea towards effective, scalable, and automatic botnet detection. For the sake of survival and financial motivation, attackers keep learning the latest machine learning based detection systems and evolving evasion techniques to generate new variants. Concept drift is the well-known vulnerability of machine learning which is exploited by attackers to launch well-crafted concept drift attacks artificially, such as mimicry attacks, gradient descent attacks and poisoning attacks.

The real-world malware concepts are not stable but change with time rapidly, which requires machine learning models quickly recognize and adapt to the hidden changes in the underlying malware data distribution. There are two types of concept drift: sudden drift and gradual moderate drift. Sudden drift means radical changes in the target concept. Single learning model is vulnerable to sudden drift, because single model only observes a particular perspective of malware data distribution.

To handle sudden drift, ensemble learning is needed that hybrid a set of diverse concept descriptions. In this paper, we maintain two much diverse learning models that observe the malware data distribution from both of vertical life-cycle perspective and horizontal traffic similarity perspective simultaneously. The hybrid model is robust to single concept drift attacks. In the future, we are going to introduce more learning models into our system based on p-values against more and more sophisticated concept drift attacks.

The gradual moderate drift induces less radical changes than sudden drift, but the change is more hidden and difficult to be detected. To recognize and react gradual moderate drift, we introduce p-values to enhance fixed, empirical threshold. The p-value gives us the insights of the underlying malware data distribution that is sensitive to gradual moderate drift attacks. SIM and DIFF algorithms can assess the gradual moderate drift affection to each feature, and feature reweighting can update the detection model actively before the cumulative radical drift. In the future, we will design different feature reweighting algorithms for various learning models that could make good use of various model characteristics.

The malware problem is totally different from optical character recognition, speech recognition, bioinformatics and so on, where a trained model could be used for many years with excellent performance. As time goes, the cumulative concept drift of malware will be more and more enormous, that the current concept of underlying malware data could be much different to previous concept. The old malware concept will be noise to current learning model, so that sliding time window is important for malware problem to select malware data relevant to the current concept. The time window moves over recently arrived malware data, and the learnt concepts are only used for detection in the immediate future. The time window size can be fixed or heuristically determined. In the future work, we will introduce sliding time window into our system.

6 Conclusion

For the survival and financial motivation, botnets keep introducing more and more sophisticated evasion techniques, such as concept drift against machine learning detection. To build a sustainable and secure learning model, we need to quickly recognize and react to the concept drift of underlying botnet data distribution. In this paper, we proposed a hybrid botnet detection system based on p-values using both vertical life-cycle algorithm and horizontal traffic similarity algorithm as the underlying scoring classifiers. And average p-value assessment is introduced to recognize gradual concept drift. The feature reweighting could update detection model actively before model performance starts to degenerate.

The hybrid botnet learning system is an open and scale platform which could be built on any other threshold based scoring classifiers. In the future, we will integrate more diverse scoring classifiers into our system to understand the underlying botnet data distribution from more diverse perspectives. And we are going to improve the efficiency of this predictor such as introducing sliding window to online learn the latest concepts and dynamically remove aging data automatically.

Acknowledgements

This material is based upon the work supported by Tianjin Research Program of Application Foundation and Advanced Technology under the Grant No.15JCQNJC41500. This study is also supported by the Soonchunhyang University Research Fund.

References

[1] G. Gu, R. Perdisci, J. Zhang and W. Lee, Botminer: Clustering Analysis of Network Traffic for Protocol- and Structure-independent Botnet Detection, *17th Conference on USENIX Security Symposium*, San Jose, CA, 2008, pp. 139-154.

[2] G. Gu, J. Zhang and W. Lee, Botsniffer: Detecting Botnet Command and Control Channels in Network Traffic, *Annual Network and Distributed System Security Symposium*, San Diego, CA, 2008.

[3] S. Garca, A. Zunino and M. Campo, Survey on Network-Based Botnet Detection Methods, *Security and Communication Networks*, Vol. 7, No. 5, pp. 878-

903, May, 2014.

[4] M. Ektare, *Botnet-as-a-Service is for Sale This Cyber Monday!*, 2016, https://www.zingbox.com/blog/botnet-as-a-service-is-for-sale-this-cyber-monday/

[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas and Y. Zhou, Understanding the Mirai Botnet, *26th USENIX Security Symposium, Vancouver*, Canada, 2017, pp. 1093-1110.

[6] B. Krebs, *Krebsonsecurity Hit with Record DDoS*, 2016, https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

[7] O. Klaba, *Octave Klaba Twitter*, 2016, https://twitter.com/olesovhcom/status/778830571677978624

[8] S. Hilton, *Dyn Analysis Summary of Friday October 21 Attack*, 2016, https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

[9] B. Krebs, *Did the Mirai Botnet Really Take Liberia Offline?*, 2016, https://krebsonsecurity.com/2016/11/did-the-mirai-botnet-really-take-liberia-offline/

[10] AV-Test, *Malware Statistics*, 2017, https://www.av-test.org/en/statistics/malware/

[11] A. Demontis, M. Melis, B. Biggio, D. Maiorca, D. Arp, K. Rieck, I. Corona, G. Giacinto and F. Roli, Yes, Machine Learning Can Be More Secure! A Case Study on Android Malware Detection, *IEEE Transactions on Dependable and Secure Computing*, Vol. PP, No. 99, p. 1, May, 2017.

[12] S. Garca, M. Grill, J. Stiborek and A. Zunino, An Empirical Comparison of Botnet Detection Methods, *Computers & Security*, Vol. 45, pp. 100-123, September, 2014.

[13] Y.-F. Ye, T. Li, D. Adjeroh and S. S. Iyengar, A Survey on Malware Detection Using Data Mining Techniques, *ACM Computer Surveys*, Vol. 50, No. 3, pp. 1-40, June, 2017.

[14] M.-S. Kim, J.-K. Lee, J.-H. Park and J.-H. Kang, Security Challenges in Recent Internet Threats and Enhanced Security Service Model for Future IT Environments, *Journal of Internet Technology*, Vol. 17, No. 5, pp. 947-955, September, 2016.

[15] Lastline, *Protect Your Network From Advanced Malware That Fireeye Doesn't Detect*, 2017, https://www.lastline.com/resource/protect-network-advanced-malware-fireeye-doesnt-detect/

[16] Y.-Y. Zeng, K. G. Shin and X. Hu, Design of SMS Commanded-and-Controlled and P2P-structured Mobile Botnets, *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, New York, NY, 2012, pp. 137-148.

[17] K. Singh, S. Sangal, N. Jain, P. Traynor and W. Lee, Evaluating Bluetooth as A Medium for Botnet Command and Control, *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Bonn, Germany, 2010, pp. 61-80.

[18] K. Krombholz, H. Hobel, M. Huber and E. Weippl, Advanced Social Engineering Attacks. *Journal of Information Security and Applications*, Vol. 22, No. C, pp. 113-122, June, 2015.

[19] T. Yin, Y.-Z. Zhang and S.-H. Li, Dr-Snbot: A Social Network-based Botnet with Strong Destroy-resistance, *IEEE International Conference on Networking, Architecture, and Storage*, Tianjin, China, 2014, pp. 191-199.

[20] E. J. Kartaltepe, J. A. Morales, S.-H. Xu and R. Sandhu, Social Network-based Botnet Command-and-control: Emerging Threats and Countermeasures, *International Conference on Applied Cryptography and Network Security*, Beijing, China, 2010, pp. 511-528.

[21] D.-T. Truong, G. Cheng, A. Jakalan, X.-J. Guo, A.-P. Zhou, Detecting DGA-based Botnet with DNS Traffic Analysis in Monitored Network, *Journal of Internet Technology*, Vol. 17, No. 2, pp. 217-230, March, 2016.

[22] N. Šrndic and P. Laskov, Practical Evasion of A Learning-based Classier: A Case Study, *IEEE Symposium on Security and Privacy*, San Jose, CA, 2014.

[23] B. Biggio, I. Pillai, S. R. Bulò, D. Ariu, M. Pelillo and F. Roli, Is Data Clustering In Adversarial Settings Secure?, *ACM Workshop on Artificial Intelligence and Security*, New York, NY, 2013, pp. 87-98.

[24] B. Biggio, K. Rieck, D. Ariu, C. Wressnegger, I. Corona, G. Giacinto and F. Roli, Poisoning Behavioral Malware Clustering, *ACM Workshop on Artificial Intelligent and Security*, New York, NY, 2014, pp. 27-36.

[25] G. Gu, P. Porras, V. Yegneswaran, M. Fong and W. Lee, Bothunter: Detecting Malware Infection through IDS-driven Dialog Correlation, *16th USENIX Security Symposium on USENIX Security Symposium*, Boston, MA, 2007, Article No. 12.

[26] F. Tegeler, X.-M. Fu, G. Vigna and C. Kruegel, Botfinder: Finding Bots in Network Traffic without Deep Packet Inspection, *International Conference on Emerging Networking Experiments and Technologies*, Nice, France, 2012, pp. 349-360.

[27] I. Arce, The Weakest Link Revisited, *IEEE Security and Privacy*, Vol. 1, No. 2, pp. 72-76, March, 2003.

[28] A. Kantchelian, S. Afroz, L. Huang, A. C. Islam,

B. Miller, M. C. Tschantz, R. Greenstadt, A. D. Joseph and J. D. Tygar, Approaches to Adversarial Drift, *ACM Workshop on Artificial Intelligence and Security*, New York, NY, 2013, pp. 99-110.

[29] K. Singh, A. Srivastava, J. Giffin and W. Lee, Evaluating Emails' Feasibility for Botnet Command and Control, *IEEE Dependable Systems and Networks with FTCS and DCC*, Anchorage, AK, June, 2008, pp. 376-385.

[30] D. Wagner and P. Soto, Mimicry Attacks on Host-based Intrusion Detection Systems, *ACM Conference on Computer and Communications Security*, New York, NY, 2002, pp. 255-264.

[31] C. Smutz and A. Stavrou, Malicious PDF Detection Using Metadata and Structural Features, *ACM Annual Computer Security Applications Conference*, Orlando, FL, 2012, pp. 239-248.

[32] A. Deo, S. K. Dash, G. Suarez-Tangil, V. Vovk and L. Cavallaro, Prescience: Probabilistic Guidance on the Retraining Conundrum for Malware Detection, *ACM Workshop on Artificial Intelligence and Security*, Vienna, Austria, 2016, pp. 71-82.

[33] R. Jordaney, K. Sharad, S. K. Dash, Z. Wang, D. Papini, I. Nouretdinov and L. Cavallaro, Transcend: Detecting Concept Drift in Malware Classification Models, *26th USENIX Security Symposium*, Vancouver, Canada, 2017, pp. 625-642.

[34] L. van der Maaten and G. E. Hinton, Visualizing Data Using t-SNE, *Journal of Machine Learning Research*, Vol. 9, pp. 2579-2605, November, 2008.

[35] Stratosphere IPS, *Stratosphere IPS Project*, https://stratosphereips.org

Biographies

Zhi Wang received the PhD degree in information security from Nankai University in 2012. From 2005 to 2007, he was at the Fortinet Inc.as an Antivirus engineer. From 2013 to 2015, he was at S2Lab ISG RHUL as a post-doc researcher. Currently, he works at Nankai University as a lecturer. His research interests include malware analysis and machine learning.

Meiqi Tian received the BSc degree in information security from Nankai University, China, in 2016. Currently, she is a master candidate at Nankai University. Her research interests include malware analysis and machine learning.

Xiao Zhang is an Android malware researcher in Palo Alto Networks. Before that, he obtained his PhD in Department of Electrical Engineering and Computer Science, Syracuse University. His research mainly focuses on Android system security enhancement and application analysis.

Junnan Wang received the BSc degree in information security from Nankai University, China, in 2016. Currently, she is a master candidate at Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include malware analysis and machine learning.

Zheli Liu received the BSc and MSc degrees in computer science from Jilin University, China, in 2002 and 2005, respectively. He received the PhD degree in computer application from Jilin University in 2009. After a postdoctoral fellowship in Nankai University, he joined the College of Computer and Control Engineering of Nankai University in 2011. Currently, he works at Nankai University as an Associate Professor. His current research interests include applied cryptography and data privacy protection.

Chunfu Jia received the PhD degree from Nankai University, China. Currently, he works at Nankai University as a Professor. His research interests include applied cryptography and software security.

Ilsun You received the MS and PhD degrees in computer science from Dankook University, Seoul, Korea, in 1997 and 2002, respectively. He received the second PhD degree from Kyushu University, Japan, in 2012. From 1997 to 2004, he was at the THINmultimedia Inc., Internet Security Co., Ltd. and Hanjo Engineering Co., Ltd. as a research engineer. Now, he is an associate professor at Department of Information Security Engineering, Soonchunhyang University. His main research interests include internet security, authentication, access control, and formal security analysis. He is a Fellow of the IET and a Senior member of the IEEE.