Local and Public DNS Resolvers: do you trade off performance against security?

Antonia Affinito
University of Napoli "Federico II"
Napoli, Italy
antonia.affinito@unina.it

Alessio Botta
University of Napoli "Federico II"
Napoli, Italy
a.botta@unina.it

Giorgio Ventre
University of Napoli "Federico II"
Napoli, Italy
giorgio.ventre@unina.it

***Abstract*—The Domain Name System (DNS) is a vital component of the Internet, used for all the operations performed over the network and, recently, also for protecting users from malicious activities. In this work, we analyze the behavior of DNS resolvers provided by three main Italian ISPs and contrast them with open, public resolvers provided by Google and Cisco. We consider two aspects. The first one is the time spent to perform a query and obtain a response from the resolvers, which has a considerable impact on the performance of most applications on the Internet. The second one is the capability to recognize domains associated with malicious activities, blocking related requests to protect users. The DNS response time is generally shorter for local resolvers since they are closer to the users. On the other hand, public resolvers are typically considered more efficient in detecting malicious domains. We performed a large number of DNS queries towards the different resolvers, both local and public, using different sets of domain names and different Internet access networks from main Italian providers. Our results confirm that the response time of local resolvers is shorter than the public ones. However, they also show that, unexpectedly, the protection level of local resolvers is largely comparable with the one of public resolvers. Consequently, you do not have to trade off security against performance. In addition, we study the impact of DNS over HTTPs, we unveil the different mechanisms implemented to block users from accessing malicious domains and assess the impact of caching on the obtained results.**

***Index Terms*—DNS resolver, Domain Name System, DNS over HTTPs, public DNS resolvers, local DNS resolvers**

I. Introduction

Translating domain names into their associated IP addresses is the main task of the *Domain Name System* (DNS). This is an indispensable component of the Internet, distributed over a global network of servers that are constantly in communication with each other to bring users to their websites or network resources [1]. DNS is an emerging topic in literature for its decisive impact on the performance of almost all internet activities. Plenty of previous work focused on the performance of DNS resolvers from several vantage points, also obtaining contrasting results [2]–[6]. On the other hand, millions of new domain names are registered every day, including the ones used by attackers to redirect victims to malicious destinations like malware, spam, phishing, and other insecure contents [7]. DNS traffic contains several meaningful features to identify domain names associated with such malicious activities. This is why DNS is more and more used to protect users from possible threats [8], together with other techniques based on e.g. machine learning and artificial intelligence [9], [10].

ISBN 978-3-903176-48-5© ©2022 IFIP

In this paper, we study the behavior of resolvers provided by three Italian commercial ISPs (TIM, Wind, and Fastweb) - and two public ones offered by Google and OpenDNS. We collected a dataset performing a high number of queries towards these resolvers, looking at domain names from four different categories: Top 1 million, Command and Control (C&C), Malware, and Phishing. We also considered encrypting DNS queries over HTTPS, a.k.a. DoH [11], [12]. Our analysis focuses on two relevant aspects. The first one is related to the *Response Time* (RT), which is the time required to get a DNS response after issuing the request. RT has a strong impact on the performance of most Internet applications. The second one concerns the *Response Code* (also RC or RCODE in the following), included in the DNS response, useful to figure out if the DNS server provides an IP address or not, e.g. to protect users from malicious hostnames. For example, a "NXDOMAIN" message can indicate either that the domain name does not exist or that it is related to a malicious activity, blocked by the resolver. Google and OpenDNS feature several strategically located resolvers that rely on anycast. But their response time is typically higher than the local resolvers [4], which are closer to the users. On the other hand, Google and OpenDNS are expected to detect more malicious domain names because they have a wider view of the network traffic than local resolvers.

The results of the response time analysis provide many interesting insights. First, we confirm that local DNS resolvers are faster than public ones: Fastweb is up to 86ms faster than Google, and up to 129ms quicker than OpenDNS. Google is generally faster than OpenDNS, contrary to what has been reported by other works [4], [13], [14]. Additionally, we uncover that there are no significant time differences between DNS and DoH both for Google and OpenDNS. We also studied the effect of the caching in the home router and saw that up to 40% of the domains can be cached at such router for up to 4 hours. The results related to the response codes show that resolvers use different approaches to block dangerous destinations, e.g. Fastweb, TIM, and Google return "NXDOMAIN", while Wind provides a "0" RC, but the IP address is related to a courtesy page. Also, OpenDNS achieves slightly higher performance with malware and phishing domains. This is somewhat expected as we use domain names collected by Cisco Umbrella, the same company owing OpenDNS. Furthermore, we do not find significant differences between DNS and DoH both for Google and OpenDNS. The most unexpected result however is that all the resolvers considered protect from malicious domains with comparable performance. That is to say that it is not necessary to trade-off performance and security, at least not anymore.

The contributions of this work can be summarized as follows: i) we perform an analysis on the DNS resolvers using a large dataset of domain names, including most popular as well as malicious domains; ii) we show that Italian local resolvers generally provide a comparable level of security of open, public resolvers used from all over the world and deployed by large companies like Google and Cisco; iii) we unveil the mechanisms employed by the resolvers to protect users; iv) we show that using local resolvers can save up to about 130ms on average for each DNS resolution; v) we contrast previous findings in the literature (e.g., [4]); vi) we show the impact of caching on obtained results.

II. Related work

In recent years, there has been an increasing number of works studying DNS resolvers [2], [3] also with a focus on a comparison between public and commercial ones [4]–[6]. An interesting analysis of the response times and the addresses returned by the local resolvers against Google and OpenDNS was conducted by Ager et al. [4]. They claim that Google and OpenDNS, in some cases, outperform the local resolvers in terms of the observed response times. This result is in contrast with ours: local resolvers we consider in this work typically show response times smaller than public resolvers. Moreover, we analyzed in much more detail the capability of such resolvers to block dangerous domains. Current literature pays particular attention to open DNS resolvers. Kuhrer et al. performed a long-term, large-scale analysis in order to study the changes over time and classify the resolvers according to several features like device type and software version. They have also deepened the DNS responses correctness querying the "A" record of 155 domains, divided into 13 categories, towards 22 million open DNS resolvers [5]. Dagon et al. carried out a similar analysis, but they only analyzed some samples of the DNS responses, and they did not provide detailed statistics except for Chinese splash pages [15]. Another interesting analysis on open resolvers was conducted by Park et al. [6], who compared their previous findings in 2013 showing that the number of resolvers providing incorrect responses is almost the same, those providing malicious responses has increased.

Companies have been focusing on comparing local and public resolvers. They claimed that it is more convenient to use public DNS than local ones, both for their response time and security protection [13], [14]. Our results show that their protection level is largely comparable while local resolvers are always faster than public ones. Google is generally faster than OpenDNS from our vantage points, and this outcome is also confirmed by other works [13], [14]. DNSPerf, instead, shows the opposite behavior: OpenDNS is faster than Google [16]. Different protocols have been implemented to encrypt DNS queries. They provide security and privacy, and they allow clients to send DNS queries to public DNS resolvers, preventing the ISP from seeing such queries [17]. Several works analyzed possible differences in performance between DNS and other encrypted DNS protocols, like DoH. Some of them claimed that the DNS response times are higher than those of DoH [12], [17]. Other works, instead, claim that it is not simple to choose the best DNS protocol for all clients because DoH response times can be both longer and shorter than DNS ones [18]. We compare the performance of standard DNS (UDP port 53) and DNS over HTTPS (DoH) and do not observe significant differences.

III. Data and tools

In this section, we describe our experimental setup and the datasets we used. We relied on the PyDig [19], a tool written in Python, to perform queries towards DNS servers and exercise various existing and emerging features of the DNS protocol. This tool features queries through DNS over TLS and DNS over HTTPS. We queried a considerable number of domain names, divided into two different categories. The first one is related Cisco Top 1 Million list, while the second one contains malicious domains collected by Cisco Umbrella analysts.

Concerning the first category, Cisco provides, every day, the list of the first one million domains (Top 1 Million) most commonly queried from all over the world to OpenDNS resolvers [1] [20]. We relied on this list following the suggestions provided by Scheitle et al.[2] [21]. To examine more in-depth the performance achievable in a wide set of conditions, and, consequently, for the purpose of having variability in our data, we created the dataset as follows. We pulled out the first and last 10,000 rows from the top 1 Million list, resulting in a dataset of 20,000 domains with the most and least common ones. We suspected that the most popular domain names might be benign with a higher probability, unlike the less popular ones that might be benign with a lower probability. This assumption was also derived from the work by Scheitle et al., stating that Cisco Umbrella list contains test domains or several domains with non-authorized gTLDs [21]. In addition, the clients adopting OpenDNS are not only PCs but also mobile and IoT devices. To further investigate the nature of the first and last 10k domain names included in the Top 1 Million. In this work, we rely on the "Domain Status and Categorization" API. This API belongs to the groups of APIs provided by Cisco Umbrella Investigate and used in several works [20], [22]. It returns the domain status, indicating whether a domain has been flagged as malicious by the Cisco Security Labs team (score of -1 for status), it is believed to be safe (score of 1), or it is still undecided (score of 0)[3]. 78.5% of

[1] https://umbrella.cisco.com/blog/cisco-umbrella-1-million

[2] The authors observed that there is a high daily fluctuation in the Top1Million. Therefore, it is important to specify the day on which we downloaded the Top1Million file: 09/10/2020.

[3] https://docs.umbrella.com/investigate-api/docs/domain-status-and-categorization-1

these domains is benign, a small percentage (0.1%) of them is malicious, and the nature of the rest of the hostnames (21.4%) is not further specified by Cisco Umbrella. In conclusion, the Top1Million dataset mostly consists of benign hostnames. We will refer to this dataset simply as TopCisco in the following.

The second category of the dataset is characterized by malicious hostnames collected by Cisco security analysts. Thanks to a collaboration with Cisco, we had access to a wide list of malicious domain names blocked by the Cisco Umbrella platform. The list is split into three different datasets, containing different kinds of malicious activities: *C&C* - domains associated with a Command & Control systems of botnets; *Malware* - domains associated with malware threats; *Phishing* - domains associated with phishing pages.

In summary, the datasets adopted for our analysis are four: **TopCisco**, characterized by 20,000 domains, mostly benign; **C&C**, characterized by 16,021 domains associated with Command & Control activities of botnets; **Malware**, characterized by 81,217 domains associated with malware activities; **Phishing**, characterized by 658 domains, associated with phishing pages. We expect that the OpenDNS resolver shows a higher protection level than the other ones because the datasets contain domain names blocked by Cisco Umbrella, which is also the company owing OpenDNS.

A. DNS resolvers

We selected three commercial Italian ISPs and two public DNS resolvers by Google and OpenDNS. **Google** is a free, public, and open resolver adopted by a high number of users, available at 8.8.8.8 and 8.8.4.4 IP addresses. It provides two different DoH APIs [23]. We utilized the one at endpoint https://dns.google/dns-query. **OpenDNS** is a free, public, and open resolver, founded in 2005 and currently owned by Cisco. It is available at 208.67.222.222, 208.67.220.220, 208.67.222.220, 208.67.220.222 IP addresses and it also provides DoH endpoints to implement the DNS over HTTPS [24]. We used endpoint at https://doh.opendns.com/dns-query. As local resolvers, we adopted the ones provided by **TIM**, **Wind**, and **Fastweb**, three of the major and most used Internet Service Providers (ISPs) in Italy. We carried out tests in two different cities - Naples and Rome. As reported by the Fair Internet Report, https://fairinternetreport.com/Italy/Rome, in both cities TIM, Wind, and Fastweb are three of the fastest providers in Italy. We conducted all experiments under three residential fibre Internet access networks by these operators.

Figure 1 shows how DNS queries have been performed from the same ISP network to the different resolvers. In the first case (top diagram in the figure), queries are issued towards the local resolver, i.e. the default resolver provided by the DHCP of the home router. We remark that for the security analysis it is not important to distinguish the responses arriving from the Home Router (i.e. from the cache) from the responses arriving from the ISP resolver because the former resolver gets information about domain names from the ISP one. In the response time analysis, instead, the time from the User PC to the Home Router, connected to each other by a wi-fi connection, is typically smaller than the time between such PC and the ISP resolver. In the second case, in Figure 1 (middle diagram in the figure), the client sends DNS requests to Google resolver directly. This scenario differs from the previous one because the Home and ISP routers are crossed from the DNS request only at the IP level. Thus, they are not involved in the dynamics of the application/DNS layer. The third scenario is similar to the second one except that requests are issued to the OpenDNS resolver.

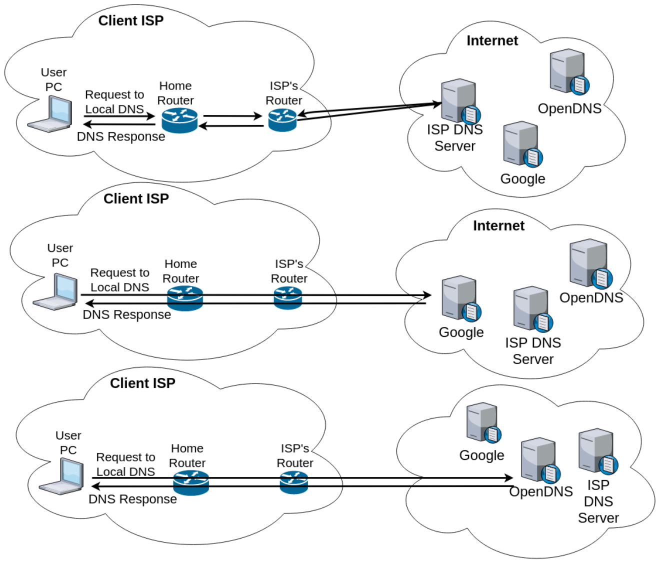

Fig. 1: Three scenarios of our system architecture

The second and third scenarios were applied to both DNS and DoH endpoints. The three scenarios were repeated under the three ISP networks: TIM, Wind, and Fastweb. In summary, under each ISP network, the queries were sent to: local DNS, Google DNS and DoH, and OpenDNS DNS and DoH. We performed the queries towards Google and OpenDNS resolvers under three ISP networks, aiming to investigate the impact of the network on the DNS resolvers.

IV. Experiments and Results

The DNS queries have been issued with Pydig specifying the DNS resolver address and the record type (e.g. *pydig(www.example.com, 8.8.8.8, A)*). The responses obtained by this tool include information related to a Resource Record In particular, it returns the following fields: **Response code** - specifies the outcome of the response. There are some common return codes that can be returned when issuing a DNS query (e.g. '0', '3', etc.) [4]. Other rare codes can appear in a few circumstances [25]. **IP** - contains one or more IP addresses associated with the requested domain name. It may also be null or contain a CNAME field. **Size** - represents the total size of the DNS response. **TTL** - specifies the Time To Live (TTL) of how long a record is cached in a DNS server. **Response time**

[4] https://support.umbrella.com/hc/en-us/articles/232254248-Common-DNS-return-codes-for-any-DNS-service-and-Umbrella-

- is the total amount of time to perform a query and receive the response. **Exception** - is true if an exception occurs during the DNS request.

We focused on the response code and the response time fields. The response code has been selected to study the resolver capability to distinguish benign and malicious domains. The analysis of the response time is aimed at understanding the timing performance of a DNS resolver.

A. Analysis of the Response Time

In this section, we focus on the response time of the DNS queries. Since PyDig is written in Python, an interpreted language, we verified the impact of the tool on the obtained values. We evaluated the difference between the response time provided by the Tshark tool and the one provided by PyDig on a sample of domain names. The average difference we observed in our setup is about 0.002s. This value may be significant for some experiments. However, it does not affect our analysis because we are using the same tool for each experiment, and we are interested in comparing the different resolvers. Figures 2, 3, 4, 5 show the comparison between local ISPs, Google (DNS,DoH), and OpenDNS - DNS, DoH - resolvers under the three ISP networks and for the four datasets.

We can make some interesting considerations. The first observation is related to the response times of the local resolver, which are smaller than those of public ones for each dataset and under the three networks. This is in contrast with the results reported by other works [4], [13], [14]. Another interesting finding is related to the Google resolver speed compared to that of OpenDNS. In all the figures mentioned above, Google-DNS and Google-DoH have slightly smaller response times than OpenDNS-DNS and OpenDNS-DoH. Besides that, under each network and for each dataset, Google DNS and DoH present similar response times, as, for example, shown in the overlap between the curves representing them. This outcome is in contrast with previous findings [17]. Similar behavior is shown by OpenDNS DNS and DoH. Some exceptions are visible for OpenDNS under the Wind network. In particular, Figures 3 (b), 4 (b), and 5 (b), show that OpenDNS-DNS is slower than OpenDNS-DoH in this case. Based on these considerations, we can infer that local DNS resolvers are faster than public ones, and,from our vantage points, Google is slightly faster than OpenDNS.

We also evaluated the distance between the curves aiming to compare response times between public and commercial resolvers to see how much time we can save by using local resolvers instead of public ones. For the sake of brevity, we report only the results related to Figure 4 (c). The differences between the curves have been calculated with the (4.1), (4.2), (4.3), (4.4), and (4.5), where F(RT)(t) is the response time function and i and j represent the various adopted resolvers. In particular, we computed: the difference between the minimum values from the two CDFs in (4.1), between the median values in (4.2), the mean values in (4.3), the standard deviation values in (4.4), and the 10-25-75-90th percentile values in (4.5). The results related to the four equations are reported in Table I.

$$RT_i - RT_j : F(RT_i) = min(F(RT)) \qquad i \neq j \tag{4.1}$$

$$RT_i - RT_j : F(RT_i) = median(F(RT)) \qquad i \neq j \tag{4.2}$$

$$RT_i - RT_j : F(RT_i) = mean(F(RT)) \qquad i \neq j \tag{4.3}$$

$$RT_i - RT_j : F(RT_i) = std\_dev(F(RT)) \qquad i \neq j \tag{4.4}$$

$$RT_i - RT_j : F(RT_i) = [10, 25, 75, 90]thF(RT) \qquad i \neq j \tag{4.5}$$

An interesting aspect is related to the `Min` column (as shown in Table I) where, in the best case, the Fastweb client is 82ms faster than Google DNS, 118ms faster than OpenDNS DNS, and Google is 36ms faster than OpenDNS DNS. On average, Fastweb is quicker by 86ms and 129ms than Google DNS and OpenDNS DNS, respectively. Google is 43ms faster than OpenDNS DNS. We report only the results related to the DNS protocol because those obtained with DoH are similar in most cases and not reported for brevity. We also looked at the impact of security on the performance, analyzing the trend of the response times split by the four datasets and the response codes. We have not found relevant differences in the results obtained from this analysis.

A further remaining issue relates to the impact of the DNS caching mechanisms on the obtained results. DNS caching can occur at different levels in a DNS lookup. The first two steps involve the operating system and the browser, and so they are related to the client. The other levels are associated with the resolver, root server and TLD server. To investigate the DNS caching impact on our experiments, we extracted 100 domains from the datasets, characterized by different TTL values, and therefore, presumably, different caching times [26]. We executed queries at different time intervals. In particular, the first execution took place after restarting the home routers of the clients used for the experiments (time 0, called *baseline* in the following). Then, we performed queries after 1 minute, 10 minutes, 1 hour, 4 hours and 24 hours. Figure 6 shows the CDFs of the response times obtained. The response times related to 1m, 10m, 1h, 4h are larger than the baseline and 24h. The number of domains kept in the cache is large for times up to one hour, decreases after 4 hours and reaches almost zero after 24 hours. After manual analysis, we have also reported a black dotted line in the plot to illustrate the DNS responses coming from the home router (response time equal to 4ms). Excluding domains cached in the router and comparing Figure 6 and Figure 4, we can claim that the local resolver is still faster than the public ones and considerations reported in the previous sections are still valid. Similar experiments and comparisons were also performed with TIM and Wind clients

i	j	Min	Median	Mean	Std_Dev	10th	25th	75th	90th
Fastweb	**Google DNS**	82	83	86	104	90	89	76	116
Fastweb	**OpenDNS DNS**	118	126	129	146	129	131	117	244
Google DNS	**OpenDNS DNS**	36	42	43	425	38	42	40	127

TABLE I: Results of the equations 4.1, 4.2, 4.3, 4.4, 4.5. All values are expressed in ms.

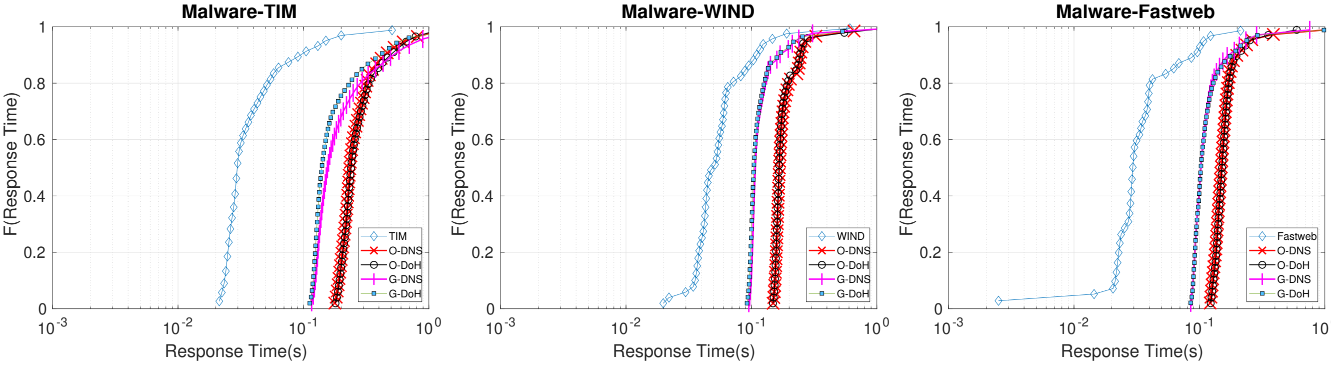

Fig. 2: Malware - Comparison local DNS with Google and OpenDNS for (a) TIM, (b) Wind, (c) Fastweb

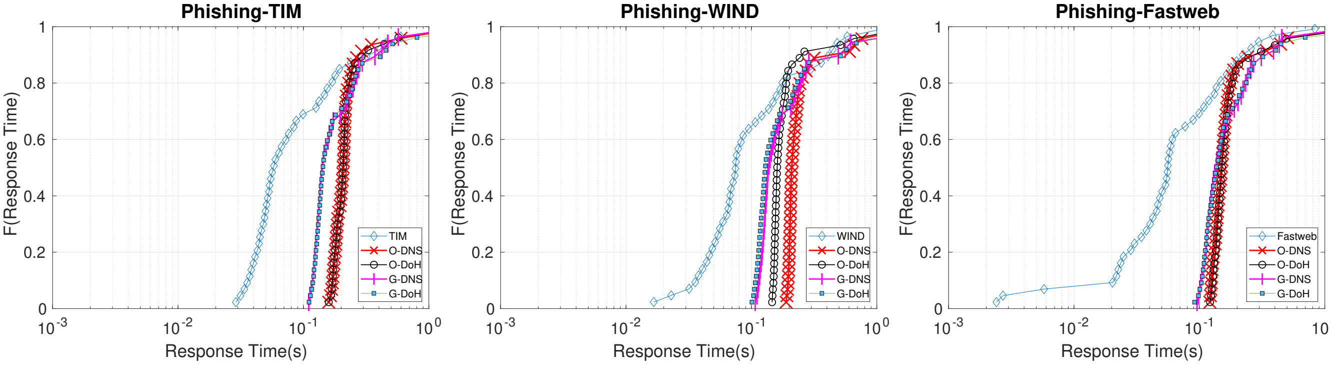

Fig. 3: Phishing - Comparison local DNS with Google and OpenDNS for (a) TIM, (b) Wind, (c) Fastweb

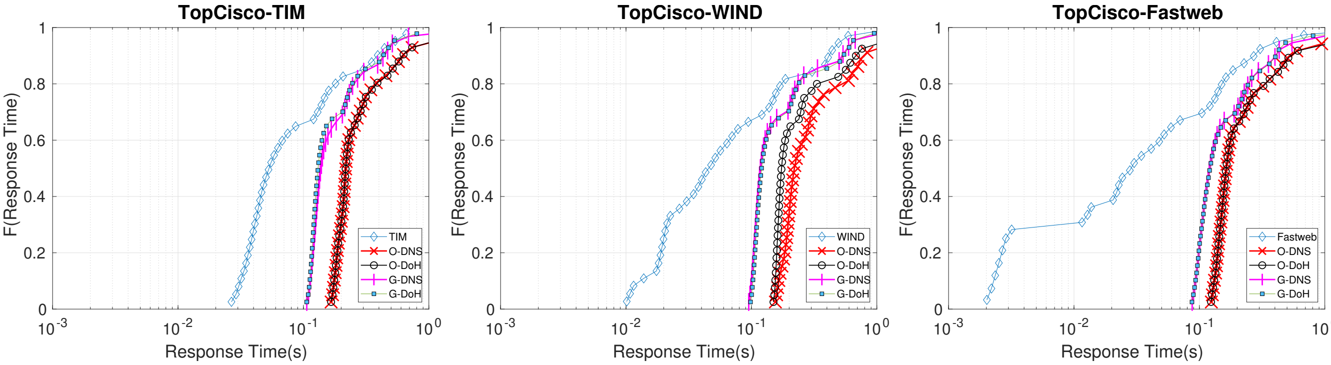

Fig. 4: TopCisco - Comparison local DNS with Google and OpenDNS for (a) TIM, (b) Wind, (c) Fastweb

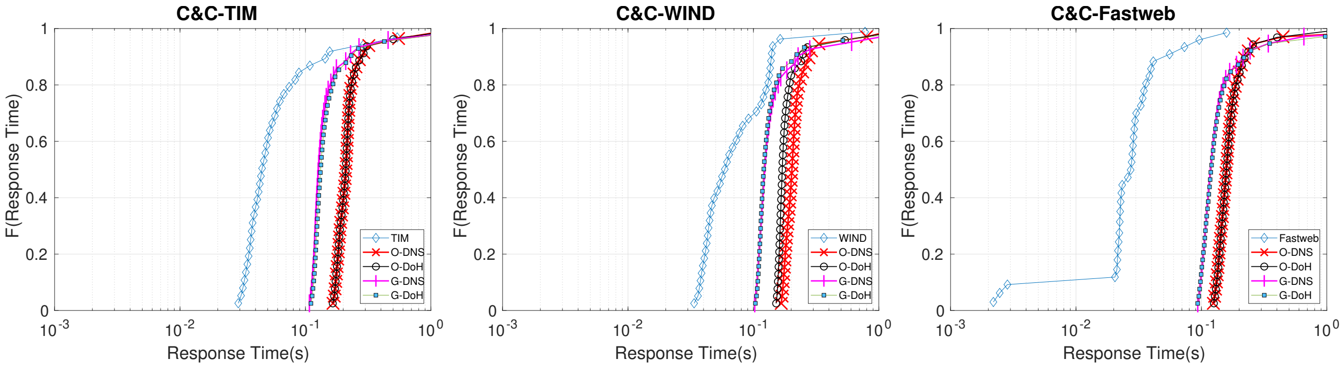

Fig. 5: C&C - Comparison local DNS with Google and OpenDNS for (a) TIM, (b) Wind, (c) Fastweb

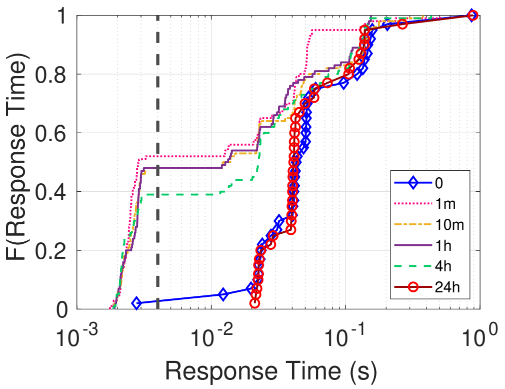

Fig. 6: Caching

DNS RC	Description	Class
0	DNS Query completed successfully	Positive
3	Domain Name does not exist	Negative
5	The server refused to answer	Negative
2	Server failed to process the query	Negative
Null	Other exceptions	Negative

TABLE II: Response codes identified in our experiments [25]

obtaining comparable considerations.

B. Analysis of the Response Code

The purpose of the response code analysis is to investigate the level of security service provided by the resolvers, to study how much they can protect users from possible threats.

Table II illustrates the DNS response codes obtained in our study. We identified four types of response codes and other exceptions due to network failures. The results differ from the ones shown by Park et al., who claim that codes "0", "3", "5", "2" decreased, and the remaining ones increased in the last years [27]. In particular, the "0" label occurs when the query is completed successfully [25]. In this label, we also included the cases in which the DNS server does not know the IP address of a host, and it returns another domain name through which the same destination can be reached (CNAME). In addition, in the same label, we added the case in which the response code is "0", but the DNS servers provide the SOA record (Start of Authority). Since the label "0" implies that the query was executed correctly and the DNS response has the IP addresses or information useful to obtain them, it represents the positive class of the confusion matrix. The latter is calculated to obtain a synthetic measure of the security capability of the resolvers [28]. The other response codes, "3", "2", "5", "Null" are errors. Therefore, we classify them as belonging to the negative class.

We summarize the occurrences of the response codes through bar plots showing a graph for each resolver adopted (locals, Google, OpenDNS), as depicted in Figures 7, 8, 9. There are four subplots in each figure relating to the four datasets: TopCisco, C&C, Malware, and Phishing. The bars in each subplot refer to a different network ISP: TIM, Wind and Fastweb. Results obtained with DoH are the same as those obtained with standard DNS and not reported for brevity. Figures 7, 8, 9 show that, for each provider, when the dataset is the TopCisco, the occurrences of "0" are more than 10K; the number of "3" is above 1K; the amount of "2", "5" and "None" differs for each resolver. The TopCisco results are in line with the expected ones because this dataset is mainly characterized by benign and existent hostnames. Therefore, we suspected that the number of "0" codes, and thus the number of queries to legitimate clients, was greater than the others. In addition, we can remark that, for the provider Wind, the number of "3" codes is lower than the other two local providers because there is a slightly higher amount of "0" and "Null". We point out that the highest percentage of "0" codes in the Wind provider is related to the courtesy page.

Considering the C&C dataset, the plots illustrate that the Wind resolver has the highest number of "0" codes and zero occurrences of "3", unlike TIM and Fastweb. The latter two are characterized, indeed, by a lower value of "0" occurrences, and also include a significant value of "3" occurrences.

Similar observations can be found regarding the Malware dataset. Looking at the two datasets in Figure 8, we can mark that Google resolver acts like TIM and Fastweb, returning a high number of "NXDOMAIN" rather than "NOERROR". OpenDNS, instead, reported in Figure 9, returns a high number of "3", but also a small percentage of "0" codes with an IP address related to a courtesy page. When the dataset is

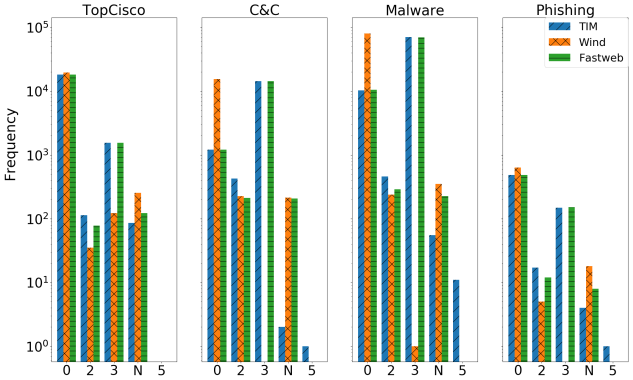

Fig. 7: Results obtained under three different IPS networks using the resolvers provided by their ISPs (i.e. using the local resolver).

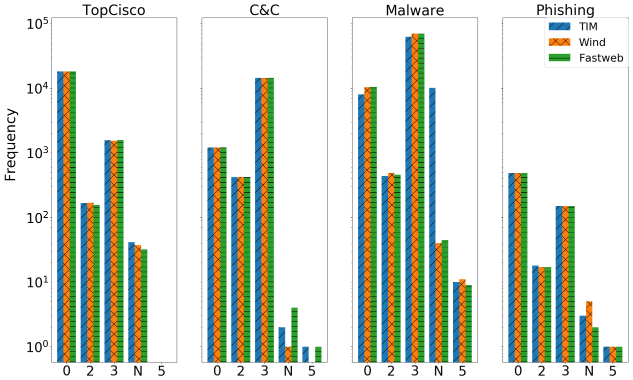

Fig. 8: Results obtained under three different IPS networks using the resolver provided by Google (i.e. using a public resolver).

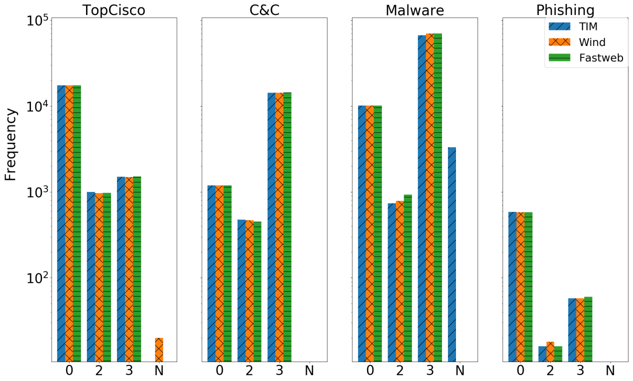

Fig. 9: Results obtained under three different IPS networks using the resolver provided by OpenDNS (i.e. using a public resolver).

Phishing, the number of "0" codes is higher than the number of "3" codes in all of the cases. In summary, all the DNS resolvers return the "NOERROR" message when the domain name is benign. Instead, when the domain is included in the C&C or Malware dataset, Wind returns the "NOERROR" message with a courtesy page IP address, while the other DNS resolvers return the "NXDOMAIN". Lastly, when the domain is related to a phishing client, all resolvers return a high number of "NOERROR" and a smaller number of "NXDOMAIN". A "NOERROR" message in the response does not always mean that everything is correct. For example, the code "0" is even common when the DNS resolver returns the IP address of a courtesy/splash page to prevent the user from accessing a potentially dangerous resource. We found that two DNS resolvers return a courtesy page in some cases: Wind and OpenDNS. For the analysis performed in this work, it is not needed to distinguish between the two cases above because the "NXDOMAIN" message still protects the user.

We further investigated the results about the "3" and "0" codes for each dataset and provider. We obtained that OpenDNS detects Malware and Phishing domain names better than Google. This behavior occurs similarly for each residential network (Tim, Wind, and Fastweb). For the sake of brevity and because it is the residential network with the fewest failures, we report only the comparison results for the Fastweb network. We filtered out the "0" code with a courtesy page address in the response. Specifically, about the Malware dataset, Google presents 0.09% more than OpenDNS for code "3", which presents 0.52% less than Google for code "0". About the Phishing dataset, Google presents 13.68% more code "3" than OpenDNS, which presents 53.5% less than Google for "0" code. We also investigated the local DNS resolvers against Google and OpenDNS. We obtained that local resolvers present higher performance than Google, as confirmed also by other works [13], [14]. Therefore, they are performing worse than OpenDNS, except for Wind provider, which shows a similar behavior about the courtesy page.

1) F-Measure and Accuracy: In the following, we report additional information regarding the security level of the resolvers. We calculated the **F-score** and **accuracy** as measures of their capability to detect malicious domains. The first step in calculating these two metrics consists in determining a confusion matrix characterized by positive and negative classes. The column "Class" of Table II summarizes the class of each RC received in our DNS responses. More specifically, if we perform a query to a benign IP, we expect a "0" code in the response. Conversely, with a malicious domain name, we should get a non-"0" rcode in the response. This is why the code "0" belongs to the positive class; the "3, 2, 5, Null" codes to the negative class. Since the domain names of the TopCisco dataset is characterized mainly by benign domains, the corresponding DNS responses include a large number of "0" labels. Consequently, this dataset belongs to the "positive" class. The other three lists contain only malicious hostnames and, hence, the corresponding DNS responses consist of a high amount of "3,2,5,Null" codes. For this reason, they belong to the "negative class". Summarizing, the elements of our confusion matrix are the following: `True Positive` (TP): IP addresses obtained querying domains from the TopCisco dataset; `True Negative` (TN): NXDomain obtained querying domains from the C&C, Malware and Phishing lists; `False Positive` (FP): IP addresses obtained querying domains from the C&C, Malware and Phishing list; `False Negative` (FN): NXDomain obtained querying domains from the TopCisco dataset. For Google and OpenDNS, we report only the results obtained until the Fastweb network. We see that Google presents a F-score equal to 71%; OpenDNS 73%, Fastweb 72%, Wind 28% and TIM 72%. Concerning the accuracy measure, TIM, Fastweb and Google reach a percentage of 88.1%; OpenDNS has an accuracy of 87.7%; Wind is accurate to about 17.5%. As mentioned before, the F-score and accuracy values for the Wind resolver are lower than the others mainly because of the huge amount of courtesy pages contained in the DNS response with a "0" code. In addition, OpenDNS shows a lower accuracy value than the other resolvers because it applies a hybrid approach.

2) Analyzing recurrent IP addresses: We also examined the IP addresses with a high number of occurrences. We report those obtained with the Wind resolver related to the Malware dataset: the IP address 40.68.249.35 occurs slightly less than 100.000 times, 86% of the times in our experiments. We checked that it corresponds to the IP address of a courtesy page. Other interesting IP addresses are: 216.218.185.162, 64.70.19.203, 34.102.136.180, 35.102.136.180. These IPs are consistently reported by all the resolvers. Querying the Whois tool, we discovered that the first IP belongs to Hurricane Electric LLC. The second one is related to CenturyLink Communications, LLC. The third and the fourth ones belong to Google LLC. These IP addresses are obtained only from the C&C and Malware datasets.

Hurricane Electric has already been traced back to malicious DNS activities. Anyone could register for a free account with Hurricane Electric's hosted DNS service. It is possible to register a zone and create A records, even causing the hijacking of legitimate domains because the provider does not check if zones created by their users have already been registered (e.g., see [29]).

a) Hurricane Electric: the address is `216.218.185.162` and the hostname is *216-218-185-162.sinkhole.shadowserver.org*. Shadow server[5] is a non-profit security organization that gathers and analyzes data on malicious Internet activity, including malware and botnet. They provide a sinkhole service used for spoofing DNS requests to prevent the resolution of malicious hostnames. It can be accomplished by configuring DNS resolvers that return a sinkhole address for a specific domain name. One of the nameservers of Shadowserver operator is sinkhole.shadowserver.org - 216.218.185.160/29, that we found in our results [30]. All domain names have a .xyz top-level domain (TLD) and a TTL value equal to 21599.

b) CenturyLink Communication: We have also investigated the IP address related to Century Link Communication. We performed reverse DNS lookup queries and got PTR records from IP addresses with the dig tool. The domain name related to this IP address is mailrelay.203.website.ws, useful to register a new .ws domain.

c) Google LLC: The last two IP addresses belong to Google LLC. In more detail, the first IP address 34.102.136. 180 is related to the 180.136.102.34.bc.googleusercontent.com domain name. This domain is adopted for multiple purposes, like cached copies of websites visited by Google search engine and to store static content including images [31]. In different cases, hackers hide malicious code inside image files that are rarely scanned for malware [32].

In conclusion, we observed that OpenDNS is slightly slower than Google and local resolvers. However, local resolvers are faster than public ones. Moreover, all resolvers analyzed protect users from most malicious domain names. OpenDNS provides a higher level of protection for Malware and Phishing domain names than Google and local resolvers. Wind presents a behavior similar to the one of OpenDNS.

V. Conclusion

In this work, we investigated the behavior of different DNS resolvers. In particular, we evaluated their capability to recognize malicious domains (i.e., to protect clients), and the response time between them and their clients. We focused on two classes of resolvers: local DNS resolvers from main Italian ISPs (TIM, Wind, and Fastweb), and public resolvers by Google and OpenDNS. We based our analysis on the *Response Time* and *Response Code* obtained from the queries. The first one has been used to understand the speed of resolution of a domain name. The response code has been used to study how much a DNS resolver can recognize a domain name associated with malicious activity.

The results about the **Response Time** show that: (i) the local DNS resolvers are generally faster than public resolvers; (ii) Google is slightly faster than OpenDNS; (iii) there are no significant differences between DNS and DoH of both Google and OpenDNS. We have also computed the time we can gain using a resolver in spite of another, obtaining that: (i) Fastweb is 86ms faster than Google on average; (ii) Fastweb is 129ms faster than OpenDNS in average; (iii) Google is 43ms faster than OpenDNS in average. We also show that the increased speed of local DNS resolvers against public ones is confirmed even if we exclude domains cached at the home router.

The results about the **Response Code** show that some local DNS resolvers and Google return a "NXDOMAIN" message for malicious domains. Other resolvers, instead, provide a "0" RCODE with a courtesy IP address. OpenDNS behaves in a hybrid manner. The resolvers analyzed achieve good security levels, protecting users from most malicious domain names. In addition, OpenDNS achieves a slightly higher level than local resolvers and Google with malware and phishing domain names. In addition, both the DNS and DoH protocols tested with Google provide the same results in terms of RCODE.

[5] https://www.shadowserver.org/

The same behavior is also observed with the two protocols tested with OpenDNS. We also examined security capabilities as a function of the dataset and we obtained no significant differences.

We believe that our analysis is first and foremost useful for the scientific community and network operators to gain a better knowledge of the DNS and how to improve it. In addition, our results may provide insights to users in choosing the most appropriate DNS and, more generally, to the community on how the DNS works, which is far beyond just translating domains into IP addresses, as originally conceived. In future work, the study should be repeated involving other non-Italian ISPs and more open DNS resolvers. In addition, it could be interesting to confirm our findings by using malicious domain names collected by other blacklists.

Acknowledgment

This work was partially supported by MIUR through the "ICT for Health" project, Dipartimento di Eccellenza (2018-2022) "Ingegneria Elettrica e delle Tecnologie dell'Informazione" and by Cisco Systems through the Sponsored Research Agreement "Research Project for Industry 4.0".

References

[1] J. Klensin, "Rfc 3467 - role of the domain name system (dns)," 2003. [Online]. Available: https://tools.ietf.org/html/rfc3467

[2] K. Schomp, M. Allman, and M. Rabinovich, "Dns resolvers considered harmful," in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, ser. HotNets-XIII. New York, NY, USA: Association for Computing Machinery, 2014, p. 1–7.

[3] C. A. Shue and A. J. Kalafut, "Resolvers revealed: Characterizing dns resolvers and their clients," *ACM Trans. Internet Technol.*, vol. 12, no. 4, Jul. 2013. [Online]. Available: https://doi.org/10.1145/2499926.2499928

[4] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig, "Comparing dns resolvers in the wild," ser. IMC '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 15–21.

[5] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going wild: Large-scale classification of open dns resolvers," in *Proceedings of the 2015 Internet Measurement Conference*, 2015.

[6] J. Park, A. Khormali, M. Mohaisen, and A. Mohaisen, "Where are you taking me? behavioral analysis of open dns resolvers," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2019, pp. 493–504.

[7] Y. He, Zhenyu Zhong, S. Krasser, and Y. Tang, "Mining dns for malicious domain registrations," in *6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010)*, 2010, pp. 1–6.

[8] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through dns data analysis," *ACM Comput. Surv.*, vol. 51, no. 4, Jul. 2018.

[9] A. Botta, G. E. Mocerino, S. Cilio, and G. Ventre, "A machine learning approach for dynamic selection of available bandwidth measurement tools," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[10] A. Affinito, A. Botta, L. Gallo, M. Garofalo, and G. Ventre, "Spark-based port and net scan detection," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, ser. SAC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1172–1179. [Online]. Available: https://doi.org/10.1145/3341105.3373970

[11] T. Böttger, F. Cuadrado, G. Antichi, E. L. a. Fernandes, G. Tyson, I. Castro, and S. Uhlig, "An empirical study of the cost of dns-over-https," in *Proceedings of the Internet Measurement Conference*, ser. IMC '19. NY, USA: Association for Computing Machinery, 2019, p. 15–21.

[12] C. Lu, B. Liu, Z. Li, S. Hao, H. Duan, M. Zhang, C. Leng, Y. Liu, Z. Zhang, and J. Wu, "An end-to-end, large-scale measurement of dns-over-encryption: How far have we come?" in *Proceedings of the Internet Measurement Conference*, ser. IMC '19, 2019.

[13] "Ranking the performance of public dns providers," https://blog.thousandeyes.com/ranking-performance-public-dns-providers-2018/, (Accessed on 02/17/2021).

[14] "Why you shouldn't use your isp's default dns server," https://www.howtogeek.com/664608/why-you-shouldnt-be-using-your-isps-default-dns-server/, (Accessed on 02/17/2021).

[15] D. Dagon, C. Lee, W. Lee, and N. Provos, "Corrupted dns resolution paths: The rise of a malicious resolution authority," in *Proc. 15th Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2008.

[16] "Dns performance - compare the speed and uptime of enterprise and commercial dns services | dnsperf," https://www.dnsperf.com/, (Accessed on 02/17/2021).

[17] A. Hounsel, K. Borgolte, P. Schmitt, J. Holland, and N. Feamster, "Comparing the effects of dns, dot, and doh on web performance," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 562–572.

[18] A. Hounsel, P. Schmitt, K. Borgolte, and N. Feamster, "Can encrypted dns be fast?" Passive and Active Measurement Conference 2021-PAM.

[19] "Github - shuque/pydig: pydig: a dns query tool written in python," https://github.com/shuque/pydig, (Accessed on 07/28/2020).

[20] A. Affinito, A. Botta, and G. Ventre, "The impact of covid on network utilization: an analysis on domain popularity," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020, pp. 1–6.

[21] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, "A long way to the top: Significance, structure, and stability of internet top lists," in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 478–493.

[22] M. R. McNiece, R. Li, and B. Reaves, "Characterizing the security of endogenous and exogenous desktop application network flows," in *Passive and Active Measurement: 22nd International Conference, PAM 2021, Virtual Event, March 29 – April 1, 2021, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2021, p. 531–546. [Online]. Available: https://doi.org/10.1007/978-3-030-72582-2_31

[23] "Dns-over-https (doh) | public dns | google developers," https://developers.google.com/speed/public-dns/docs/doh, (Accessed on 01/21/2021).

[24] "Using dns over https (doh) with opendns – opendns," https://support.opendns.com/hc/en-us/articles/360038086532-Using-DNS-over-HTTPS-DoH-with-OpenDNS, (Accessed on 01/21/2021).

[25] "Domain name system (dns) parameters," https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-6, (Accessed on 01/25/2021).

[26] A. Kumar, "Common dns implementation errors and suggested fixes," 1993. [Online]. Available: https://tools.ietf.org/rfc/rfc1536.txt

[27] J. Park, A. Khormali, M. Mohaisen, and A. Mohaisen, "Where are you taking me? behavioral analysis of open dns resolvers," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2019, pp. 493–504.

[28] "A simple guide to building a confusion matrix," https://blogs.oracle.com/ai-and-datascience/post/a-simple-guide-to-building-a-confusion-matrix#:~:text=The%20confusion%20matrix%20is%20represented,normality%20or%20a%20normal%20behavior., (Accessed on 02/11/2022).

[29] "Operation poisoned hurricane | fireeye inc," https://www.fireeye.com/blog/threat-research/2014/08/operation-poisoned-hurricane.html, (Accessed on 01/28/2021).

[30] "Understanding dns sinkholes - a weapon against malware - infosec resources," https://resources.infosecinstitute.com/topic/dns-sinkhole/, (Accessed on 02/11/2021).

[31] "Googleusercontent.com can trip you up, if you disable third-party cookies | get more done, with kerika," https://blog.kerika.com/googleusercontent-com-can-trip-you-up-if-you-disable-third-party-cookies/, (Accessed on 02/12/2021).

[32] "Hiding malware inside images on googleusercontent," https://blog.sucuri.net/2018/07/hiding-malware-inside-images-on-googleusercontent.html.