Detecting DNS hijacking by using NetFlow data

Andersen, Martin Fejrskov; Pedersen, Jens Myrup; Vasilomanolakis, Emmanouil

Published in:
2022 IEEE Conference on Communications and Network Security (CNS)

DOI (link to publication from Publisher):
10.1109/CNS56114.2022.9947264

Publication date:
2022

Document Version
Accepted author manuscript, peer reviewed version

Detecting DNS hijacking by using NetFlow data

Martin Fejrskov
Technology, IP Network and Core
Telenor A/S
Aalborg, Denmark
mfea@telenor.dk

Jens Myrup Pedersen
Cyber Security Group
Aalborg University
Copenhagen, Denmark
jens@es.aau.dk

Emmanouil Vasilomanolakis
Section for Cybersecurity Engineering
Technical University of Denmark
Kongens Lyngby, Denmark
emmva@dtu.dk

***Abstract*—DNS hijacking represents a security threat to users because it enables bypassing existing DNS security measures. Several malware families exploit this by changing the client DNS configuration to point to a malicious DNS resolver. Following the assumption that users will never actively choose to use a resolver that is not well-known, our paper introduces the idea of detecting client-based DNS hijacking by classifying public resolvers based on whether they are well-known or not. Furthermore, we propose to use NetFlow-based features to classify a resolver as well-known or malicious. By characterizing and manually labelling the 405 resolvers seen in four weeks of NetFlow data from a national ISP, we show that classification of both well-known and malicious servers can be made with an AUROC of 0.85.**

***Index Terms*—NetFlow, IPFix, DNS, hijacking, malware**

I. Introduction

The integrity protection offered by Domain Name System (DNS) security measures, such as DNS-over-TLS and DNSSec, can be completely circumvented by changing the configuration of DNS clients to use malicious DNS resolvers instead of trustworthy resolvers. DNSSec can be circumvented because it is the role of the DNS resolver, not the DNS client, to perform DNSSec validation. This approach has therefore historically been used by several malware families such as DNSChanger, DNSUnlocker, Koobface and others for diverse purposes such as pushing adware, redirecting to phishing or malware web pages, etc. [1] [2] [3]. Although these malware families target Windows machines, taking control of home routers in order to use DHCP to extend the malicious DNS configuration to all devices in a household is also an approach used in practise for example by the GhostDNS malware or in on-premises attacks [4] [5] [6].

The DHCP based approach limits the malware detection options, as typical IoT devices and home routers do not support host-based detection mechanisms such as anti-virus software available for mainstream operating systems. As an alternative to host-based detection, network-based detection mechanisms that work by passively inspecting the payload of the DNS traffic between the home router and 3rd party resolvers could be deployed by an Internet Service Provider (ISP) [7]. This is, however, not legal to implement in the European Union for privacy reasons [8]. For purposes of malware detection, ISPs are only allowed to process data found in customer traffic, if the data is already processed for transmission purposes (such as the information found in NetFlow records), and only if the data is anonymized before processing [9]. NetFlow records are emitted by routers, and typically contain information about the flows observed on a particular router interface, such as timestamp, source/destination IP address, TCP/UDP source/destination ports and similar flow-level information. An anonymized NetFlow based approach is therefore a legally viable option, and this detection approach will therefore be pursued in our paper.

Funded by Telenor A/S and Innovation Fund Denmark, 2022.

A. Research question 1

Although many papers analyse the maliciousness of the DNS traffic itself (such as DNS traffic used in DDoS attacks), including some that are based on NetFlow level information [10] [11] [12], we are not aware of any work that only uses NetFlow level features to assess whether a resolver performs record manipulation with either benign or malicious intent. Determining maliciousness solely based on NetFlow features could present a simpler (and therefore more desirable) option to an ISP, as the ISP would then not need to rely on procuring additional threat intelligence for resolver labelling. This observation provides the base for the first research question (RQ) examined in this paper:

RQ1: Can public resolvers be correctly classified as either malicious or non-malicious using ISP-level NetFlow data?

It is important to make the distinction between manipulation performed with benign (e.g. desired or regulatory) intent or malicious intent. There are many desired, benign or regulatory reasons for performing record manipulation (such as captive portals, parental control, ad removal, censorship, load balancing etc.), as well as many reasons for performing manipulation with malicious intent (malware infection, credential stealing, adware pushing, etc.). Furthermore, a single, specific record containing the IP address of an advertisement hosts could be manipulated differently with different intent. This distinction is illustrated in Figure 1. In this paper, all malicious resolvers are considered to be manipulating, but not all manipulating resolvers to be malicious, and to limit the scope of the paper, RQ1 will focus on public resolvers with malicious intent. In this

paper it is assumed that all malicious resolvers are public, as the malicious actor would otherwise need to explicitly disallow any non-infected clients, which does not seem like a viable approach.

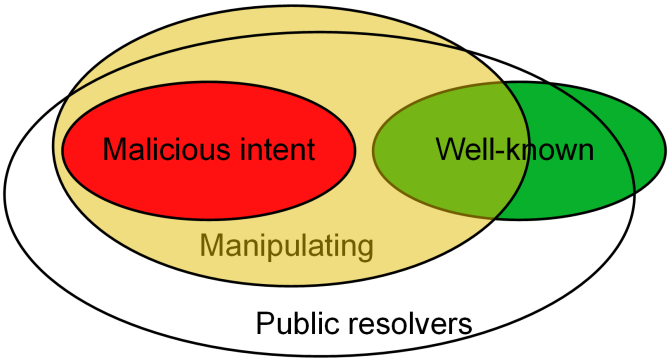

Fig. 1. *Venn diagram showing the relation between manipulating resolvers, well-known resolvers, public resolvers, etc.*

B. Research question 2

Whether a resolver is malicious or not is a property that describes the resolver itself. It does not describe if the user deliberately chose the resolver or if the resolver was chosen for the user by malware. Most users do not know or care about which resolver they use, and as a result, they use the default resolver assigned by equipment manufacturers or ISPs. For this paper it is assumed that if a user (or equipment manufacturer) should actively choose which resolver to use, the user will choose a well-known resolver operator. Well-known resolvers are defined as all DNS resolvers that are known (through an associated web page or similar) to be run by a publicly known organisation, no matter the amount of filtering or censoring applied for benign/desirable/regulatory purposes[1]. This implies that all well-known resolvers are benign, but not that all benign servers are well-known, as illustrated in Figure 1. This assumption and definition provides the base for the second research question examined in this paper:

RQ2: Can public resolvers be correctly classified as either well-known or not using ISP-level NetFlow data?

C. Contributions

RQ1 and RQ2 classify resolvers in one of four classes, depending on whether they are considered well-known or not, and if they are considered malicious or not. Following the arguments presented above, all public, malicious resolvers will be a subset of the manipulating and non-well-known resolvers. This resolver categorization can potentially be used in firewalls by ISPs to black/white-list resolver IPs on behalf of a group of consenting users, or the categorization can be combined with user-specific NetFlow records to discover and notify consenting users of a potential malware infection.

The contributions of the paper are therefore twofold: First, we introduce the concept of using whether a resolver is well-known or not for classification. Second, we show how accurately NetFlow data can be used to identify well-known and/or malicious resolvers.

This remainder of the paper is organized as follows: Section II describes the method used to answer the research questions. Section III shows the result of applying the method, and the results are discussed in Section IV. Section V describes related work and Section VI concludes the paper.

II. Method

To answer the research questions posed in the introduction, we apply the following four steps, which are described in further details in this section. First, the IP addresses of the DNS servers that are considered public resolvers are identified, and the NetFlow records related to any other IP addresses are discarded. Second, a number of features are extracted from the remaining NetFlow records and auxiliary features such as the DNS PTR records of the resolver IP addresses are added. The third step is to establish a set of labels that are used as ground truth for supervised machine learning. The fourth step is to apply machine learning to show if the classification is feasible, thereby answering the research questions posed in the introduction.

A. Identifying public resolvers

Some DNS servers assume the role of both public resolver and authoritative servers, and for the purpose of this paper, these are considered public resolvers and their authoritative role is ignored. An IP address is considered to host a public resolver if all of the following three criteria are satisfied.

First, NetFlow data relating to port 53 or 853 must show unidirectional TCP or UDP traffic flows both to and from the IP address. Due to sampling, it is not a requirement that the unidirectional flows are related. TCP traffic must contain more than 54 bytes. The purpose of these criteria is to eliminate traffic related to port scans, TCP connections/handshakes with no DNS payload, DDoS amplification attacks and other irregular or irrelevant use cases.

Second, a DNS A-type query is issued towards the IP address using both DNS and DNS-over-TLS for a domain name under our control, where the valid response and authoritative servers are therefore known. The response must contain a syntactically valid no-error response record that contains an IP address. The purpose of this is to eliminate private resolvers and authoritative-only servers.

Third, the Recursion Available flag is considered, as authoritative servers and non-public resolvers are expected to set this flag to False. If the Recursion Available flag is set to True, the

[1]Examples include resolvers run by ISPs, resolvers run as an auxiliary service by organisations such as Google, Cisco, Baidu and Yandex, resolvers run by anti-virus vendors as part of their security service such as Avast and Norton, resolvers run by VPN providers such as NordVPN and PrivateInternetAccess, resolvers run by organisations with the purpose of avoiding filtering/censorship, such as TurboDNS and SmartDNS, resolvers run specifically to implement some kind of filtering such as Gamban for banning of gambling, and resolvers run by equipment manufacturers such as Dlink. Examples of non-well-known include home routers that are unintentionally configured to allow DNS resolution/forwarding and resolvers deployed on Azure/M247/Amazon infrastructure (with no public ownership information).

server is considered a resolver no matter if the IP address in the response is correct or incorrect. However, if the Recursion Available flag is set to False, the IP address in the response must be correct for the server to be considered a resolver. This is done in order to eliminate a number of authoritative-only servers that respond non-authoritatively to queries (for example with an IP address hosting a web page with a "This page does not exist" banner instead of providing an NXDOMAIN response) and to avoid eliminating resolvers that answer with the correct response record, but try to evade detection by setting the RA flag to false.

B. Features used to characterize public resolvers

The features chosen to characterize the public resolvers are listed in Table I. Two features warrant further elaboration in the following paragraphs.

ResolverPrefix: During a preliminary data analysis, we found several servers (both benign and malicious) within the same prefix. To exploit this as a feature, we choose to use a /24 prefix as a more narrow feature than the more traditional measures such as geographical location or BGP AS number.

PtrCategory: Many benign DNS resolvers have a valid PTR record indicating the role as DNS server. Similarly, many ISPs create a default PTR record for all their customers, that contains the IP address itself. To exploit the PTR record as a feature, the PtrCategory of a server is set to "DNS" if the PTR record contains the words "dns", "ns[1-4]", "ns0[1-2]", "resolver" or starts with "ns.". The PtrCategory is set to "IP" if the record contains four numbers separated by ".". The PtrCategory is "NoPTR" when no PTR record exists, and "Uncategorized" if none of the above applies.

Some features are for various reasons intentionally not used to describe DNS resolvers, such as

- features directly or indirectly controllable by the malicious actor. This includes many NetFlow features such as packet/byte counts, query source port number and TCP vs. UDP. It also includes features found by actively probing the DNS resolver, whether the resolver is also an authoritative server, or whether there are any services available on other TCP/UDP ports on the resolver server.
- features unavailable due to anonymization requirements, such as an mxtoolbox.com lookup of the client IP address. Mxtoolbox.com provides information about whether a particular IP address is listed in popular public/commercial threat intelligence databases.
- features unavailable due to the use of a high sampling rate when creating NetFlow records. This includes features that require that several related flows are all observed in NetFlow records, such as if a flow towards a resolver is preceded by a DNS query towards the ISP's default DNS resolvers, or if a certain sequence of flows is always observed towards certain resolvers.

C. Features used to label public resolvers

Table II lists the labels used as ground truth. Three features warrant further elaboration below.

AdResponse: The purpose of some benign DNS resolvers is to remove advertisements. To identify these, the AdResponse feature denotes if an A record query for 6 popular ad hosts[2] return IPs owned by Doubleclick/Google, as identified by a PTR record ending in "1e100.net.". If the IP contained in any A record is listed on any of the blacklists on mxtoolbox.com (a web page that queries the blacklists of multiple blacklist vendors), the AdResponse feature is set to "malicious". If some A records are correct, and some are incorrect (but not blacklisted), the AdResponse feature is set to "inconsistent".

UpdateResponse: Malicious DNS resolvers could block access to the update servers of anti-virus products, operating systems or similar, in order to avoid that any updates to these products would trigger a malware detection or detection of a choice of malicious resolver. To identify such resolvers, the same approach as for the AdResponse is used for 8 domains[3].

Webreference: This feature indicates the result of a manually performed search for *all* of the IPs identified as resolvers IP and their associated PTR record using Google, the Whois database and various publicly available lists of resolvers IPs. A resolver is classified as Benign if it satisfies the criteria for a well-known resolver as defined in Section I, as Malicious if the search indicates that the IP belongs to a malicious resolver, and as Unknown if the search did not provide any further insight.

The approach for the AdResponse and UpdateResponse features are inspired by Kührer et al. [13]. Although they include more feature categories (without disclosing the exact domains used), we consider AdResponse and UpdateResponse the most relevant to our paper. The specific choice of domain names are based on the market prevalence of the related companies, the company's documentation about which domains are used for the purposes, and the prevalence of the domains as observed in Telenor Denmark's DNS resolvers.

The features AdResponse, UpdateResponse, Blacklisted and Webreference are combined into two binary labels, called Wellknown and Malicious, as elaborated in Table II. The use of the combination of the Blacklisted and Webreference features to construct the Malicious feature is necessary as known resolvers such as CloudFlare's 1.1.1.1 and several DNS resolvers related to VPN services can be found on multiple blacklists. This could be caused by the VPN DNS service being located on the same IP/prefix as the VPN outlet, if any VPN customers are exhibiting malicious behaviour.

D. Algorithm

The features outlined above are both categorical and numerical in nature, and data is labelled and binary, which suggest that a supervised classification algorithm within the class of decision trees such as Random Forest (RF) or Gradient Boosted Trees (GBT) should be the most appropriate. The Area Under

[2]ad.doubleclick.net, www.google-analytics.com, googlesyndication.com, googleads.g.doubleclick.net, tpc.googlesyndication.com and pagead2.googlesyndication.com

[3]sadownload.mcafee.com, ncc.avast.com, ds.kaspersky.com, dc1.ksn.kaspersky-labs.com, dci.sophosupd.com, liveupdate.symantec.com, ctldl.windowsupdate.com and download.windowsupdate.com.

Source	Name	Description	Value type
NetFlow	ResolverPrefix	The /24 prefix of the resolver	Integer
	ClientCount	Number of unique client /24 IP prefixes seen in NetFlow records related to the resolver.	Integer
	DayCount	Number of days in which traffic from/to the resolver is observed	Integer
	RecordCount	The log_{10} count of NetFlow records related to the resolver	Integer
	RecordCountPerClient	The log_{10} count of NetFlow records related to the resolver divided by the number of clients	Integer
Auxillary	PtrCategory	The category of the resolver's PTR record. Feature values: DNS/ IP/ NoPTR/ Uncategorized	Categorical
	Qname	True if an A record with the resolver's IP observed by the ISPs own DNS resolvers	Boolean

TABLE I

Feature overview. The following features are used to describe each public resolver. The IP address of the resolver identifies the resolver, but is not used as a feature, and is therefore omitted from this list.

Source	Name	Description	Value type
Resolver probing	AdResponse	Indicates if the resolver answered with the correct IP address for a number of advertisement hosts. Feature values: Correct/Incorrect/Inconsistent/Malicious	Categorical
	UpdateResponse	Indicates if the resolver answered with the correct IP address for domains hosting software updates. Feature values: Correct/Incorrect/Inconsistent/Malicious	Categorical
Auxillary	Blacklisted	Indicates if the resolver IP is listed on a blacklist according to a lookup on mxtoolbox.com (excluding the SpamHaus PBL that simply lists IPs assigned to broadband customers)	Boolean
	Webreference	Indicates if the resolver is referenced on a website. Feature values: Benign/Unknown/Malicious	Categorical
Inferred	Wellknown	Indicates if Webreference is Benign or not. Label for RQ2.	Boolean
	Malicious	Indicates if either Blacklisted has value True (and Webreference is not Benign), Webreference has value Malicious, or AdResponse or UpdateResponse has value Malicious. Label for RQ1.	Boolean

TABLE II

Label overview. Input from four different features are combined to form the labels used as ground truth for each research question.

Receiver Operating Characteristic (AUROC) is used as hyper-parameter optimization metric through a 5-fold cross-validation using all combinations of three hyper-parameters: Tree depth (5, 10, 15, 20, 30), maximum number of bins (discretize continuous features) (10, 50, 100, 150) and number of trees (10, 20, 30, 40). 80% of the resolvers are used for model training and cross-validation, 20% of are used for test/prediction/evaluation.

The number of indices for the categorical ResolverPrefix feature will probably be close to the number of observed resolvers. To avoid such a large number of indices, and to avoid the large number of feature columns created by a one-hot-encoding, the prefix is converted to its integer representation and considered as a continuous feature instead. As a continuous feature, the ResolverPrefix feature will be binned, making it more likely that numerically close prefixes will be classified similarly by the model. This seems like a reasonable approach given that organisations are typically allocated larger IP prefixes than individual /24 prefixes.

III. Results

This section describes the results of applying the method described in Section II. In subsection III-A, a description is provided of the data used, as well as how the data is characterized in terms of the features and labels introduced in Section II. In subsection III-B, the results of applying machine learning algorithms to predict labels are presented. The results are discussed in Section IV.

A. Data characteristics

The primary data source used in this paper is four weeks of NetFlow data collected from 2021-11-25 to 2021-12-22 with a sample rate of 1:1024 at the Border Gateway Protocol (BGP) Autonomous System (AS) border routers of Telenor Denmark. Telenor Denmark is a national ISP in Europe with 1.5M mobile and 100k broadband subscriptions.

For legal end ethical reasons, the client (Telenor customer) IP is anonymized to a /24 prefix in each NetFlow record before any further processing. IP address truncation is chosen as anonymization technique, as this is the only technique that does not allow for re-identification of a host [14]. Features and labels are extracted at least every 5 days during the collection period. After this process all NetFlow records (including the anonymized client IP) are discarded for that time period. Therefore, the resulting dataset used for analysis in this paper does not depend on storing any Personal Identifiable Information (PII) relating to the clients. As the resolver IPs could potentially belong to subscribers at other ISPs, the dataset used will only be retained for the duration of the writing of this paper. We find these measures sufficient to reduce privacy risks to a satisfactory level.

The criteria listed in Section II-A for identifying public resolvers in actual use by Telenor customers are satisfied by 405 IP addresses during the data collection period. Of the 405 resolvers, 62 have the label Malicious set to True, and 259 have the label Wellknown set to True. As expected from the model introduced in I, none of the resolvers have both labels set to True. We will therefore for the ease of reference refer to the resolvers as being either malicious, wellknown or unknown (when both the malicious and well-known label is set to False). Of the 62 malicious resolvers, 5, 2 and 25 are categorized as Malicious in the AdResponse, UpdateResponse or Webreference features, respectively. 35 of the 62 are categorized as True in the Blacklist feature.

Figures 2-6 illustrate the number of resolvers that are tagged with which label for each of the features used to describe the resolvers (as described in Table I).

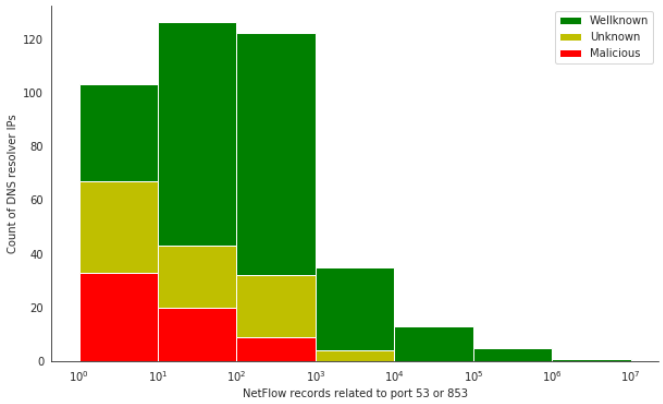

Fig. 2. *Histogram showing the count of DNS resolver IP addresses for which a certain number of NetFlow records are reported. As an example, between 10 and 100 NetFlow records are observed from approximately 125 different DNS resolvers IPs.*

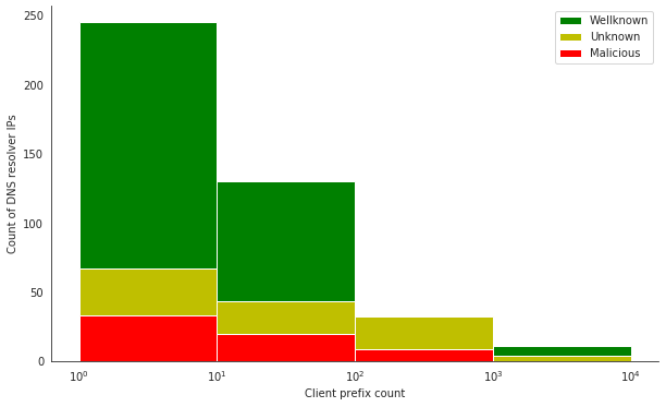

Fig. 3. *Histogram showing the count of DNS resolver IP addresses that are observed to be used by a certain number of client /24 prefixes. As an example, approximately 30 DNS resolvers IPs are used by between 100 and 1000 client prefixes.*

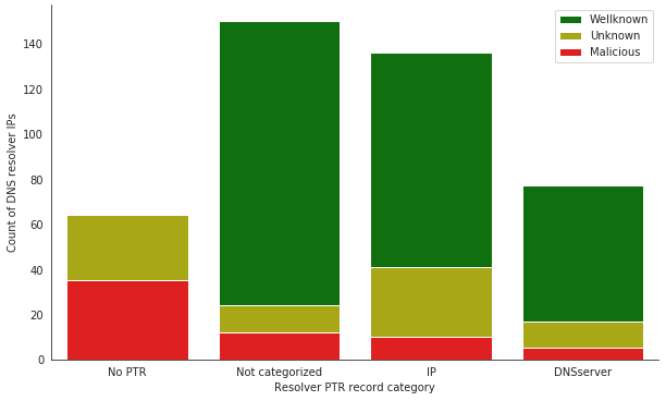

Fig. 4. *Histogram showing the count of DNS resolver IP addresses that have a PTR record in a certain category. As an example, approximately 75 IP addresses have a PTR record whose name hints that this could be a DNS server.*

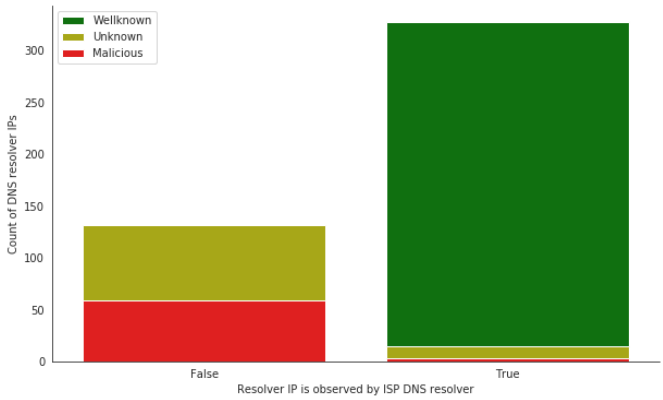

Fig. 5. *Histogram showing the count of DNS resolver IP addresses that are also observed in response records in Telenor's default DNS resolvers. As an example, 130 resolver IPs were not observed in Telenor's DNS resolver response records.*

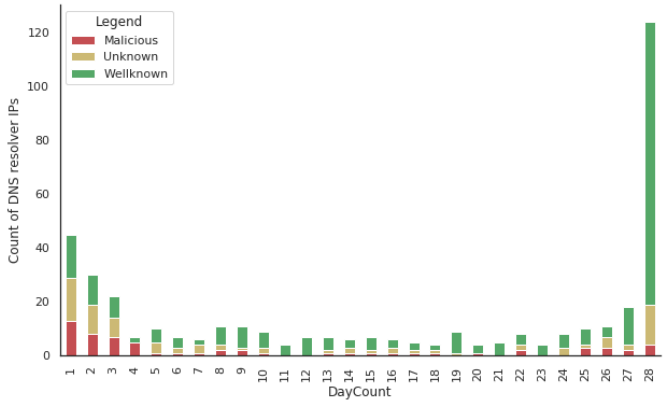

Fig. 6. *Histogram showing the count of DNS resolver IP addresses that are observed in a certain number of days. As an example, approximately 20 different DNS resolvers IPs are observed for a total of exactly 3 days during the data collection period.*

B. Label prediction

As outlined in Section II, both the Malicious and Wellknown labels are predicted by either a Random Forest or Gradient Boosted Tree algorithm. The results in this paper are found using the implementation provided by PySpark.

The set of hyperparameters with the highest AUROC for each label and each algorithm are listed in Table III. Unless noted otherwise, the rest of this paper only presents details relating to the best model found for each label, which is an RF based model for the Wellknown label and a GBT based model for the Malicious label.

The ROC curves can be found in Figures 7 and 8. These show the True Positive Rate and False Positive Rate at various probability threshold settings.

A confusion matrix for each of the models can be found in Tables IV and V. The confusion matrix for the Wellknown label shows the absolute number of resolvers in the test set that were predicted to have label Wellknown set true or false, as compared to the curated label. Both matrices are created using

Label	Algo	Best model			
		AUROC	MaxDepth	MaxBins	NumTrees
Wellknown	RF	0.85	5	50	10
	GBT	0.77	5	100	N/A
Malicious	RF	0.83	5	100	30
	GBT	0.85	5	100	N/A

TABLE III
Hyperparameters yielding the best AUROC for each label and algorithm.

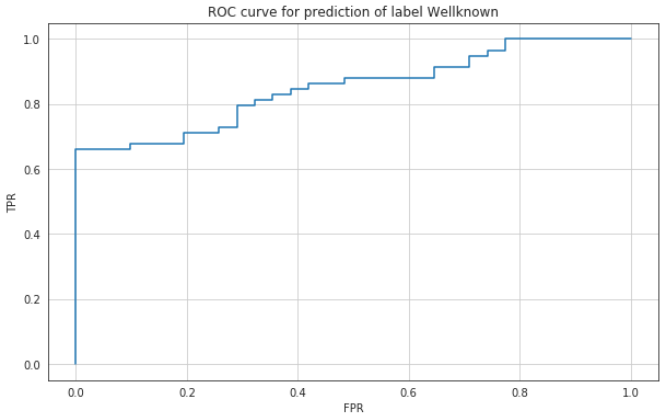

Fig. 7. *ROC curve for the prediction of the label Wellknown.*

a probability threshold of 0.5, as this threshold is among the set of threshold values that provide a high F-score. The accuracy ($\frac{TP+TN}{P+N}$) represented by the two matrices is 0.74 and 0.87.

The feature importances for each of the models can be found in Table VI. This quantifies the importance of a particular feature as an average across all trees in the model [15].

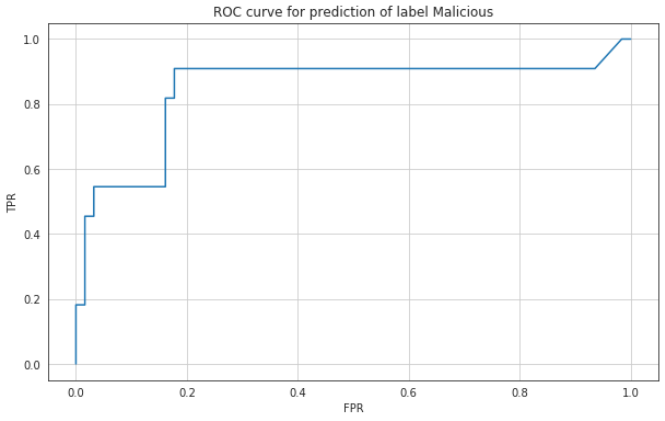

Fig. 8. *ROC curve for the prediction of the label Malicious.*

	Predicted True	Predicted False
Labelled False	15	16
Labelled True	51	8

TABLE IV
Confusion matrix for the prediction of the label Wellknown.

	Predicted True	Predicted False
Labelled False	4	58
Labelled True	6	5

TABLE V
Confusion matrix for the prediction of the label Malicious.

Feature	Wellknown label	Malicious label
ResolverPrefix	0.18	0.47
ClientCount	0.07	0.06
DayCount	0.17	0.09
RecordCount	0.24	0.10
RecordCountPerClient	0.15	0.13
PtrCategory	0.19	0.13
Qname	0.01	0.02

TABLE VI
Feature importances.

IV. Discussion

The exclusion of authoritative-only servers based on the Recursion Available flag did not increase the accuracy of the model as much as expected. Running the model training without considering the RA flag yields an AUROC of 0.84 and 0.82 (instead of 0.85) based on 455 servers instead of 405 servers. As the RA value returned by a malicious resolver can be controlled by the malicious actor, it might be a better option not to consider this flag at all.

Although benign use cases exist for responding with incorrect or no IP address for a lookup for an ad domain (as indicated by the AdResponse feature), we can think of no benign use case for responding with incorrect or no IP address for a lookup for a software update domain (as indicated by the UpdateResponse feature). It is therefore tempting to consider such values of the UpdateResponse feature as an indication of a malicious resolver. However, experiments show that this will also label known benign servers as malicious, such as some of the servers owned by Neustar and Norton. Therefore, we consider it the best approach only to label resolvers as malicious if they answer any query with an IP that is known to be malicious.

Figures 2-6 reveal both expected and unexpected observations about the labels assigned to resolvers. Figures 2 and 3 show, as expected, that servers with much traffic and many clients are typically not malicious. However, they also reveal a surprising amount of low-volume resolvers, for example resolvers with less than 10 clients or less than 1000 NetFlow records. Further investigation of the low-volume servers reveal that they seem very diverse and hard to characterize in general terms. Figure 4 shows that only a fraction of DNS resolvers have a PTR record that indicates that the host is a DNS resolver. DNS resolvers without such a PTR record are typically hosted on cloud infrastructure, where the PTR record indicates the IP address of the host and/or the cloud providers name instead. Conversely, Figure 5 shows that the IP address of well-known servers are typically represented in an A record that is served by Telenor's default resolvers. This is surprising, as a DNS lookup is not a typical precursor to the use of a DNS resolver, as a resolver is typically configured in a device using the IP

address of the resolver (except for DoH and DoT resolver configurations).

The number of days within the 4 week dataset where traffic is observed for a given resolver is illustrated in Figure 6. An interesting observation about this graph is that although the total number of resolvers increased, the overall shape of the histogram did not change much no matter how many weeks of data were used for the graph. This could indicate that the data collection period used for characterization could be reduced to much less than the 4 weeks used in this paper.

The AUROC values and the confusion matrices indicate that labels can indeed be predicted based on NetFlow data, although we do not consider the AUROC values high enough for operational/production use. During the data analysis we observed that the AUROC values of 0.85 can vary depending on the specific sampling in the training/test data split. This variance is not systematically analyzed, however, the reported value of approximately 0.85 seems to appear often, but values as low as 0.80 and as high as 0.87 have also been observed.

Neither label is well balanced (259 well-known resolvers and 62 malicious resolvers from a total of 405 resolvers) and therefore the class imbalance problem needs to be considered. For this purpose we repeated the model training with undersampling, and results indicated that this yielded slightly lower AUROC values (in the range of 0.78 to 0.82) for both labels and both algorithms. It seems reasonable to assume that this is caused by the relatively small dataset available for training. It could therefore be interesting to repeat the experiment on an even larger dataset, preferably with a larger fraction of malicious resolvers.

The feature importances listed in Table VI show that most features contribute to the model. The Qname feature, indicating if the ISP's resolvers have seen the public resolvers IP address in a DNS response record, is the least significant feature, also when training with undersampling. This is surprising given that Figure 5 indicates that most well-known resolvers have Qname=True, and most malicious resolvers have Qname=False. We have no credible explanation for this discrepancy.

V. Related work

The general topic of DNS hijacking can be be split into 4 different subtopics, where the hijacking is implemented by manipulating response records in (1) resolvers and forwarders, (2) middleboxes such as firewalls, (3) authoritative name servers, or (4) by manipulating the DNS resolver IP address configuration on client devices to direct DNS traffic to malicious resolvers [16]. As outlined in the introduction, the focus of our paper is restricted to *malicious* resolvers in the *client hijacking* use case and the *passive* collection of *NetFlow* data. The related work is described in this section and the properties highlighted above are summarized in Table VII.

Two papers focus specifically on the use case of changing the DNS resolver IP. Dagon et al. call this use case a corruption of the dns resolution path and discuss the various options for detecting the use case [17]. The authors inspect DNS records obtained passively at a local campus gateway, and by establishing a set of name servers and request open resolvers to query these name servers. Trevisan et al. passively analyzes DNS responses and compare individual responses to detect any manipulated answers [7]. Interestingly, the techniques used by the two papers mentioned above can be used for the use case of detecting all the DNS hijacking use cases mentioned above, not only for detecting malicious DNS resolver IP changes.

	Related Work					
Aspect	7,17	18,19	13,20	21,22,23,24	25	16,26
Client hijack	✓					
Passive approach	✓			✓	✓	✓
NetFlow data					✓	
Maliciousness	✓		✓			✓

TABLE VII
Notable related work and aspects in focus.

Several studies quantify the number of open and/or malicious resolvers through an Internet-wide, active probing for open resolvers [18] [19] [13]. Most recently, Park et al. concluded that about 3 million open resolvers exist, and show that more than 26k open resolvers return an incorrect and malicious IP address reported to serve malware, phishing attempts etc. [20].

Some papers measure which resolvers are used by introducing *observer-controlled authoritative servers* and zones. These are combined with advertisement campaigns [22], visits to self-owned websites [23] or a large number of remotely controlled clients in various world regions that send DNS requests [24]. Approaches relying on data from browsers or the installation of specific apps will not capture any traffic from unsupported device types, such as IoT devices, home routers etc., which is central to our paper.

Other papers focus on inspecting DNS data obtained by passively mirroring *DNS traffic at the application layer* at a non-authoritative point in the DNS chain. This includes dump of DNS flows at application layer at an ISP [7], at a LAN gateway [26], at a campus gateway [17] and using DNS data from the Farsight database [16]. It should be noted that the Farsight database is built upon voluntary participation by resolver owners, and so it is unlikely that queries towards intentionally malicious resolvers would be represented in this database.

Finally, use of *data from clients* collected in the Open Observatory of Network Interference (OONI) database is used by Radu et al. [21], and passively mirroring *DNS traffic at the network/transport layer* through NetFlow at a national ISP is used by Fejrskov et al. [25]. Their focus is, however, on the use of major/well-known 3rd party resolvers, not on the more rarely used, potentially malicious resolvers.

The maliciousness of DNS responses are evaluated using various techniques, such as by use of open threat intelligence [20] [17], by probing HTTP/POP3/ IMAP/SMTP services on the resolved IPs [13] [26], by detecting differences in responses for similar queries [7] and by detecting NS record changes [16].

Of the papers mentioned above, only Dagon et al. and Trevisan et al. focus on the use case of changing the DNS resolver IP [17] [7]. Interestingly, the techniques used by the

two papers can be used for detecting all the DNS hijacking use cases mentioned initially in this section, not only for detecting malicious DNS resolver IP changes. As mentioned above, these papers inspect DNS data to achieve their results and are therefore fundamentally different from our paper.

VI. Conclusion

This paper investigates if it is possible to classify public resolvers as malicious and/or well-known using features derived from NetFlow data. Our suggested NetFlow based approach comes with a number of advantages compared to existing methods: *i)* it is legal to be deployed by ISPs in the EU, *ii)* it does not rely on excessive Internet-wide scanning, *iii)* it does not rely on features that are controllable by a malicious actor.

Using Random Forest and Gradient-Boosted Trees on 7 different NetFlow-related features we show that it is indeed possible to classify a resolver as well-known or malicious with an AUROC of 0.85 (around 0.80 using undersampling). This shows that NetFlow features can indeed contribute to the classification, although the value is not high enough for a NetFlow-only approach to be considered for operational use. It may be possible to create a better model if a larger data set is used, especially if the dataset has a better balance between malicious and benign resolvers.

In our paper, active probing of resolvers is intentionally only used for labelling / model training purposes, not as features, as the purpose is to investigate the value of a NetFlow-only approach. To increase the accuracy of the classification, we consider a hybrid approach adding such features as the most interesting approach for future work.

References

[1] V. Verónica and R. Gibb, "Adware's new upsell: malware," BSides Calgary, 2016. [Online]. Available: http://dx.doi.org/10.13140/RG.2.1.1218.1365

[2] S. Alrwais, A. Gerber, C. Dunn, O. Spatscheck, M. Gupta, and E. Osterweil, "Dissecting Ghost Clicks: Ad Fraud Via Misdirected Human Clicks," ACM Annual Computer Security Applications Conference (ACSAC), 2012. [Online]. Available: http://dx.doi.org/10.1145/2420950.2420954

[3] J. Baltazar, J. Costoya, and R. Flores, "The real face of KOOBFACE: The largest Web 2.0 botnet explained," Trend Micro Threat Research, 2009. [Online]. Available: http://www.trendmicro.com.ph/cloud-content/us/pdfs/security-intelligence/white-papers/wp_the-real-face-of-koobface.pdf

[4] G. Ye, "70+ different types of home routers(all together 100,000+) are being hijacked by GhostDNS," Netlab 360, 2018. [Online]. Available: https://blog.netlab.360.com/tag/ghostdns/

[5] Z. Yin, "Domain Resolution in LAN by DNS Hijacking," International Conference on Computer Engineering and Networks (CENet), 2020. [Online]. Available: https://doi.org/10.1007/978-981-15-8462-6_98

[6] J. Galloway, "AirBNBeware: Short term rentals, long term pwnage," Blackhat, 2016. [Online]. Available: https://www.blackhat.com/us-16/briefings/schedule/#airbnbeware-short-term-rentals-long-term-pwnage-2891

[7] M. Trevisan, I. Drago, M. Mellia, and M. M. Munafò, "Automatic Detection of DNS Manipulations," IEEE International Conference on Big Data, 2017. [Online]. Available: https://doi.org/10.1109/BigData.2017.8258415

[8] The European Parliament and of the Council, "Directive 2002/58/ec (the ePrivacy directive)," 2002. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002L0058

[9] M. Fejrskov, J. M. Pedersen, and E. Vasilomanolakis, "Cybersecurity research by ISPs: A NetFlow and DNS Anonymization Policy," International Conference on Cyber Security And Protection Of Digital Services, 2020. [Online]. Available: https://doi.org/10.1109/CyberSecurity49315.2020.9138869

[10] D. Huistra, "Detecting reflection attacks in DNS flows," 2013. [Online]. Available: https://pdfs.semanticscholar.org/4ad8/24537f212f70e25e4cbab55498f5a8e43942.pdf

[11] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak, "Detecting DGA malware using NetFlow," IFIP/IEEE International Symposium on Integrated Network Management, 2015. [Online]. Available: https://doi.org/10.1109/INM.2015.7140486

[12] R. Hananto, C. Lim, and H. P. Ipung, "Detecting network security threats using domain name system and NetFlow traffic," ICCSP: International Conference on Cryptography, Security and Privacy, 2018. [Online]. Available: https://doi.org/10.1145/3199478.3199505

[13] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going Wild: Large-Scale Classification of Open DNS Resolvers," IMC: Internet Measurement Conference, 2015. [Online]. Available: http://dx.doi.org/10.1145/2815675.2815683

[14] D. Sauter, M. Burkhart, D. Schatzmann, and B. Plattner, "Invasion of privacy using fingerprinting attacks," 2009. [Online]. Available: https://pub.tik.ee.ethz.ch/students/2008-HS/MA-2008-22.pdf

[15] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," Springer Series in Statistics, 2009. [Online]. Available: https://doi.org/10.1007/978-0-387-84858-7

[16] R. Houser, S. Hao, Z. Li, D. Liu, C. Cotton, and H. Wang, "A Comprehensive Measurement-based Investigation of DNS Hijacking," International Symposium on Reliable Distributed Systems (SRDS), 2021. [Online]. Available: https://cpb-us-e2.wpmucdn.com/faculty.sites.uci.edu/dist/5/764/files/2021/10/srds21.pdf

[17] D. Dagon, N. Provos, C. P. Lee, and W. Lee, "Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority," Network and Distributed System Security Symposium, 2008. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2017/09/Corrupted-DNS-Resolution-Paths-The-Rise-of-a-Malicious-Resolution-Authority-paper-David-Dagon.pdf

[18] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson, "Global Measurement of DNS Manipulation," USENIX Security Symposium, 2017. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-pearce.pdf

[19] W. Scott, S. Berg, and A. Krishnamurthy, "Satellite: Observations of the Internet's Stars," University of Washington. Tech Report, 2015. [Online]. Available: https://dada.cs.washington.edu/research/tr/2015/06/UW-CSE-15-06-02.pdf

[20] J. Park, R. Jang, M. Mohaisen, and D. Mohaisen, "A Large-Scale Behavioral Analysis of the Open DNS Resolvers on the Internet," IEEE/ACM Transactions on Networking, 2021. [Online]. Available: https://doi.org/10.1109/TNET.2021.3105599

[21] R. Radu and M. Hausding, "Consolidation in the DNS resolver market – how much, how fast, how dangerous?" Journal of Cyber Policy, 2019. [Online]. Available: https://doi.org/10.1080/23738871.2020.1722191

[22] P. Callejo, R. Cuevas, N. Vallina-Rodriguez, and Ángel Cuevas Rumin, "Measuring the Global Recursive DNS Infrastructure: A View From the Edge," IEEE Access, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2950325

[23] C. A. Shue and A. J. Kalafut, "Resolvers Revealed: Characterizing DNS Resolvers and their Clients," ACM Transactions on Internet Technology, 2013. [Online]. Available: http://dx.doi.org/10.1145/2499926.2499928

[24] B. Liu, C. Lu, H. Duan, Y. Liu, Z. Li, S. Hao, and M. Yang, "Who Is Answering My Queries: Understanding and Characterizing Interception of the DNS Resolution Path," USENIX Security Symposium, 2018. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-liu_0.pdf

[25] M. Fejrskov, E. Vasilomanolakis, and J. M. Pedersen, "A study on the use of 3rd party DNS resolvers for malware filtering and censorship circumvention," To appear in International Conference on ICT Systems Security and Privacy Protection (IFIP SEC), 2022.

[26] C. Huang, P. Zhang, Y. Sun, Y. Zhu, and Y. Liu, "SFDS: A Self-Feedback Detection System for DNS Hijacking Based on Multi-Protocol Cross Validation," International Conference on Telecommunications (ICT), 2019. [Online]. Available: https://doi.org/10.1109/ICT.2019.8798832