IMRG Workshop on Application Classification and Identification Report

Tim Strayer
BBN Technologies
strayer bbn.com

Mark Allman
ICSI
mallman@icir.org

Grenville Armitage
Swinburne University
garmitage@swin.edu.au

Steve Bellovin
Columbia University
smb@cs.columbia.edu

Shudong Jin
Case Western Reserve University
jins@case.edu

Andrew W. Moore
University of Cambridge
andrew.moore@cl.cam.ac.uk

This article is an editorial note submitted to CCR. It has NOT been peer reviewed. Authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

Categories and Subject Descriptors: C.2.0 General

General Terms: Documentation

Keywords: Application Identification and Characterization, Workshop Report

1. INTRODUCTION

The Internet Research Task Force's (IRTF) Internet Measurement Research Group (IMRG) continued its practice of sponsoring workshops on current topics in computer network measurement with the Workshop on Application Classification and Identification (WACI) held October 3, 2007, at BBN Technologies in Cambridge, Massachusetts. There has been much general interest within the community recently in finding techniques to identify traffic without examination of the service ports used in the TCP or UDP header, as these increasingly do not accurately indicate the application generating the traffic. The workshop agenda was formed around six abstracts that were selected from an open call by the workshop organizing committee.[1]

2. KEYNOTE

Andrew W. Moore (Cambridge University Computer Laboratory) gave the keynote talk. He asked why we want to identify the applications generating the traffic. The answer: because we are interested in what the traffic is doing. Using port numbers alone to is not sufficient for identifying the applications; Andrew noted that 30% of the traffic in a recent sample used ports not listed in the IANA services database, and of the 70% that did use official ports, 29% of that traffic was incorrectly identified by the port. What is an application, anyway? How do we classify web-mail through a web browser? It is a poor but prevalent practice to use port numbers as the ground truth for the traffic mix of a sample. Yet we want to make routers smarter so they can react to applications better, and we want to protect our networks against malicious or badly acting applications. Knowledge of the mix of application traffic also gives insight into the conversations and relationships between servers and clients. The key points of the remainder of the talk are:

- Until a full flow is "decoded"—that is, the content of the flow is examined and the intent of the flow is ascertained—it is difficult to know what is really going on in the flow. The content is what really identifies an application, not the mechanisms for getting the content to and from the endpoints. Andrew showed that very good accuracy is achieved only by deeply inspecting the traffic, but that this method is time consuming and computationally expensive. Further, there will always be errors from encrypted payloads, covert channels, and samples that are too small to decode.

- Probabilistic learning approaches also show promise. Here, models are built by training on known traffic. Andrew reported accuracy of 65% for Naive Bayes, and greater than 90% for more sophisticated learning algorithms. The key to these approaches, of course, is the feature set used to form the rules for placing flows in categories. Naturally, the server port number is a particularly good attribute, but it must be augmented by many other features, such as the number of pushed packets, the initial window sizes, the average segment size, and roundtrip times. Nonetheless, we are still fairly naive in our approach to using learning models.

- Netflow data offers some interesting opportunities. It is a common and hugely available data source, and the data itself is often held for long-term archive. Netflow does suffer from sampling, but it may reveal some useful flow structures and may be useful as a fingerprint mechanism.

- There are many motivations for why we classify applications, including attack identification, network understanding, accounting for use, enabling dynamic handling for specific applications, tracing performance, and building better models for traffic generation in testbeds. We cannot compare results from papers if there is no common method for extracting the application classes, even when the papers start with the same data.

- It is important, however, to mention the law and the need for privacy, and their chilling effect on access to

[1]The committee forms the authorship of this report. The lead author chaired the committee.

data. Certain laws are forcing an internal awareness at ISPs, and upper management is willing to fund the placement of monitors in networks, but this leads to privacy issues and potential abuse of the collected information.

3. TALKS

After the keynote a variety of talks were given that inevitably gave way to group discussions. These presentations and discussions are sketched in the following subsections.

3.1 Treatment-Based Traffic Signatures

Mark Claypool (Worcester Polytechnic Institute) presented observations aimed at the home network [2]. With the proliferation of devices on the home network, there is a need to be able to distinguish between applications within the network to provide better overall service. One way to classify applications is by delay sensitivity, loss, and jitter, but that requires external effort. Most people just want to plug in devices, so the infrastructure needs to figure out (classify) the applications (QoS treatment). While looking at the nature of the traffic, we don't really need to distinguish down to application. Using ports alone does not provide the accuracy, and examining the payload is difficult. In the context of the home, users aren't trying to hide; the real challenge is that the users want traffic distinction for QoS treatment done in real-time. Mark offered a 3-D cube to describe the application behavior: transmission spacing, packet size tendency, and nature of reverse traffic. Mark also observed that application behavior may change mid-flow and thus may need different treatments at different times.

3.2 Spotting Spam in Sampled sFlow

Richard Clayton (University of Cambridge) presented work on spotting spam in sampled sFlow data [3]. The challenge is to spot outgoing spam at a smart host that processes email logs. Looking at the email delivery failures tells a lot about spam. Further, spam can be found by telltale traffic patterns, since spam does not look like normal mail. In this case, using the well known port to classify the email is actually a good discriminator, since email overwhelmingly uses port 25. The methodology used in this study consisted of observing excessive variations in the amount of time particular hosts were active as captured in sFlow data.

3.3 Comparison of Internet Traffic Classification Tools

Hyunchul Kim (CAIDA) presented a comparison of Internet traffic classification tools [5]. The methodology used in this study involves applying several classification techniques to a number of traces. The classification schemes tested were: CoralReef, based on port numbers; BLINC, based on host behavior; and six machine learning algorithms using the WEKA tool. CoralReef identified several applications nicely, but did poorly at identifying applications where the port number is not standard. BLINC did not perform well for applications with strong port correlation. BLINC was additionally found difficult to tune (28 parameters) and the link characteristics caused trouble. Of the six machine learning algorithms tested, the Support Vector Machine algorithm did well with smallest size of training set. A high-order message from this talk is that ports are still the major discriminator for classification algorithms.

3.4 Appmon: An Application for Accurate per Application IP Traffic Classification

Demetres Antoniades (Foundation for Research and Technology, Hellas, Crete) described Appmon, a tool for accurate per-application network traffic characterization [1]. He observed that the "elephants" are getting bigger. That is, fewer protocols are carrying a larger fraction of the traffic, exacerbating the need for good traffic characterization tools. Appmon uses packet signatures and deep packet inspection in order to characterize and visualize the application traffic in real time. There are three "trackers" that filter the traffic; what survives one moves to another. He showed test results that suggest that Appmon is able to monitor traffic at gigabit speeds and exhibited good performance in a real production environment.

3.5 The Problem of Large-Scale Application Traffic Characterization—Can We Do Better?

Kostas Anagnostakis (Software Systems Security) described ways to help automate the process of traffic characterization [4]. The work is inspired by zero-day worm fingerprinting using a rolling hash over a flow's content, specifically, a rolling histogram of packet size over time. He also observes that a type of application found at a particular host will likely be employed at that host again in the future. This helps with accuracy.

3.6 Identifying Rogue/Nefarious Applications

David Lapsley (BBN Technologies) presented a system for aggregating monitored traffic looking for evidence of botnets so that botnet-based attacks can be attributed to a particular botnet controller [6]. The approach employs a set of increasingly more complex analyzers, filtering out unlikely flows at each step and leaving the most computationally intensive analysis is done on a reduced traffic set. Next, a machine learning classification technique was applied to the remaining traffic. The final step performs a correlation algorithm looking for groups of flows that may be related by being part of the same botnet. A set of manufactured botnet traces were inserted into a public dataset. The system was able to identify the flows involved in the botnet cluster.

4. ROUND-TABLE DISCUSSION

After the six presentations, the workshop participants engaged in an open round-table discussion about the challenges and promise of traffic classification. The following is a sketch of some of the points made during the round-table.

- This conjecture was made: A peer-to-peer network will always look like a P2P network. One can make a P2P network from a variety of mechanisms, including gmail, but it still looks like a P2P network, not mail. The point is that there are variants, but also fundamental properties. More complicated applications take longer to train on.

- What is the use of the classification? The intent of the classification speaks to how accurate we need to be.

- Are we attempting to classify traffic for good or evil? One side says we should know everything and then let policy dictate what we do with that knowledge.

And what about evasion? That is, what do we do when mechanisms are specifically put into place to circumvent application identification and traffic classification? Sometimes evasion is done for evil, but sometimes evasion is not adversarial, and is a natural evolution. Is "good" evasion at cross-purposes with "good" application identification? And what about network neutrality?

- It was stated that someone should stand up a good dataset with ground truth in it as the common dataset, perhaps by instrumenting a group of users to establish ground truth. Another point is that we should be sharing tools, not data, and that the tools should be written with an eye towards release. It was observed that in hard science, such as biology, practitioners develop detailed protocols to aid in replication of results, but publishing reproduced results is not easy within computer science.

- Where does deep packet inspection (DPI) fit? Most data sets provide only headers, and full packet traces are typically rare and small. Most people agreed that DPI had privacy issues but was fine "for the health of the network." There are a variety of laws attempting to codify the privacy and public safety aspects of DPI. We cannot talk about DPI or classification in general without also talking about legality. DPI also has performance issues, and is obviously useless when the data is encrypted.

- A couple of open question were raised: How do you detect an application you have never seen before? What about protocols? Is IP protocol 6 always TCP?

- One suggestion is to have an IMRG document on classes of applications, and have a registry of application classes. It may be easy to get such a document to be about 95% inclusive, but getting the remaining amount will be hard. Also, the "95%" will depend on the metric we are using to count applications.

5. WORKSHOP MATERIALS

The slides and abstracts from the talks are available from the workshop web page:
http://www.icir.org/imrg/waci07/.

Attendees

Mark Allman (ICSI), Kostas Anagnostakis (Software Systems Security), Demetres Antoniades (Foundation for Research & Technology), Mark Claypool (Worcester Polytechnic Institute), Richard Clayton (Cambridge University - Computer Laboratory), Alberto Dainotti (CAIDA), Lars Eggert (Nokia), Ashley Flavel (University of Adelaide), Manish Karir (MERIT Network), Hyunchul Kim (CAIDA), Christian Kreibich (ICSI), David Lapsley (BBN Technologies), Dinesh Makhija (ELLACOYA), Allison Mankin (National Science Foundation), Andrew W. Moore (Cambridge University - Computer Laboratory), Rahul Patel (Cisco), Shudong Jin (Case Western Reserve University), Tim Strayer (BBN Technologies), Anthony Vardaro (University of Massachusetts, Lowell), Craig Wills (Worcester Polytechnic Institute), Charles Wright (John Hopkins University), and Peiter "Mudge" Zatko (BBN Technologies).

Acknowledgments

The IMRG is grateful for the hard work put in by the organizing committee. In addition, we thank the speakers and attendees for an interesting day of talks and discussion. Finally, we are indebted to BBN Technologies for sponsoring and hosting the workshop and in particular to Janet LeBlond and Tim Strayer for handling the workshop logistics.

6. REFERENCES

[1] Demetres Antoniades, Michalis Polychronakis, and Evangelos P. Markatos. Appmon: An Application for Accurate per Application IP Traffic Classification.

[2] Mark Claypool, Robert Kinicki, and Craig Wills. Treatment-Based Traffic Signatures.

[3] Richard Clayton. Spotting Spam in Sampled sFlow.

[4] Kirk Jon Khu, Periklis Akritidis, Angelos Stavrou, and Kostas Anagnostakis. The Problem of Large-Scale Application Traffic Characterization—Can We Do Better?

[5] Hyunchul Kim, Marina Fomenkov, kc claffy, Nevil Brownlee, Dhiman Barman, and Michalis Faloutsos. Comparison of Internet Traffic Classification Tools.

[6] David Lapsley, Robert Walsh, and W. Timothy Strayer. Identifying Rogue/Nefarious Applications.