Understanding the Challenges of Blocking Unnamed Network Traffic

Kaspar Hageman*, Egon Kidmose*, René Rydhof Hansen†, Jens Myrup Pedersen*
*Department Electronic Systems, †Department of Computer Science
Aalborg University, Denmark

***Abstract*—Network traffic that is not preceded by any Domain Name System (DNS) resolutions is referred to as unnamed traffic. Any DNS-based security system is ineffective against malicious content distributed through this traffic. In this paper, we introduce a novel method for identifying unnamed traffic based on the correlation of flows and DNS responses extracted from raw network traces. We describe two challenges that affect the validity of our method, and how to handle them. By applying our method to a one-week trace of network traffic, we illustrate that unnamed traffic is ubiquitous in a university network across nearly all client systems, destination IP addresses, and destination services. We conclude by presenting several open problems that prevent us from blocking unnamed traffic for security reasons.**

***Index Terms*—DNS, network flows, unnamed traffic, network security**

I. Introduction

For network administrators of large-scale networks with a multitude of clients, it is of natural interest to ensure the safety of the clients and prevent criminal activities from taking place. Automated security mechanisms focus on eliminating access to certain hosts that are considered malicious. A popular class of these methods employs the analysis of Domain Name System (DNS) traffic for identifying hosts and quantifying the intent of traffic towards them. These methods are naturally challenged by *unnamed traffic*, the traffic that is not relying on DNS to resolve domain names to IP addresses, which completely circumvents these DNS-based security methods. Not only malicious applications [1]–[6] but also benign applications [7], [8] are known to rely on unnamed traffic, making network operators unable to simply block out all unnamed traffic. This dilemma enables the unnamed traffic to exist within networks, while potentially being a crucial component of malicious activities.

In this work, we take a first step towards identifying and analyzing unnamed traffic from raw traffic, with the future goal of preventing malicious software to communicate through this traffic with services outside a network. More specifically, we contribute with the following:

- We describe a novel process of collecting, processing and correlating raw network traffic into anonymized named and unnamed network flows.
- We analyze the impact of DNS encryption methods and caching of DNS records on the validity of this process.
- We apply our method to traffic from a medium-sized university network to illustrate the characteristics of existing unnamed traffic.

II. Background

The DNS provides a method for translating domain names to IP addresses [9]. Client systems resolve domains to IP addresses by querying sets of distributed *name servers*, or NSs, for *resource records*, or RRs. Each RR comes with a Time-to-live (TTL) value that specifies how long the records should be cached. Plain DNS resolutions are transmitted over a network unencrypted, allowing network operators to monitor the traffic and make restriction policies for certain domains. Several privacy extensions to DNS have been proposed over the years, most notably DNS-over-HTTPS (DoH) [10] and DNS-over-TLS (DoT) [11], which both rely on Transport Layer Security (TLS) for establishing an encrypted channel to communicate over. These extensions threaten the ability of network operators to monitor the queries if clients do not employ the network operator's DoH or DoT-enabled DNS server.

Flow monitoring was introduced as a scalable alternative to raw traffic monitoring. Instead of capturing information about individual packets, the packets are aggregated into *flows*. A flow is identified as 'a set of IP packets passing an observation point in the network during a certain time interval, such that all packets belonging to a particular flow have a set of common properties' [12]. The set of common properties is referred to as a *flow key*, and is often a 5-tuple[1]. In addition to this key, each flow contains aggregated data fields, such as a beginning and end timestamp, byte counts, and packet counts.

III. Related work

Earlier works on worm spreading mitigation recognized that a lack of DNS traffic was suspicious and blocking such traffic prevents the worms from fully spreading across a network [1]–[6]. These projects have performed controlled experiments, in which an isolated network is infected with a worm, but they do not consider the impact of blocking unnamed traffic on benign traffic in their results. Janbeglou *et al.* produced several papers related to unnamed traffic from a benign traffic standpoint. In [7], they conduct a passive analysis on several datasets to understand its content, followed up by [8] in which they find that several popular benign peer-to-peer applications

[1] Source and destination IP addresses, source and destination port numbers and the IP protocol number

978-1-6654-0601-7/22$31.00 © 2022 European Union

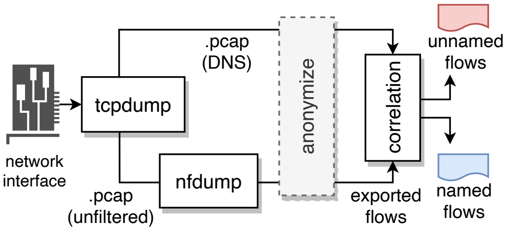

Fig. 1: The different components to extract the data

rely on unnamed traffic to function. Both of these directions give insight into a specific *known* type of traffic (*i.e,* worm spreading or popular applications). This paper complements these works by focusing on the challenges of the unnamed traffic extraction process and by analyzing unnamed traffic on a more general level rather than a per-application basis.

IV. Unnamed Traffic Identification

We propose a pipeline of components to process raw network traffic (as captured from a network interface) and produce named and unnamed flows (Figure 1). This process relies on a monitoring device to passively collect network traffic passing through a network. Using `tcpdump`, raw traffic is captured and written to `.pcap` files. These `.pcap` files are used to extract both flows and DNS responses, using `nfdump` and `tcpdump` with a filter respectively. More specifically, we extract any IP addresses from DNS responses for A/AAAA queries, *i.e,* queries for resolving domain names to IP addresses, and follow CNAME records. Both the resulting flows and DNS responses are anonymized and afterwards we identify the preceding DNS resolutions for each flow. Flows without a preceding DNS resolution are reported as unnamed.

a) Anonymization: The anonymization process must preserve the ability for us to correlate flows and DNS resource records, and as such there must be a one-to-one mapping between an unanonymized and anonymized IP address. For IP addresses within the monitored network, we employ Crypto-PAN [13] to anonymize CIDR host identifiers and preserve the network prefixes. As a result, we are still able to identify which hosts are located within the monitored network but do not know the original IP address (before anonymization). All MAC addresses are overwritten with a default value, removing any identifiable information contained in these fields. For the anonymization of `.pcap` files, we recompute the checksum of packets so that conventional network tools accept these traces.

b) Correlation: For a flow to be preceded by a DNS RR, they must match on the following properties: (1) the source IP address of the flow must be equal to the requesting IP address of the RR, (2) the destination IP address of the flow must be equal to the resolved IP address of the RR, and (3) the start timestamp of the flow must occur after the observation of the RR and before its TTL expires.

A. Validation

We conduct a set of controlled experiments to ensure that our method correctly correlates flows and DNS RRs. For these experiments, we prepared a Docker container to execute a simple task and collected the resulting traffic in a raw `.pcap` file. We run the pipeline on this file and analyze the identified number of named and unnamed flows. We evaluated the following use cases: an idle Ubuntu container without any applications running, a single HTTP request, a basic Nmap scan, a Chrome or Firefox browser visit `facebook.com` or play a Youtube video. We hypothesize that all experiments, except for the one involving Nmap, generate no unnamed traffic.

TABLE I: Results of correlating traffic from several basic network activities, showing the number of DNS queries, and breakdown of the number of flows (F = Firefox, C = Chrome).

Activity	**DNS**	**Flows (excl. DNS)**		
	Responses	Total	Named	Unnamed
Idle	0	0	0	0
HTTP req.	1	1	1	0
Nmap	0	49991	0	49991
Facebook (F)	53	44	44	0
YouTube (F)	71	79	78	1
Facebook (C)	2	10	10	0
YouTube (C)	7	22	22	0

The results are shown in Table I, illustrating the number of observed DNS responses, total flows, named flows, and unnamed flows. For the reported flows, we exclude flows that correspond to DNS resolutions. Nmap only generates TCP packets with the `SYN` flag set, to evaluate if a particular port is open, hence the large number of observed unnamed flows. The results confirm our hypothesis, except for Firefox generating a single unnamed flow when playing a YouTube video. Although these experiments are limited in scope, they show that even benign traffic – in this case watching a YouTube video – can lead to unnamed traffic, breaking the assumption that network traffic is generally preceded by DNS.

V. Results

We collected a dataset from a medium-sized university network in Denmark, spanning one week from 11:00 AM, October 4th, 2021 until 09:30 AM, October 11th, 2021. The data was collected using our pipeline deployed at the VPN server of the university network, monitoring the traffic of any client that is connected to the university's VPN. We are primarily interested in blocking network traffic from leaving a network and therefore we only considered traffic between internal and external IP addresses, omitting any internal traffic. Moreover, the dataset comprises IPv4 traffic only, as the university's VPN server operates with IPv4 addresses only. Raw traffic was fed into our pipeline in two-minute intervals.

In the process of extracting named and unnamed traffic, we ran into several challenges that potentially could threaten the validity of the extraction process, of which we handle two cases: TTL caching and the usage of alternative DNS resolvers.

a) TTL caching: We must take into account that the clients whose traffic we monitor have already locally cached

TABLE II: Estimated DNS query breakdown in type

Resolver types	# Queries	Percentage	# Nameservers
Local (plain)	11,498,808	92.61	2
Open (plain)	598,903	4.82	331
DoT	0	0.00	0
DoH	318,059	2.56	8

a significant number of DNS resource records prior to our measurement. Any flow matching those cached records would be (incorrectly) identified as unnamed in our measurements, as the DNS resolutions are missing in our dataset. The vast majority (99.16%) of DNS RRs in our dataset have TTL values lower than a day. By running our correlation on the full DNS dataset and discarding the first day of the flow measurement, we reduce the number of flows falsely marked as unnamed.

b) Alternative DNS resolvers: It is also possible for a flow to be falsely labeled unnamed if the preceding DNS resolution was unobserved by the DNS monitoring setup, which happens when clients use an alternative DNS resolver. We evaluate how impactful the use of these other resolvers is for the validity of our results. We focus on encrypted DNS traffic (DoT and DoH), and open DNS resolvers using plain DNS. Since we are unable to access the content of encrypted DNS traffic, and do not store the resolver IP addresses in our DNS dataset, we must in all cases estimate the usage of the various resolver types based on flows. For plain DNS, we estimate every packet to port 53 to be an individual DNS resolution, and distinguish between the university and open DNS resolvers based on the destination IP addresses of the flows. For DoT and DoH, the query volume is estimated by observing traffic towards TCP port 853 [11] for DoT and by observing traffic to known port 443 for IP addresses of known public DoH providers [14] for DoH respectively. In addition, for IP addresses for (1) which we observed plain DNS resolutions towards and (2) which we successfully can resolve a DNS query towards, we also consider traffic towards port 443 as DoH traffic. The query count is estimated to be the number of packets contained in the flows. For DoT and DoH, these flows include the packets for establishing a TCP and TLS connection, and as such relying on the packets in the flow overestimates the number of actual DNS requests. The estimated number of queries is shown in Table II. The vast majority of queries (97.44%) remain observable with the remaining queries being forwarded to eight DoH resolvers (notably missing any DoT resolvers), and as such we do not address encrypted DNS resolutions in our data analysis.

A. Data overview

Table III shows an overview of the collected data, both its flow part and its DNS part. For the flows, we report the total number of identified instances (*e.g,* flow count, unique IP addresses), and the instances involved in unnamed and named flows. Similarly, the DNS portion of the table denotes the total resolutions and involved IP addresses, and the instances related to DNS lookups that are (and are not) succeeded by at least one flow.

TABLE III: General overview of the data

	Flows		
	Total	Named	Unnamed
Flow count	36.83 M	18.93 M	17.90 M
Unique IP$_{\text{src}}$	975	956	975
Unique IP$_{\text{dst}}$	343,462	52,039	317,044
Unique services	69,419	6,297	66,392
Packets (sent)	1.64 B	0.98 B	0.66 B
Packets (rcvd.)	1.59 B	0.95 B	0.64 B
Bytes (sent)	464.09 B	284.42 B	179.67 B
Bytes (rcvd.)	2702.94 B	1738.26 B	964.68 B
	DNS		
	Total	Succeeded	Not succeeded
Resolution count	5.72 M	4.38 M	1.34 M
Unique IP$_{\text{req}}$	1,077	1,077	1,075
Unique IP$_{\text{res}}$	79,483	75,073	33,737
Unique query name	119,911	113,671	33,838

We collected nearly 37 million individual flows, of which 18.93 million (or 48.61%) were unnamed. The percentage of packets and bytes exchanged are similar, ranging between 35.69% for received bytes to 40.42% for sent packets. Interestingly, a significant portion of DNS resolutions was never followed up by a flow, indicating that these DNS resolutions have been performed unnecessarily.

a) Service and IP addresses breakdown: We aim to understand how the usage of unnamed traffic is distributed across source IP addresses, destination IP addresses, and the services that are being accessed (as inferred from the destination port and protocol). For each IP address and service, we compute the percentage of traffic origination from, or destined towards, that entity, in terms of flows, sent packets, and received packets. The cumulative distributions of these percentages are shown in Figure 2, only including services and addresses involved in at least fifty flows.

Nearly 40% of all services with at least fifty flows are accessed purely through unnamed traffic, and only 2% being accessed purely named (top figure). For the common web browsing services – HTTP and HTTPS – we observe 17.6% and 20.6% of flows to be unnamed, whereas DNS is almost exclusively accessed unnamed (as expected), as is illustrated in the figure using the dotted line matching the corresponding percentage on the x-axis.

The distribution of unnamed traffic usage across source IP addresses (middle figure) is highly concentrated between 20% and 80% of flows (solid black trace) being unnamed. Only 1.46% of these IP addresses has an unnamed traffic percentage outside this window. As a result, we cannot identify a small set of hosts that is responsible for most unnamed traffic. Instead, many source IP addresses contribute evenly to the amount of observed unnamed flows. The distribution of sent and received packets is even more skewed towards a higher unnamed percentage.

The distribution of the percentage of traffic for the destination IP addresses (bottom figure) shows that 13.55% and 9.19% of destination IP addresses are accessed exclusively through unnamed and named traffic respectively. For the IP

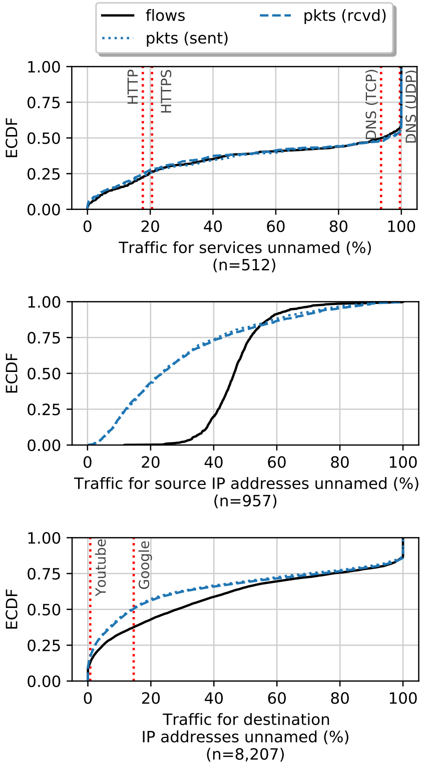

Fig. 2: The percentage of flows, and packets towards or originating from services (top), source IP addresses (middle) and destination IP addresses (bottom) being unnamed.

addresses associated with youtube.com and google.com, 0.77% and 14.55% of traffic is unnamed respectively.

VI. Discussion

Our results show that almost half of all flows in our data set are unnamed, Furthermore, we show that there is an almost universal existence of unnamed traffic towards services, originating from clients and towards destination IP addresses. From a network management perspective, this suggests that blocking unnamed traffic might negatively affect the normal operations of a large number of services, and nearly the entire client population of a network. Generally speaking, DNS-based protection mechanisms should be complemented by other countermeasures for protecting networks against attacks. Although we described several caveats in this work, there are several notable remaining challenges. These include both implementation (*e.g,* the performance of unnamed traffic extraction on the scale of a large network) and correctness challenges, of which the latter we address below.

a) Flow exporting interval: The methodology that we took in this work relies on `.pcap` being generated at a two-minute interval. This approach likely splits a single flow into two separate flows in some cases, of which the later flow may be falsely marked as unnamed. By increasing the interval, this likelihood decreases but remains present. Alternatively, one could export flows in real-time using a router that supports exporting flows. However, these flow exporters typically export flows prematurely for performance reasons, which makes this method imperfect too. The relatively short measurement interval in our data set could partially explain why unnamed traffic is so prevalent across all source IP addresses.

b) Changing of IP addresses: We collected traffic from a VPN server, which assigns IP addresses each time a client connects to the server. This IP address assignment does not guarantee that a computer system obtains the same IP address every time it establishes a connection. As such, our methodology is incapable of matching DNS records resolved with an old IP address against a flow after a change of the IP address of a client. To overcome this problem, the allocation of IP addresses by the VPN server could be taken into account, allowing us to correlate flows and DNS records across different source IP addresses. However, this would require the integration of a VPN server and our solution.

c) Clients not adhering to TTL values: The validity of the correlation of flows and DNS records is heavily dependent on the extent to which clients adhere to the TTL values that name servers return. These TTL values are a suggestion, and there is no enforcement that clients actually do so. It is possible for a client to cache a record indefinitely – resulting in potentially many falsely identified unnamed flows – but the extent to which this is done in practise is unknown. We recommend a more in-depth study into the DNS caching behavior of various stub resolver implementations (*i.e,* the DNS client running locally on an end user's system).

VII. Conclusion

In this paper, we proposed a method for extracting named and unnamed traffic from a raw network traffic trace, *i.e,* traffic that is or is not preceded by a DNS resolution respectively. By applying our proposed method to a one-week dataset, we illustrate two potential challenges, which we address. Firstly, we take into account that DNS records can be locally cached prior to monitoring network traffic, and conclude that nearly all records have a TTL of shorter than a day. Secondly, over 97% of DNS resolutions are unencrypted and can be observed, with the remaining DNS resolutions being a cause of misidentifying flows. A deeper dive into the unnamed traffic in our dataset reveals that unnamed traffic is ubiquitous in our dataset, being generated by nearly all source IP addresses, towards a significant number of services and affecting the majority of destination IP addresses. Relying on the "unnamedness" of traffic to make decisions about blocking this traffic is at this preliminary stage infeasible, and as such we recommend the research community to further investigate this traffic to better understand the intent behind unnamed traffic.

Acknowledgments

The authors would like to thank Finn Büttner for his continuous efforts towards data capturing at our university network. This research is carried out under the SecDNS project, funded by Innovation Fund Denmark.

References

[1] G. R. Ganger, G. Economou, and S. M. Bielski, "Self-securing network interfaces: What, why and how," 2002.

[2] D. Whyte, E. Kranakis, and P. C. Van Oorschot, "DNS-based detection of scanning worms in an enterprise network," in *NDSS*, 2005.

[3] K. Shahzad and S. Woodhead, "Towards automated distributed containment of zero-day network worms," in *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE, 2014, pp. 1–7.

[4] ——, "Empirical analysis of rate limiting + leap ahead (RL+LA) countermeasure against witty worm," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*. IEEE, 2015, pp. 2055–2061.

[5] M. A. Ahmad and S. Woodhead, "Containment of fast scanning computer network worms," in *International Conference on Internet and Distributed Computing Systems (IDCS)*. Springer, 2015, pp. 235–247.

[6] M. A. Ahmad, S. Woodhead, and D. Gan, "A countermeasure mechanism for fast scanning malware," in *2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)*. IEEE, 2016, pp. 1–8.

[7] M. Janbeglou, H. Naderi, and N. Brownlee, "Effectiveness of DNS-based security approaches in large-scale networks," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2014, pp. 524–529.

[8] M. Janbeglou, "Understanding and controlling unnamed Internet traffic, chapter 5: Blocking unnamed Internet traffic," Ph.D. dissertation, University of Auckland, 2017.

[9] P. Mockapetris, "Domain names - implementation and specification," Internet Requests for Comments, RFC Editor, STD 13, November 1987. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1035.txt

[10] P. Hoffman and P. McManus, "DNS queries over HTTPS (DoH)," Internet Requests for Comments, RFC Editor, RFC 8484, October 2018.

[11] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Specification for DNS over transport layer security (TLS)," Internet Requests for Comments, RFC Editor, RFC 7858, May 2016.

[12] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information," Internet Requests for Comments, RFC Editor, STD 77, September 2013. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7011.txt

[13] J. Xu, J. Fan, M. Ammar, and S. Moon, "Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme," in *10th IEEE International Conference on Network Protocols, 2002. Proceedings.*, 2002, pp. 280–289.

[14] DNS Privacy Project, "Public resolvers," https://dnsprivacy.org/public_resolvers/, accessed: 08-10-2021.