A First Look at Traffic Classification in Enterprise Networks

Taoufik En-Najjary
Eurecom, France

Guillaume Urvoy-Keller
Eurecom, France

***Abstract*—Enterprise networks have a complexity that sometimes rival the one of the larger Internet. Still, enterprise traffic has received little attention so far from the research community. Most studies rely on port numbers to identify applications.**

In this work, we introduce a method to build statistical classifiers to detect specific intranet applications.

We exemplify the approach with traces collected within the Eurecom network. We demonstrate that our statistical classifiers are able to classify the majority of the flows in our traces. For the cases when the traffic on a specific port cannot be fully identified with our application/protocol decoder, e.g., encrypted traffic, we demonstrate that our approach can be used to test the homogeneity of the traffic, i.e., that the corresponding flows share a common statistical signature that differs from the one of the rest of the traffic.

I. Introduction

Accurate identification of network traffic according to application type is a key concern for most companies, including ISPs. To overcome the limitations of the early solutions based on port numbers, deep packets inspection tools and several statistical classification techniques were proposed [7], [1], [6], [8], [11], [2]. These techniques were heavily tested on Internet traffic.

However, to the best of our knowledge, no work has tackled the problem of enterprise traffic[1] classification. Indeed, nearly all enterprise traffic studies in the literature rely on port numbers to identify applications inside enterprise networks [15].

In this work, we propose an approach to build statistical classifiers (one per application of interest – see Section III-C) for intranet traffic that do not require port numbers to accurately identify specific intranet technique. We rely on a supervised machine learning approach (logistic regression) to build those classifiers. As such, we need to train the classifiers on proper training sets, which are packet traces for which the ground truth, i.e., the application generating the flows in the data set, is known. However, to the best of our knowledge no publicly available deep packet inspection tool embeds signatures for intranet traffic. Commercial tools rather focus on traffic at the boundary of enterprise network, i.e, mostly Internet traffic. Though there is a number of applications that might be used in both types of environment, there are applications and protocols specific to intranets, e.g., NFS or NetBios. In this work, we turn tshark into a DPI tool by leveraging its ability to decode over 1000 protocols/applications (http://www.wireshark.org/docs/dfref/). We use tshark, whenever it was possible (when an appropriate decoder was available), to select flows on popular ports that indeed correspond to the applications that should flow on these ports, e.g., LDAP flows on port 389. This allows us to build training sets for our statistical classifier.

We exemplify our approach on two one-hour long traces collected inside the Eurecom's network, that aggregate the traffic between all the servers, grouped in a specific VLAN, and end user's machines.

We obtain the following results:

- The use of the protocol/application decoding capacity of tshark allows us to validate that traffic behind the vast majority of the most popular ports in our traces corresponds to the legacy applications behind those ports.
- Our statistical classifiers feature high accuracy and precision for most of the ports for which tshark could provide the ground truth.
- For the cases where tshark was not able to decode the protocol, e.g., Eurecom's antivirus application or when it could only provide partial information, e.g., that traffic behind port 636 (LDAPS) was flowing over SSL, we show that our classifier can still be used to test the homogeneity of the corresponding traffic. Homogeneity means that all the flows on the port under study share similar statistical behavior, which significantly differs from the other flows in the data set and suggests that they have been generated by the same application. Knowledge of the actual server behind an IP address further validates the effectiveness of our classifiers.
- We used as classification features an extension of the technique proposed in [1] that considers the size and direction of the first few packets of a connection as a signature of traffic. This technique is considered as a state of the art technique for classifying Internet traffic and, given our results, it seems that approach, originally proposed for Internet application also works for enterprise applications.

II. Problem statement

A flow is defined as a sequence of packets with the same source IP address, destination IP address, source port, and destination port. Our approach, is to build a specific classifier per application. Let A be such an application and let Y be a random variable that takes value one if the flow is generated by application A and 0 otherwise. Let X be the n-dimensional random variable corresponding to the flow features. To each flow a vector x consisting of the n measured features is

[1] By enterprise traffic, we specifically mean the traffic exchanged between hosts and servers within the boundary of the enterprise network, which can extend over the Internet with the use of virtual private networks to connect branches together or to a data center. Note that we use the terms intranet traffic and enterprise traffic interchangeably in this work.

associated. Consider a flow with the following features vector $x = (x_1, x_2, \cdots, x_n)$. We want to estimate the probability that this flow is generated by application A or not. Formally, we can state this as:

$$p(Y = 1|X = x) = P(x, \beta), \tag{1}$$

where $p(Y = 1|X = x)$ is the conditional probability that the flow with features $x = (x_1, x_2, \cdots, x_n)$ is generated by application A and P is a function of x parametrized by the weights vector $\beta = (\beta_0, \beta_1, \cdots, \beta_n)$.

We cast this problem as a logistic regression problem. Logistic regression is designed for dichotomous variables, i.e., to model the relation between a binary variable (true vs. false) and a set of covariates. The use of logistic regression modeling has proliferated during the past decade. From its original use in epidemiological research, the method is now commonly used in many fields including but not nearly limited to biomedical research [16], business and finance [14], criminology [17] and linguistics [10].

III. Learning classifier using Logistic regression

A. Logistic regression model

Within the Logistic regression framework, one assumes a specific function P:

$$P(x, \beta) = \frac{e^{\beta_0 + \sum_{j=1}^{n} \beta_i x_i}}{1 + e^{\beta_0 + \sum_{j=1}^{n} \beta_i x_i}}, \tag{2}$$

From the above equation, we can derive a linear function between the odds of having application A and the features vector x, called the logit model:

$$\log\left(\frac{P(x, \beta)}{1 - P(x, \beta)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n, \tag{3}$$

Unlike the usual linear regression model, there is no random disturbance term in the equation for the logit model. That does not mean that the model is deterministic because there is still room for randomness in the probabilistic relationship between $P(x, \beta)$ and application A.

To implement any logistic regression model, one needs to choose the $\beta_0, \ldots, \beta_n$ values based on a given training set, i.e., a set of flows for which we know whether they have been generated by A or not. We discuss this issue in the next section.

B. Parameter estimation

Assigning the parameters to the logit model boils down to estimating the weights vector β, which is usually done using maximum likelihood estimation.

Consider a training data set of N flows characterized by the features vectors $X = (X_1, X_2, \cdots, X_n)$, where $X_i = (x_1^i, x_2^i, \cdots, x_n^i)$ is the features of flow i, and let the vector $Y = (y_1, y_2, \cdots, y_n)$ be such that $y_i = 1$ if flow i is generated by application A and $y_i = 0$ otherwise. The likelihood function is given by a standard formula (see [4]):

$$\begin{aligned} P(X, \beta) &= \prod_{j=1}^{N} p(Y = y_j | X_j) \qquad (4) \\ &= \prod_{j=1}^{N} (p(Y = 1|X_j)^{y_j} (1 - p(Y = 1|X_j))^{1 - y_j} \end{aligned}$$

As the values of p are small, it is common to maximize the log-likelihood $L(X, \beta) = \log P(X, \beta)$ instead (see [4]), to avoid rounding errors,

$$L(X, \beta) = \sum_{j=1}^{N} [y_j log(p(Y = 1|X_j)) + (1 - y_j) log(1 - p(Y = 1|X_j) \tag{5}$$

By substituting the value of $p(Y = 1|X_j)$ by its value defined in Equation (2) we get the log-likelihood for the logistic regression:

$$L(X, \beta) = \sum_{i=1}^{N} \left[y_i \beta^T X_i - log(1 + e^{\beta^T X_i}) \right] \tag{6}$$

In the logistic regression model, we wish to find β that maximizes Equation (6). Unfortunately, this can not be achieved analytically. In this work, we compute it numerically using the Newton-raphson algorithm [4]. The Newton-Raphson algorithm has been shown to converge remarkably quickly [5]. In this work, it takes less than one second to output an estimate of β.

C. Classification process

Logistic regression falls into the class of supervised machine learning techniques[13]; thus it consists of two main steps. A training step and a classification step.

Training step consist of building a classifier for each application of interst. Consider, for example, the application IMAP. Using Newton-raphson algorithm we estimate a vector β_A that maximize the probability of being IMAP for all IMAP flows and minimize this probability for all non-IMAP flows.

The classification step is done as follows: a given feature vector $x = (x_1, \cdots, x_p)$ is classified as generated by application A if $P(x, \beta_A)$ is larger than a threshold th. A usual choice of the threshold is $th = 0.5$ [5], [4]. By using Equation (3), this boils down to deciding that the new flow x is generated by application A if $\beta_0 + \sum_{i=1}^{n} x_1 \beta_i > 0$.

The choice of $th = 0.5$ is very conservative, as the logistic regression has a strong discrimination power. For example, when considering the IMAP application in the result section, more than 90% of non-IMAP flows have a probability to be IMAP flow less than 0.01, and more than 90% of IMAP flows have a probability of being a IMAP larger than 0.95.

We find the logistic regression technique very convenient as it allows to add a new classifier for each new application we want to analyze. In addition, the computation complexity of logistic regression in the classification phase is very low, as it is linear with the input features.

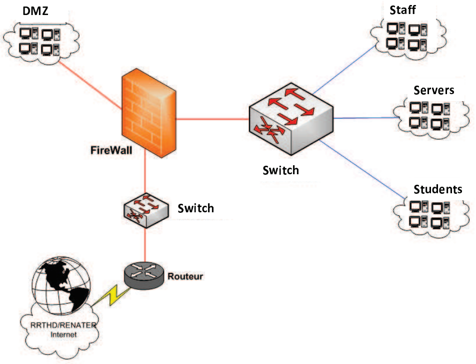

Fig. 1. Architecture of the network

IV. Experiment Setting

A. Data sets

We tested the validity of our classification technique on traffic from our own network. Fig. 1 presents a high level view of our network. This networking infrastructure, which consists of around 800 workstations equipped with a variety of operating systems. The network is organized into several VLANs (servers, staff, DMZ, . . .) connected via a Cisco multi-layer switch. We collected two traces (on October, 28. 2009) of one hour long (one in the morning, between 10 and 11 am and one in the afternoon, between 3 and 4pm) of all traffic flowing between the servers and the end users machines within the Eurecom network. We restrict our attention to TCP flows as they represent more than 97% of flows in each trace, and they carry over 99% of the bytes.

B. Flow Features

Most studies on traffic classification rely on statistics computed once all the packets of a flow have been observed, e.g., duration, number of packets, mean packet size, or inter-arrival time [13]. This clearly prevents any online classification. In contrast, we evaluate the feasibility of application identification in the early stage of a connection. A few works have tackled this challenge. In particular, [1] showed that the size and direction of the k first data packets of each connection, where k is typically in the range of 4 to 5 packets, lead to a good overall classification performance. In this paper we use $k = 4$. We enrich this set with a push flag indicator that indicates whether a data packet has its PUSH flag set or not. Thus, for each data packet we have 3 parameters; direction of the packets (1 for up and 0 for down), PUSH flag indicator (1 if the flag is present and 0 if not) an the size of the packet. We end up having a mix of quantitative (size of data packet) and qualitative (direction, push flag) features. The ability of logistic regression to handle both types of parameters was the main reason that lead us to use it in this paper.

As we are using as features information derived from the first 4 data packets, we de facto exclude all flows with less than 4 data packets as well as the ones for which we did not observe the initial three way handshake. This leave us 59% of the flows that are carrying 99.77% of the bytes in our traces.

C. Performance Metrics

We use the accuracy and precision metrics to assess the quality of our statistical classifier. They are built upon the notion of True Positives (TPs), True Negatives (TNs), False Positives (FPs) and False Negatives (FNs). These notions are defined with respect to a specific class. Let us consider such a specific class, say the IMAP class. TPs (resp. FNs) are the fraction of IMAP flows that are labeled (resp. not labeled) as IMAP by the statistical classifier. FPs (resp. TNs) are the fraction of flows not labeled as IMAP by the ground truth tool that are labeled (resp. not labeled) as IMAP by the statistical classifier.

We use the following metrics to assess the performance of the classification method:

- *Accuracy* , *a.k.a Recall*: Accuracy corresponds to the fraction of flows of a specific class correctly classified. It is the ratio of True Positivesto the sum of True Positives and False Negatives for this class. For example, an accuracy of 50% for the IMAP class means that only half of the IMAP flows are labeled similarly by the statistical classifier.
- *Precision* : For a given class, it is the ratio of True Positives to the sum of True Positives and False Positives. Precision relates to the purity of a class. For example, a precision of 100% for the IMAP given class means that the statistical classifier has put in this class only IMAP flows. This result is satisfactory only if all IMAP flows are actually in this class, which is measured by the accuracy. From a general point of view, a classifier works well if it offers both high accuracy and precision for all classes.

V. Evaluation

We use a supervised classification technique to classify traffic. To avoid a classification error coming from a bad estimation of the statistical model, we limit ourselves to build classifiers for ports hit by more than 250 flows. Using this rule, we end up with 10 ports, whose complete list, along with the legacy application using this port is: 25 (SMTP), 389 (IMAP), 445 (NetBios), 443 (HTTPS), 636 (LDAPS), 993 (IMAPS), 1025 (DCE/RPC), 2799 (unknown application[2]) and 9920 (Antivirus) It might sound puzzling at first that the Antivirus port falls into our definition of intranet traffic, but in practice, the internal hosts connect to an internal server to obtain viral database updates that this server regularly downloads from the servers of the Antivirus company using the Internet connection.

As can be seen from Table I, those 10 ports account for more than 80% of the flows in our traces but they represent only a moderate fraction of bytes, respectively 18 and 56.6%. Most of the bytes are indeed carried on port 2049, the legacy NFS port, as we use NFS v3 at Eurecom that can flow over TCP. The third column of Table I indicates that if we add traffic on

[2] Port 2799 is used between two internal servers connected to the multilayer switch and running database applications

TABLE I
TRAFFIC BEHIND THE 10 MOST POPULAR PORTS

trace	% flows	% bytes	% bytes with 2049	# flows on 2049
Morning	82%	18%	85.6%	138
Afternoon	80%	56.7%	96%	170

port 2049 to the traffic over the 10 most popular port, we end up observing over 85% of the bytes in each trace. However, the number of flows on port 2049 is too low (fourth column of Table I) to permit a reasonable statistical analysis and we leave for future work an in depth study of this port.

We proceeded as follows to analyze the two traces. We first use the protocol/application decoding capabilities of tshark to decode the traffic flowing on the ports we focus on. For each port number, we next build a stistical model on one trace and test it on the second trace. Note that we use the whole traffic (not only the traffic targeting the 10 selected port numbers) in our training data set and test data set. We present results only for the case when training on the morning trace and testing on the afternoon trace, as the reverse case offers highly similar results.

TABLE II
OVERALL TRAFFIC CLASSIFICATION SCORES

Port number	accuracy (%)	precision (%)
25 (SMTP)	95	97.2
143(IMAP)	99.6	99.5
445(NetBios)	91.5	98.7
443(https)	97.5	100
389(LDAP)	91.4	92
636(LDAPS)	95	92.4
993(IMAPS)	99.2	100
1025(DCE/RPC)	75.5.8	93.4
2799	100	100
9920 (Antivirus)	94.3	97.1

A. Ground Truth Establishment

While collecting the traces, we set the snapshot length to 96 bytes, which means that we capture up to 42 bytes of payload (as Ethernet+IP+TCP headers without option sum to 54 bytes). For a given port, we first demultiplexed all the connections targeting this port. We next applied the corresponding tshark decoder and parse the summary file of tshark to check if at least one data segment has been correctly decoded[3]. If this is the case, we consider that this connection corresponds to this application. For the case of IMAPS and LDAPS, we instructed tshark to look for SSL traffic. Also, for port 445, we looked for NetBios and for port 1025 for DCE/RPC. For ports 2799 and 9920, we have no decoder that we can use. We will see in the next section that we can still use our statistical classifier to test the homogeneity of the traffic behind this port.

Applying the above strategy, we find that close to 100% of the connections targeting the ports we consider indeed correspond to the legacy application flowing behind the corresponding port.

B. Classification results

In this section, we first discuss overall results for all internal traffic from and to users machines and also from and to the DMZ[4] (DMZ servers only to internal servers). We next focus on DMZ traffic only.

1) Overall Results: Table II presents the classification scores – accuracy and precision – obtained for each port. We observe very high accuracy and precision for all ports for which tshark could help in building proper training sets. The only exception is port number 1025 corresponding to Microsoft applications using DCE/RPC. While it shows a high precision, its accuracy is low (74.5%). A high precision means that all flow labeled as 1025 are indeed of DCE/RPC type, while a low accuracy means that the classifier is not able to dig up all the flows targeting this particular port number. It is likely that several Microsoft applications communication are using this same port, hence, generating a mixed profile.

For the case of encrypted traffic where tshark was only able to confirm that the traffic flows over SSL, i.e., HTTPS, IMAPS and LDAPS, our classifiers enable to prove that the corresponding traffics are homogeneous, i.e., leave a statistical fingerprint that highly differs from the one of the rest of traffic. As it was possible to verify that the flows indeed targeted the correct servers (HTTPS server, etc.) in the Eurecom network, this further validates the effectiveness of our classifiers for encrypted traffic.

For the case of ports 2799 and 9920, we have no ground truth at our disposal. We can however still apply the same strategy: using the whole traffic behind this port in the first trace and test it on the second trace. The fact that all traffic in the second trace that targeted those ports indeed passed the classification test (high accuracy) and only this traffic (high precision) is a clear indication that the traffic behind those ports is highly homogeneous. Indeed, according to our classifier these flows share the same statistical characteristics in both traces.

2) DMZ: In our network, only four applications are supposed to send data from the DMZ to the internal servers, namely SMTP, IMAP, IMAPS and LDAPS. We trained classifiers for the four corresponding port numbers to see how they behave in this low diversity network. Table III presents the classification scores for each port. These figures show that in this particular case, the logistic regression fully captures the statistical behavior of each application and is able to detect all the traffic. This result suggest to use logistic regression for anomaly detection, i.e., to detect if a flow that does not pass the test is either malicious or malformed (application error).

VI. RELATED WORK

Several studies have recently been performed on corporate networks. For a much more complete survey, see [15].

[3] Note that tshark is not a fully automatic deep packet inspection tool, as its default behavior is to try to decode the legacy application(s) behind a specific port. However, forcing tshark to test all its 1000 decoders to each packet in a trace is highly expensive from a computational viewpoint.

[4] DMZ (demilitarized zone) is a sub-network that contains and exposes an organization's external services to a larger untrusted network, usually the Internet http://en.wikipedia.org/wiki/DMZ_(computing).

TABLE III
CLASSIFICATION SCORES FOR DMZ TRAFFIC

Port number	accuracy (%)	precision (%)
25 (SMTP)	100	100
143(IMAP)	100	100
636(LDAPS)	100	100
993(IMAPS)	100	100

In [9], the authors presented a first work of its kind focusing on the traffic of large enterprise network. They contrast pure intranet traffic to traffic to and from the Internet in terms of volumes and applications and report on some specific phenomena like the existence of failures to establish specific connections internally. They leveraged their knowledge of protocol semantics (and thus go beyond port numbers) to check if failures are widespread among local hosts (it turns out to be the case) or not. However, they did not try to identify the root causes behind those observations.

In [3], the authors look at the health of a typical enterprise network using a metric based on the fraction of useful flows generated by end hosts. Flows considered non-useful are those that explicitly fail or else do not elicit a response from the intended destination. Examining traces collected from 350 mobile hosts at Intel Research, they find that about 34% of the flows are not useful. While high, this figure is not alarming as it does not translate necessarily into users experiencing performance issues.

In [12], the authors tackle the problem of role classification of hosts within enterprise networks. Role classification consists in grouping hosts into related roles so as to obtain a logical view of the network in terms of who is using which resources. Based on communication graphs, the authors proposed to algorithms to uncover logical groups based on roles. For instance, sales hosts as well as and managers of engineers were clustered differently from the other engineers as they were using distinct types of applications hosted by different servers.

VII. CONCLUSION AND FUTURE WORK

In enterprise networks, there is an ever-increasing volume and variety of traffic. In this paper, we propose an approach to build statistical classifiers for intranet traffic. Our starting point is the common belief that applications do not hide themselves in intranets and use legacy ports. We used a well-known protocol/application decoder to validate that this actually holds for our network. We thus obtain a ground truth basis on top of which we can build statistical classifiers to detect key intranet applications that can be used a priori in any enterprise network (note that it does not port number nor payload information). Our approach works both for unencrypted and encrypted traffic. For the cases where it was possible to verify with the protocol/application decoder the application behind a port, we show that a classifier can be built to test the homogeneity of the traffic behind this port.

We consider a number of future extensions to this work. First, further experiments in our network and also on traffic from different networks should be carried out to test the robustness of the method and build classifiers for more applications (e.g., NFS). Second, we considered so far TCP traffic only, however, there can be a significant portion of traffic in intranets that flow over UDP and we intend to extend the approach to handle such traffic.

ACKNOWLEDGMENTS

This work has been supported by the Agence Nationale de la Recherche (ANR) under project ELAN (2008VERS00803).

The authors would like to thank Valérie Loisel and Patrick Petitmengin from Eurecom for their help in capturing the traces and insights about the internal Eurecom network.

REFERENCES

[1] Laurent Bernaille, Renata Teixeira, and Kave Salamatian. Early application identification. In *ACM CoNEXT '06*, pages 1–12, 2006.

[2] Jeffrey Erman, Martin Arlitt, and Anirban Mahanti. Traffic classification using clustering algorithms. In *ACM MineNet '06*, 2006.

[3] Saikat Guha, Jaideep Chandrashekar, Nina Taft, and Konstantina Papagiannaki. How healthy are today's enterprise networks? In *ACM IMC '08*, pages 145–150, 2008.

[4] James W. Hardin and Joseph W. Hilbe. *Generalized Linear Models and Extensions, 2nd Edition*. StataCorp LP, 2007.

[5] David W. Hosmer and Stanley Leshow. *Applied Logistic Regression*. New York ; Chichester ; Brisbane : J. Wiley and Sons, cop., 2001.

[6] Hyunchul Kim, KC Claffy, Marina Fomenkov, Dhiman Barman, Michalis Faloutsos, and KiYoung Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In *ACM CONEXT '08*, pages 1–12, 2008.

[7] Wei Li, Marco Canini, Andrew W. Moore, and Raffaele Bolla. Efficient application identification and the temporal and spatial stability of classification schema. *Computer Networks*, 53(6):790 – 809, 2009.

[8] Andrew W. Moore and Denis Zuev. Internet traffic classification using bayesian analysis techniques. In *ACM SIGMETRICS '05*, pages 50–60, 2005.

[9] Ruoming Pang, Mark Allman, Mike Bennett, Jason Lee, Vern Paxson, and Brian Tierney. A first look at modern enterprise traffic. In *ACM IMC '05*, page 2, 2005.

[10] John C. Paolillo. *Variable Rule Analysis: Using Logistic Regression in Linguistic Models of Variation*. Chicago University Press, 2002.

[11] Marcin Pietrzyk, Jean-Laurent Costeux, Taoufik En-Najjary, and Guillaume Urvoy-Keller. Challenging statistical classification for operational usage : the adsl case. In *ACM IMC '09*, 2009.

[12] Godfrey Tan, Massimiliano Poletto, John Guttag, and Frans Kaashoek. Role classification of hosts within enterprise networks based on connection patterns. In *USENIX Annual Technical Conference*, pages 15–28, 2003.

[13] G Armitage TTT Nguyen. A survey of techniques for internet traffic classification using machine learning. *Communications Surveys and Tutorials, IEEE*, 10(4):56–76, 2008.

[14] Jon Tucker and Dr Jon Tucker. Neural networks versus logistic regression in financial modelling: A methodological comparison. In *in Proceedings of the 1996 World First Online Workshop on Soft Computing (WSC1*, 1996.

[15] Guillaume Urvoy-Keller. Enterprise networks: Related work on traffic analysis based studies (http://elan.eurecom.fr/d1-1.pdf). Technical report, Eurecom, 2009.

[16] Eric Vittinghoff, Davis V. Glidden, and Stephen C. Shiboski. *Regression methods in biostatistics : linear, logistic, survival, and repeated measures models*. Springer New York, 2005.

[17] Jeffery T. Walker and Sean Maddan. *Statistics in criminology and criminal justice, Third Edition*. Sudbury, Mass., USA, Jones and Bartlett Publishers, 2009.