EURECOM
Department of Networking and security
2229, route des Crêtes
B.P. 193
06904 Sophia-Antipolis
FRANCE

Research Report RR-11-247

Toward systematic methods comparison in traffic classification

January 15th, 2011
Last update January 15th, 2011

Marcin Pietrzyk, Lucjan Janowski , and Guillaume Urvoy-Keller

Tel : (+33) 4 93 00 81 00
Fax : (+33) 4 93 00 82 00
Email : {pietrzyk@eurecom.fr}
{janowski@kt.agh.edu.pl}
{guillaume.urvoy-keller@unice.fr}

[1] EURECOM's research is partially supported by its industrial members: BMW Group, Cisco, Monaco Telecom, Orange, SAP, SFR, Sharp, STEricsson, Swisscom, Symantec, Thales.

Toward systematic methods comparison in traffic classification

Marcin Pietrzyk, Lucjan Janowski , and Guillaume Urvoy-Keller

Abstract

A host of methods and algorithms have been proposed to solve the issue of traffic classification in the recent years. However, results comparison between different works is very hard due to the lack of structure and common understanding of notions in the domain, especially a precise definition of application classes. In this work we aim to fill this gap and propose a first attempt to systematic classes traffic definitions. To fulfill this goal, we take advantage of the ontology paradigm.

Index Terms

traffic classification, ontology

Contents

List of Figures

1 Introduction

There has been a host of methods and algorithms proposed to solve the issue of traffic classification in the recent years [1–6]. So many different algorithms demand a formalism which makes it possible to compare them. However, comparing the relative merits of different classification techniques is hard due to the lack of structure and common understanding of notions, e.g., traffic class. For instance, the traffic category called in two works "WEB", can carry fundamentally different types of flows depending on the network which is monitored (for instance HTTP Streaming, HTTP file download or standard browsing). As a consequence one can have many ambiguities that often prevent a direct comparison of methods.

In this work we aim at formalizing the definitions of traffic classes. For this purpose, we use the ontology paradigm, which defines an explicit formal specification of the terms in the domain and their relations [5]. Using classical guidelines on how developed an ontology, we iteratively build a consistent and structured application categorization that helps removing ambiguities in the definitions of traffic classes. One of our objectives is to foster cooperaration within the traffic classification community so as to further develop our initial ontology, by basically populating it with new applications. To this end, we created a wiki page of the project `http://www.pluton.kt.agh.edu.pl/~ljanowski`.

The remainder of this paper is organized as follows. We report on our assumptions and high level strategy in Section 3. We describre the resulting categorization tree in Section 4. Examples of the benefits of adopting our approach are provided in Section 6. Section 7 concludes the paper.

2 Related work

Ontology has been applied to a number of domains. Hereafter, we cite several examples of relevant works that describe ontologies, or methods to develop ontologies. Several ontologies have been proposed for web annotations [7]. Authors in [8] propose ontology for traffic classification. However, their proposal is at high abstraction level compared to our approach and does not address the issue of possible ambiguities of traffic classes definition.

The specific example of traffic classification i.e, malware traffic classification uses ontology and numerous of different examples of such ontologies can be found in literature [9–11]. The malwere traffic ontologies can be general [10] or very specific for example [11] presents DDoS (Distributed Denial of Service) ontology only. Nevertheless, the malwere traffic is so specific that we cannot use such ontology to classify legitimate traffic. In our ontology we created malwere class which should be extended by a malwere traffic specialist.

General problem of creating an ontology is well know. Authors in [12] present overview of knowledge sharing and problem-solving methods including ontologies. To develop our methodology, we follow the methodology proposed in [7].

3 Ontology

In this section we briefly describe what is ontology and what are the advantages of using it in our scenario. We further describe our ontology assumptions and its development procedure. The resulting categorization will be described in Section 4.

Ontology is defined as an explicit formal specification of the terms in a domain and their relations [7]. An ontology defines a common vocabulary for researchers who share information in a domain. The reasons why ontology development is useful are many fold: (i) To share common understanding of the structure of information between people working in the domain, (ii) To make domain assumptions explicit (iii) To enable easy comparison and reuse of domain knowledge.

3.1 Our assumptions

Our starting point is an existing application classification tool, which is currently used by a major European ISP [13]. This tool belongs to the family of deep packet inspection (DPI) tools, which means that it seeks application level signatures within users' payload to detect applications. A typical example of classes used by our DPI, along with examples of applications for each class, are presented in Table 2.

To help us building an ontology, we consider the following questions that might help us organizing classes. We consider either general purpose questions or network specific applications. Table 1 presents the answer to each question for the set of classes presented in Table 2.

Category	Purpose	Interac.	Action	Content	Dist.	Style	Band	Hiding	L4	Protocol	Encrypted
[WEB BR]	Con. ex.	H	A	TXT/IMG/Flash	Cent.	H2M	M	N	TCP	Open	Both
[P2P]	Con. ex.	L	L	ANY	Decent.	M2M	H	Y	TCP,UDP	Both	Both
[DOWN.]	Con. ex.	L	L	ANY	Cent.	H2M	H	N	TCP,UDP	Open	Both
[STR.]	Con. ex.	H	L	A/V	Cent./Decent.	M2M	H	Y/N	TCP,UDP	Both	Plain
[MAIL]	Comm.	M	M	ANY/TXT	Cent.	H2H	L	N	TCP	Open	Both
[CHAT]	Comm.	H	H	TXT/V	Cent./ p-p	H2H	L	N	TCP,UDP	Both	Both
[VOIP]	Comm.	H	H	VOICE	Cent./ p-p	H2H	L	Y	TCP,UDP	Both	Both
[GAMES]	Other	H	H	*	*	H2H	*	*	*	*	
[CONT.]	Other	*	*	*	*	*	*	*	*	*	
[DB]	Other	*	*	*	*	*	*	*	*	*	
[OTHERS]	Other	*	*	*	*	M2M	*	*	*	*	

Table 1: Operational traffic categories, and general characteristics.

General questions:

1. What is the primary purpose of the application category (from the user's perspective)? We consider four options {Content Exchange, Communication, Other, Malware}

2. What is the level of interactivity required from the user? {High, Medium, Low}

3. What is the level of presence required from the user? {High, Medium, Low}

4. What is the dominant type of content exchanged by the application? {Audio, Video, Text, Voice, Any}
5. What is the distribution technique? {Centralized, Distributed, Point to Point}
6. Who is communicating with whom? Machine to Machine, Machine to Human, Human to Human {M2M, M2H, H2H}
7. Bandwidth consumption (per user)? {Low, Medium, High}
8. Does application try to ”hide” for some reasons {Y,N}

Network oriented questions:

1. Transport layer protocol? {TCP, UDP, None}
2. Protocol type {Proprietary, Open}
3. Encrypted traffic {Yes, No, BBoth}

Category	Example
[WEB BROWSING]	Website browsing
[P2P]	EDONKEY, BITTORRENT, GNUTELLA
[DOWNLOAD]	One click hosting, e.g. rapidshare [14]
[STREAMING]	Youtube [15], Sopcast, Windows Media
[MAIL]	SMTP, POP2, POP3, IMAP, WEBMAIL
[CHAT]	MSN, ICQ, IRC, Gtalk
[VOIP]	Skype, SIP, H.323, Gtalk
[GAMES]	Quake, HTTP Games
[CONTROL]	Telnet, SSH, VNC
[DB]	LDAP, MSSQL, ORACLE
[OTHERS]	ICMP, ROUTING

Table 2: Operational traffic categories (based on internal DPI tool), along with examples.

3.2 The special case of HTTP traffic

In the recent years we observe a come back of the HTTP traffic, which is once again taking over the lead in terms of traffic generated in the residential networks at the expense of P2P [13, 16]. Indeed, a growing variety of applications either migrate to web based clients or at least have a web based equivalent. Even peer-to-peer networks experience growing competition from HTTP download services [14]. Thus, in our classification HTTP traffic is broken into several classes depending on the application implemented on top: Webmail will be categorized as mail, HTTP streaming as streaming, HTTP file transfers as HTTP DOWNLOADS etc.

4 Ontology for Traffic Classification - Application Part

The simplest kind of ontology is a decision tree where each branch is a particular answer to a question. In this work, we favored the use of such a type of ontology for two reasons: (i) it appears to be rich enough to describe all the applications we know of[1], (ii) it is simple enough to enable an easy adoption by the researchers in the traffic classification domain.

During the iterative process of the ontology development, we kept only the most meaningful questions/answers among the ones listed in Section 3.1. For instance, networking oriented question turned out to be too specific to be discriminative, as, for example, many applications can use both TCP and UDP.

In general most of the recent applications can not only use different connections but they can also be used to significantly different tasks e.g. VoIP and file transfer can be run using Skype application. The consequence is that much more applications should be classified (similarly to HTTP traffic described in Section 3.2) to different classes. We decided to categorize applications just to one class which is chosen based on their dominating use from a user perspective. For example, Skype, will be classified as an application for voice/video communication, even though it can be used to perform file transfers.

Since the user is a key entity for any network operator, we decided to create a user driven ontology. This is why the first feature differentiating classes considered in our ontology is question 1 : "What is the primary purpose of the application?" Of course such a question can have a lot of different answers corresponding to different levels of granularity. Since this is the first question of our ontology the answers which should be general enough. We thus created two main classes: CONTENT_EXCHANGE and COMMUNICATION. Note that those two classes describe two fundamentally different user's behaviors. If he/she uses an application to communicate, he/she has to sit in front of her computer. On the other hand, he/she is not necessarily present in front of his/her computer when a download is in progress. Also if the user is communicating with another user, the traffic should be relatively symmetric. This is in contrast with the case of CONTENT_EXCHANGE where one expects the traffic to be (in most cases) asymmetric. As always in the case of networking applications, we can find special cases that do not fall in one of the two sub-classes (CONTENT_EXCHANGE and COMMUNICATION). For instance, the e-mail application is used to communicate but generates traffic even if it is not interactively used. Accounting for such particularities would lead to a much more complex ontology. The created ontology is a trade between completeness and usability.

The question we address at the second level of the ontology is: "With whom you are communicating?". In case of CONTENT_EXCHANGE such question

[1] Applications featuring similar characteristics are in general clustered close to each other in the ontology.

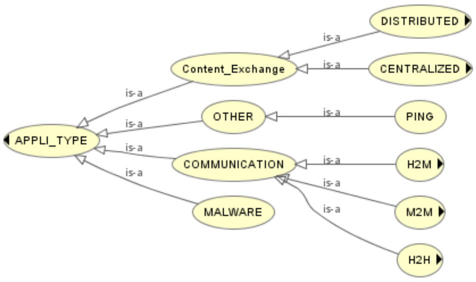

Figure 1: The root of application ontology.

changes to question 5, i.e., "Is content distributed in a centralized or distributed manner?" It gives us two classes: CENTRALIZED and DISTRIBUTED.

In the case of COMMUNICATION, the question "With whom you are communicating?" can be answered directly (see question 6). Communication means interaction but the remote party can be of different nature. Therefore, the main classes are human to human (H2H), human to machine (H2M) and machine to machine (M2M) communications.

The First two layers of the ontology are shown in Fig. 1. There are two more classes not described in details. The first one is MALWARE and the second is OTHER. MALWARE is not analyzed in details since it is out of scope of this work. What is more there exist already works that addressed this particular issue [17,18]. The OTHER class is used to aggregate all traffic not fitting to any other category, e.g., ICMP traffic.

Fig. 1 presents ontology classes and the relations between them.A small arrow attached to a class name means that some subclasses of this class are defined but not shown in this view. In the next sections, similar plots are used to show other parts of the ontology.

4.1 CONTENT_EXCHANGE Classes

Two CONTENT_EXCHANGE subclasses, DISTRIBUTED and CENTRALIZED, have been introduced to describe the way content is exchanged, even though it might be the same type of content. Therefore, the next question is the same for both of these subclasses: "What is the content type?". Similarly to the previous questions, we focus on the user. Therefore, the answers are limited to live streaming content and other content. The main reason behind this choice is that the user's behavior is strongly different for live streaming than for other type of content, i.e, picture or text or even progressive download. For the latter types of content, the user has to wait until the content is fully downloaded before watching or hearing it.

The next level relates to specific applications, i.e., there are no more questions but we are adding applications. As we are not able to exhaustively list all the applications in this part of the tree, we created special collectors like GENERIC_TRANSFER_CLIENT,

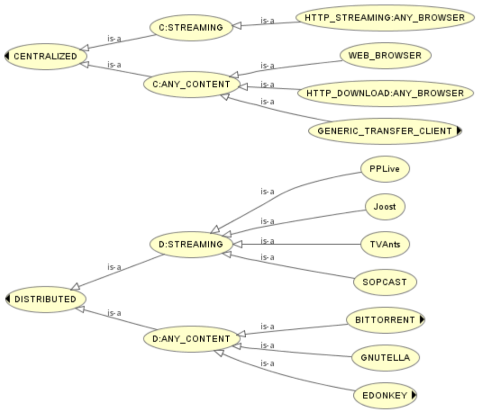

Figure 2: The CONTENT_EXCHANGE part of the ontology.

which aggregates different applications. This part of the ontology, which lies below the CONTENT_EXCHANGE class, is presented in Fig. 2.

4.2 COMMUNICATION Classes

Similarly to the two CONTENT_EXCHANGE subclasses, DISTRIBUTED and CENTRALIZED, the subclasses of the COMMUNICATION class – H2M, M2M and H2H – are further divided based on the content which is exchanged (question 4).

Note that not all content types are possible, dependending on who is communicating with whom. Once content is specified, applications are listed similarly to the CONTENT_EXCHANGE case. Similarly also, some collectors are attached to some specific applications whose complexity requires further details to be fully characterized from a traffic classification viewpoint.

In the case of H2M, we did not specify any content type but directly application collectors since in this case, the application determines what the user is actually doing.

In the case of M2M, we introduced the ROUTING class, which constitutes a key subclass of M2M applications.

The last class, H2H, is the most interesting one in our opinion as it is the richest in terms of variety of content and applications. We have introduced four different subclasses – CHAT, GAMES, VOIP and MAIL – which are further divided into different applications. Note that for most of the classes, we have the ANY_BROWSER subclass, to account for the fact that a lot of applications are now implemented over HTTP. GAMES could be further divided into FIRST_PERSON and MMORPG. Nevertheless, since we are not experts in this domain, we prefer to leave refinement of this part of the ontology to the researchers in this field.

The COMMUNICATION part of the ontology is presented in Fig. 3.

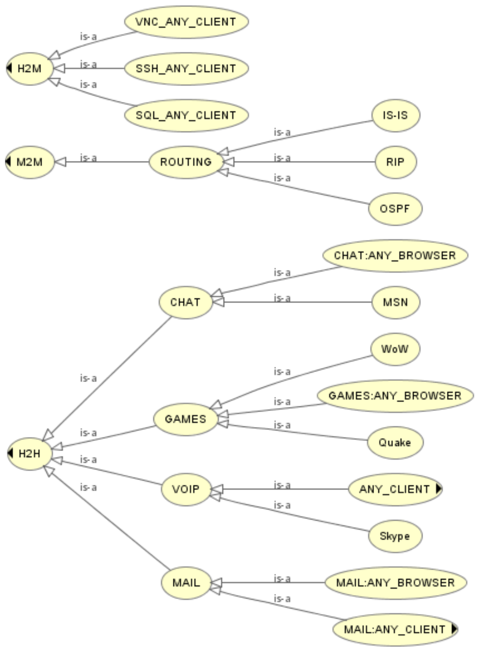

Figure 3: The COMMUNICATION part of the ontology.

5 Ontology Flowchart

The ontology presented in the previous section makes it possible to classify any application we are aware of. Nevertheless, we are interested in classifying flows and not applications since flows are the entities actually sent on the Internet. Indeed, the traffic classification task most of the time consists in associating an observed flow to an application. To better understand why we put the emphasis on flows here, let us consider the case of eMule. eMule is used to exchange content (files) but also to perform communication between users and machine so as to locate content – one of the functions of the p2p overlay of eMule. Therefore, subclasses representing those different types of flows generated by eMule should be created in the ontology

Dividing applications into different classes of flows is a difficult task since a per application detailed analysis has to be done. Therefore, we are not trying to add such subclasses for all applications we know. Instead we exemplify this task with the case of EDONKEY.

The flow ontology created for EDONKEY class is presented in Fig 4. Beware that we use the wording class both in a traffic classification context and in a ontology context. The latter refers to the nodes in the tree we build, while the former refers to a set of similar applications – see Table 2. An ontology enables to fully specify a “traffic classification class” by mapping it a set of leaves in the ontology, as we exemplify here for the EDONKEY class.

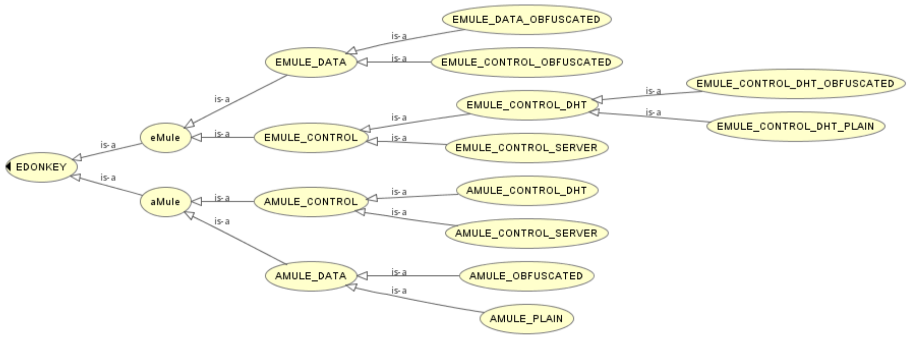

Figure 4: The flow ontology created for EDONKEY application (class).

Flows generated by the applications in the EDONKEY class cannot be divided into (ontology) subclasses by answering the same questions because those applications feature different behaviors. We first introduce subclasses that represent linux (aMule) and windows (eMule) clients. Note that for an other application, such an information might be irrelevant. Although, both eMule and aMule clients connect to the same network (eDonkley) some of the protocol properties actually differ, which can affect the classification method required for each application.

The next step is to distinguish between the different types of flows each version of eDonkey generates, i.e., data and control flows[2].

The following subclasses are strongly application driven and make it possible to specify very detailed application behaviors. Note that the most important advantage of such a precise flow classification is that we can specify which flow is really detected if a traffic classification technique describes its classes thanks to our ontology. For example if an algorithm can detect EMULE_DATA_PLAIN but not EMULE_DATA_OBFUSCATED, we can expect that the unknown traffic left by the specific classification technique is partly generated by encrypted eMule flows. Moreover, if one claims that an algorithm can classify the EMULE class, it means that the algorithm can classify EMULE_CONTROL_DHT as well. Since detecting the control traffic can be considered as crucial[3], it should be made clear if the algorithm detects it correctly or if alternatively, it focuses only on the actual content transfers. Our ontology should help uncovering such characteristics and more generally what is actually detected by a specific algorithm.

6 Ontology usage examples

Let us now demonstrate how the ontology we propose would look like in the case of a real traffic classification technique. In Table 3, we present comparison

[2] Note that any subclass of particular version of an application has name starting with the name of the superior class. Therefore, data flows of eMule and aMula are called EMULE_DATA and AMULE_DATA respectively.

[3] For instance blocking only signaling layer is sufficient to prevent application usage.

Class	Ontological name
WEB	CONTENT_EXCHANGE/ANY_CONTENT/CENTRALIZED/ANY_BROWSER/HTTP_BROWSING
HTTP-STR	CONTENT_EXCHANGE/VIDEO_LIVE/VIDEO_CENTRALIZED/ANY_BROWSER/HTTP_STREAMING
EDONKEY	CONTENT_EXCHANGE/ANY_CONTENT/DISTRIBUTED/EDONKEY/*/*_DATA/*
BITTORRENT	CONTENT_EXCHANGE/ANY_CONTENT/DISTRIBUTED/BITTORRENT/*/BITTORRENT_DATA/BITTORRENT_DATA_PLAIN
MAIL	COMMUNICATION/MAIL/*
UNKNOWN	Other not recognized by ODT

Table 3: Ontology vs. standard approach

of the "traffic classification classes" used in our previous study [13] (left column) versus the ontological approach (right column).

The EDONKEY class is an illustrative example of the benefit of using a standardized naming convention. The traffic generated by EDONKEY clients is heterogeneous in nature, as we have control plain (DHT based or centralized) and data plain traffic, which can further be obfuscated or consists of plain transfers. Fig. 4 presents part of our ontology, describing the EDONKEY traffic. The detection method required and its difficulty can change dramatically based on which of the four types of EDONKEY traffic we try to classify. Most of the studies (including [13]) target the detection of EDONKEY data transfers only. However, this information is often missing or hidden. Using our ontology makes such information explicit, a first and mandatory step toward understanding the merits of different traffic classification studies.

Many application classes and types exhibit heterogeneous behavior in terms of traffic they generate. Another illustrative example can be Skype or the legacy FTP protocol, which use distinct channels for control and data traffic. The use of ontology facilitates comparison of different studies and methods and makes domain assumptions clear and precise.

7 Discussion and Conclusions

In this work we addressed the problem of the class definition ambiguity that concerns most of the works in the domain of traffic classification. We proposed a first classification categorization tree, based on the ontology paradigm. The categorization is generic enough to be extended by researchers working on the classification of a particular group of applications. We exemplified how the ontology should be developed with the example of the EDONKEY traffic class, and highlighted the advantages of our approach by contrasting it with standard application definition used in scientific papers. To encourage collaborative efforts to extend our work, we share our ontology definition in a open source format that can be easily accessed and extended.

References

[1] G. A. TTT Nguyen, “A survey of techniques for internet traffic classification using machine learning,” *Communications Surveys and Tutorials, IEEE*, vol. 10, no. 4, pp. 56–76, 2008.

[2] L. Bernaille, R. Teixeira, and K. Salamatian, “Early application identification,” in *CoNEXT ’06: Proceedings of the 2006 ACM CoNEXT conference*. New York, NY, USA: ACM, 2006, pp. 1–12.

[3] W. Li, M. Canini, A. W. Moore, and R. Bolla, “Efficient application identification and the temporal and spatial stability of classification schema,” *Computer Networks*, vol. 53, no. 6, pp. 790 – 809, 2009.

[4] M. Pietrzyk, G. Urvoy-Keller, and J.-L. Costeux, “Revealing the unknown adsl traffic using statistical methods,” in *COST 2009 : Springer : Lecture Notes in Computer Science, Vol 5537, 2009.*, May 2009.

[5] T. Karagiannis, A. Broido, N. Brownlee, K. C. Claffy, and M. Faloutsos, “Is p2p dying or just hiding?” in *Proceedings of the GLOBECOM 2004 Conference*. IEEE Computer Society Press, November 2004.

[6] I. Trestian, S. Ranjan, A. Kuzmanovi, and A. Nucci, “Unconstrained endpoint profiling (googling the internet),” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 279–290, 2008.

[7] N. F. Noy and D. L. McGuinness, “Ontology development 101: A guide to creating your first ontology,” Online, 2001. [Online]. Available: http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html

[8] M.-M. Naing, E.-P. Lim, and D. G. Hoe-Lian, “Ontology-based web annotation framework for hyperlink structures,” *Web Information Systems Engineering Workshops, International Conference on*, vol. 0, p. 184, 2002.

[9] R. Bisbey and D. Hollingsworth, “Protection analysis project - final report,” USC/Information Sciences Institute, Tech. Rep., 1978.

[10] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, “A taxonomy of computer program security flaws,” *ACM Comput. Surv.*, vol. 26, no. 3, 1994.

[11] J. Mirkovic and P. Reiher, “A taxonomy of ddos attack and ddos defense mechanisms,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 39–53, April 2004.

[12] A. G. Perez and V. R. Benjamins, “Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods,” in *In*, 1999, pp. 1–1.

[13] M. Pietrzyk, J.-L. Costeux, G. Urvoy-Keller, and T. En-Najjary, “Challenging statistical classification for operational usage: the adsl case,” in *IMC ’09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. New York, NY, USA: ACM, 2009, pp. 122–135.

[14] D. Antoniades, E. P. Markatos, and C. Dovrolis, “One-click hosting services: a file-sharing hideout,” in *Internet Measurement Conference*, 2009, pp. 223–234.

[15] Youtube, “http://youtube.com/.”

[16] G. Maier, A. Feldmann, V. Paxson, and M. Allman, “On dominant characteristics of residential broadband internet traffic,” in *IMC ’09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. New York, NY, USA: ACM, 2009, pp. 90–102.

[17] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, “Taxonomy of computer security flaws,” *ACM Computing Surveys*, vol. 26, no. 3, pp. 211–254, September 1994.

[18] J. Mirkovic and P. Reiher, “A taxonomy of ddos attack and ddos defense mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 39–53, 2004.